

计算机最新技术培训教材

PEKING UNIVERSITY PRESS

周珂伟 编著

Visual Basic 6.0

数据库开发

学习教程



- 详细讲述 Visual Basic 6.0 数据库开发的方方面面
- 从入门到精通的 Visual Basic 6.0 数据库开发教程

北京大学出版社

<http://cbs.pku.edu.cn>



计算机最新技术培训教材

Visual Basic 6.0 数据库开发学习教程

周珂伟 编著

北 京 大 学 出 版 社
北 京

内 容 简 介

本书是一本讲述 Visual Basic 6.0 数据库编程实例的书籍。第一、第二和第三章介绍了有关 Visual Basic 6.0 和数据库方面的一些基本知识,从第四章开始讲述用 Visual Basic 6.0 开发数据库应用程序的具体实例。在本书的第十、第十一及十二章中详细地介绍了关于 ADO 对象、ADO 属性和 ADO 方法的实例。本书的最后一章给出了一个较为大型的数据库应用实例。

图书在版编目(CIP)数据

Visual Basic 6.0 数据库开发学习教程/周珂伟编著. —北京:北京大学出版社,2000.4
ISBN 7-301-01764-2

I. V... I. 周... II. BASIC 语言-程序设计-教材 N. TP312

中国版本图书馆 CIP 数据核字(2000)第 06659 号

书 名: Visual Basic 6.0 数据库开发学习教程

著作责任者: 周珂伟

责任编辑: 杨锡林

标准书号: ISBN 7-301-01764-2/TP·120

出版者: 北京大学出版社

地 址: 北京市海淀区中关村北京大学校内 100871

网 址: <http://cbs.pku.edu.cn>

电 话: 出版部 62752015 发行部 62754140 编辑室 62765013

电子信箱: zpup@pup.pku.edu.cn

排 版 者: 兴盛达激光照排中心

印 刷 者: 河北省滦县印刷厂

发 行 者: 北京大学出版社

经 销 者: 新华书店

787 毫米×1092 毫米 16 开本 17.375 印张 429 千字

2000 年 5 月第一版 2000 年 5 月第一次印刷

定 价: 28.00 元

前 言

在计算机技术日益发展的今天，开发数据库程序已经成为计算机应用的一个重要的领域。如今，各种数据库的开发工具层出不穷，而 Visual Basic 以其完美的界面、方便的操作、强大的功能在各种数据库开发工具中独树一帜。

作为微软公司的重点产品，Visual Basic 具有其它数据库开发工具所无法比拟的优势。它易学易用，因此得到了广大软件开发人员及计算机爱好者的青睐。现在，Visual Basic 已经发展到 6.0 版本，它是一种完全面向对象的和可视化的开发工具，可以开发几乎所有的 Windows 应用程序。本书采用的是 Visual Basic 6.0 中文版，用它来实现 Windows 下数据库应用程序的开发，并采用实例和解析的方式，详细地讲述这个开发工具的各种功能，以及进行数据库应用程序开发的操作方法和实用技巧，使读者尽快地用学到的知识开发出自己的数据库应用程序。

本书由孙景利策划，由周珂伟主编。参加编写的人员还有：张凯峰、娄毅、王苗苗、李浩、李锐、陈和霖、聂伯敏、张焰、王元元、邹坚、董蜀峰、冯红霞、杨远、黄建华、瞿磊、张燕秦、郑洪强、侯斌、李华斌、刘炳刚、李永辉、徐峰、孙旋、徐建国、艾力江、杨宝勇、李隽、朱广慧等。

由于时间仓促，以及笔者的水平有限，本书中难免存在不足之处，望广大读者朋友不吝提出批评。

周珂伟

1999 年 12 月 16 日

目 录

第一章 了解 Visual Basic	(1)
1.1 Visual Basic 入门	(1)
1.2 Visual Basic 编程初步	(11)
1.3 Visual Basic 程序控制流	(17)
1.4 Visual Basic 的内部函数	(19)
第二章 数据库概述	(24)
2.1 数据库系统与模型	(24)
2.2 关系数据库	(27)
2.3 SQL 语言	(30)
第三章 Visual Basic 数据库开发	(40)
3.1 客户/服务器结构	(40)
3.2 ActiveX 数据对象简介	(42)
3.3 ADO 对象	(43)
3.4 Visual Basic 对数据和数据源的访问	(46)
第四章 配置数据源实例	(55)
4.1 ODBC 数据源介绍	(55)
4.2 Access 数据源的配置	(56)
4.3 SQL Server 数据源的配置	(58)
第五章 连接数据库实例	(61)
5.1 Connection 对象介绍	(61)
5.2 连接到数据库的实例	(68)
第六章 使用 Data 控件	(72)
6.1 Data 控件的实例	(72)
6.2 ADO Data 控件的实例	(76)
6.3 Remote Data 控件	(82)
第七章 使用 DataCombo 和 DataList	(85)
7.1 使用 DataCombo 控件的实例	(85)
7.2 使用 DataList 控件的实例	(90)
第八章 使用 DataGrid 控件	(92)
8.1 更改数据源的实例	(92)
8.2 关于列的实例	(98)
8.3 DataGrid 视图	(101)
8.4 使用书签的实例	(103)
8.5 类模块一起使用 DataGrid 控件	(105)

第九章 使用 Data Report 实例	(111)
9.1 创建数据报表	(111)
9.2 扩展数据报表	(118)
第十章 关于 ADO 对象的实例	(126)
10.1 DataControl 对象实例	(126)
10.2 DataSpace 对象和 CreateObject 方法实例	(128)
10.3 DataFactory 对象实例	(129)
第十一章 关于 ADO 属性的实例	(132)
11.1 AbsolutePage、PageCount 和 PageSize 属性实例	(133)
11.2 AbsolutePosition 和 CursorLocation 属性实例	(134)
11.3 ActiveConnection 等属性实例	(134)
11.4 ActualSize 和 DefinedSize 属性实例	(136)
11.5 Attributes 和 Name 属性实例	(136)
11.6 BOF、EOF 和 Bookmark 属性实例	(137)
11.7 CacheSize 属性实例	(139)
11.8 Connect 属性实例	(141)
11.9 ConnectionString 等属性实例	(142)
11.10 Count 属性实例	(143)
11.11 CursorType 等属性实例	(144)
11.12 Description 等属性实例	(145)
11.13 ExecuteOptions 和 FetchOptions 属性实例	(146)
11.14 Filter 和 RecordCount 属性实例	(147)
11.15 FilterColumn 等属性实例	(149)
11.16 IsolationLevel 和 Mode 属性实例	(150)
11.17 MarshalOptions 属性实例	(152)
11.18 MaxRecords 属性实例	(153)
11.19 NumericsCale 和 Precision 属性实例	(154)
11.20 OriginalValue 和 UnderlyingValue 属性实例	(155)
11.21 Prepared 属性实例	(156)
11.22 Provider 和 DefaultDatabase 属性实例	(157)
11.23 Recordset 和 SourceRecordset 属性实例	(158)
11.24 ReadyState 属性实例	(160)
11.25 Server 属性实例	(160)
11.26 Source 属性实例	(162)
11.27 SQL 属性实例	(163)
11.28 State 属性实例	(165)
11.29 Status 属性实例	(166)
11.30 Type 属性实例	(167)
11.31 Value 属性实例	(168)

11.32	Version 属性实例	(169)
第十二章	关于 ADO 方法的实例	(170)
12.1	AddNew 方法实例	(170)
12.2	Append 和 CreateParameter 方法实例	(174)
12.3	AppendChunk 和 GetChunk 方法实例	(176)
12.4	BeginTrans 等方法实例	(177)
12.5	Cancel 方法实例	(179)
12.6	CancelUpdate 方法实例	(180)
12.7	Clone 方法实例	(182)
12.8	ConvertToString 方法实例	(184)
12.9	CreateRecordset 方法实例	(185)
12.10	Delete 方法实例	(185)
12.11	Execute 等方法实例	(189)
12.12	GetRows 方法实例	(195)
12.13	Move 方法实例	(196)
12.14	MoveFirst 等方法实例	(198)
12.15	NextRecordset 方法实例	(199)
12.16	Open 和 Close 方法实例	(200)
12.17	OpenSchema 方法实例	(203)
12.18	Refresh 方法实例	(204)
12.19	Resync 方法实例	(206)
12.20	SubmitChanges 方法实例	(206)
12.21	Supports 方法实例	(208)
12.22	Update 和 CancelUpdate 方法实例	(210)
12.23	UpdateBatch 和 CancelBatch 方法实例	(213)
第十三章	电话管理系统实例	(215)
13.1	系统的环境	(215)
13.2	系统的功能介绍	(215)
13.3	数据库描述	(218)
13.4	程序的模块说明	(223)
13.5	模块的程序代码	(251)

第一章 了解 Visual Basic

计算机世界日新月异地发展着,在 Microsoft Windows 被成千上万的用户所使用的今天,开发 Windows 下的图形用户界面程序(GUI)对于广大的程序员来说已经成为一件刻不容缓的事。在过去,开发 Windows 应用程序需要有专门的 C 程序员,他们面对的是让一般程序员所望而生畏的繁琐的 Windows API 函数,这使得编写一个最为简单的 Windows 应用程序也得需要成百上千行的代码。正是因为如此,当 Visual Basic1.0 版本问世时,Microsoft 公司的总裁及首席执行官 Bill Gates 称它为“惊世骇俗的”。如今,Visual Basic 已经发展到了 6.0 版本,它是一种完全的面向对象和可视化的开发工具。使用 Visual Basic 6.0 可以开发几乎所有的 Windows 应用程序。本书讲述的是使用这个功能强大的开发工具来实现 Windows 下数据库程序的开发,采用的版本为 Visual Basic 6.0 中文版(这个全名很费口舌,为了简便起见,从现在开始,本书中一般将它称之为 Visual Basic)。当你读完这本书之后,你会惊讶地发现编写近乎完美的 Windows 数据库应用程序已经不再是梦想。

在本章中我们主要讲述以下几个部分:

- Visual Basic 入门
- Visual Basic 编程初步
- Visual Basic 程序控制流
- Visual Basic 的系统函数

1.1 Visual Basic 入门

1.1.1 什么是 Visual Basic

Microsoft 的 Visual Basic,是一种提供了开发 Windows 应用程序最迅速、最简捷的开发工具。不论是 Windows 应用程序的资深专业开发人员还是初学者,Visual Basic 都为他们提供了整套工具,以方便开发应用程序。特别是对于那些开发 Windows 应用程序的初学者而言,Visual Basic 提供了一条通往 Windows 应用程序的捷径,由于 Visual Basic 的面向对象特征和封装性,使得 Windows 程序员们无需面对数以千计的 API 函数,从而摆脱了无休止的界面设计,把主要精力投入到程序的总体性构思和模块的编写中去。正如 Charles Petzold 所说的“对于那些试图让我们相信编写 Windows 应用程序是如何困难的人来说,Visual Basic 对他们的生存之道产生了真正的威胁”。

那么,何谓 Visual Basic 呢?“Visual”的英文意思是“可视的”,它指的是开发图形用户界面的方法。程序开发人员不需编写大量代码去描述界面元素的外观和位置,而只要把预先建立的对象加到屏幕上即可(例如可以拖动一个命令按钮到屏幕上)。“Basic”指的是 Basic (Beginners All-Purpose Symbolit Instruction Code) 语言,一种在计算技术发展历史上应用得最为广泛的语言。Visual Basic 是在原有 Basic 语言的基础上进一步发展起来的,至今包含了数百

条语句、函数及关键词,其中很多和 Windows GUI 有直接关系。专业人员可以用 Visual Basic 实现其它任何 Windows 编程语言的功能,而初学者只要掌握几个关键词就可以建立实用的应用程序。

1.1.2 Visual Basic 的安装

1. Visual Basic 的三种版本

- 标准版:生成 Microsoft Windows 和 Windows NT 的应用程序,包含了所有的固有控件。
- 专业版:提供了专业程序员的 Visual Basic 开发方案,提供了全套的工具。除了标准版的功能之外,还附加了 ActiveX 控件, IIS 应用程序开发环境集成的数据工具和数据环境,动态 HTML 页开发环境等。
- 企业版:具有专业版所提供的所有功能,外加 BackOffice 工具。

2. 安装 Visual Basic 对系统的要求

安装 Visual Basic 需要 90MHz 以上的奔腾级 CPU(建议采用 166MHz 以上的奔腾级 CPU);VGA 兼容的显示卡,并支持高于 640×480 的分辨率;在 Windows 95 下,需要 24MB 的内存空间(建议有 32MB 以上的内存空间),在 Windows NT 下需要 32MB 的内存空间(建议有 48MB 以上的内存空间)。Windows 的版本应该不低于 Windows 95,NT 的版本应该不低于 NT3.51,有不低于 4.01 版本的 Internet Explorer(IE)以便来支持 Visual Basic 的 DHTML(动态的 HTML 语言)。

以下是对硬盘空间的要求:

	标准版	专业版	企业版
典型安装	48MB	48MB	128MB
完全安装	80MB	80MB	147MB

外加 Microsoft 开发者文档 67MB。

3. 安装 Visual Basic

Visual Basic 安装盘内包含着安装程序 Setup.exe,当运行这个安装程序时,安装程序会检查 Visual Basic 先前的版本号及相应的组件,如果发现系统中存在先前的 Visual Basic 版本,安装程序会给出一个提示问你是否要继续,当继续安装时,它就会覆盖掉先前的版本。在这儿,我们不准讨论安装时的细节问题,读者只需跟随着安装向导来进行就可以了。图 1.1 为 Visual Basic 的安装程序。

完成安装以后,Visual Basic 便会自动地安装在自己的程序组中,用户可以通过根据以下的选择来运行 Visual Basic:开始→程序→Microsoft Visual Basic 6.0 中文版。当然,用户还可以其他方法来运行 Visual Basic:可以通过移动到 Visual Basic 的目录中并双击 Visual Basic 的图标,来从 Microsoft Windows 的桌面启动 Visual Basic;还可以通过打开运行对话框并输入 Visual Basic 的路径来运行 Visual Basic;最通用的方法是在桌面上新建一个指向 Visual Basic 的快捷方式(这也是笔者所建议采用的方式),每当要运行 Visual Basic 时,只需双击这个快捷方式即可。

现在,我们就可以来运行 Visual Basic 了。当运行时,首先会看到屏幕上显示的有关版权的信息,它会告诉读者谁是该份 Visual Basic 拷贝的合法使用者,在此之后,就会出现 Visual



图 1.1 Visual Basic 6.0 中文专业版的安装

Basic 的初始屏幕,关于这方面的内容我们将在下一部分 Visual Basic 的使用中详细介绍。

1.1.3 Visual Basic 的使用

1. 集成环境简介

Visual Basic 是最优秀的 Windows 应用程序开发工具之一,它的开发过程完全是按照所见即所得的要求来实现的。Visual Basic 本身就带有一个供程序开发人员使用的非常完美的开发环境,如图 1.2 所示。在这个开发环境中,Visual Basic 提供了几乎所有开发者将可能用到的功能,包括工程的建立、应用程序界面的设计、源代码的编写、程序的调试运行和最终可执行文件的生成等功能。如果读者安装了 MSDN(Microsoft Developer Network),就可以十分方便地查到所需要的帮助材料。关于 MSDN,我们将在后面的联机帮助系统中做详细的介绍。

现在,就让我们进入 Visual Basic 的集成环境中,去看一下 Visual Basic 为我们开发 Windows 应用程序提供了哪些功能。在图 1.2 的上方是 Visual Basic 的菜单条,在菜单条中,我们发现不仅有我们极为熟悉的文件、编辑、视图、窗口、帮助等菜单项,还具有工程、调试、外接程序等在用 Visual Basic 开发应用过程中经常要用到的菜单项。菜单条下方的是工具条,Visual Basic 把那些最为常用的菜单项功能(例如添加窗体、打开工程、保存工程等)放在工具条上面,使开发者避免了每次都要到菜单条中去选取所需要的菜单功能项,从而大大方便了开发人员的开发过程。在集成环境的中间是开发者建立的工程,它以图形化的方式展现在开发者的面前,工程是一个在开发过程中的概念名词,读者可以把它同建筑业上的工程相类比。如同建筑业上的工程,在 Visual Basic 中,应用程序就是在工程的基础上一步一步地构造起来的。集成环境的左边是工具箱(注意:不要把工具箱和工具条混淆),那些在开发过程中经常要用到的控件被安放在工具箱上。右边有两个窗口,上面一个是工程资源管理器,工程中包括的内容在这里得以结构化地显示。下面一个窗口称之为属性窗口,它显示了在工程中得到焦点的控件属

性,从这个意义上来看,窗体(Form)也可被视为控件。在接下来的几部分里,我们将对集成环境中的菜单条和工具箱做详细介绍,读者在看这几部分时,最好能亲自动手到计算机上试一试。

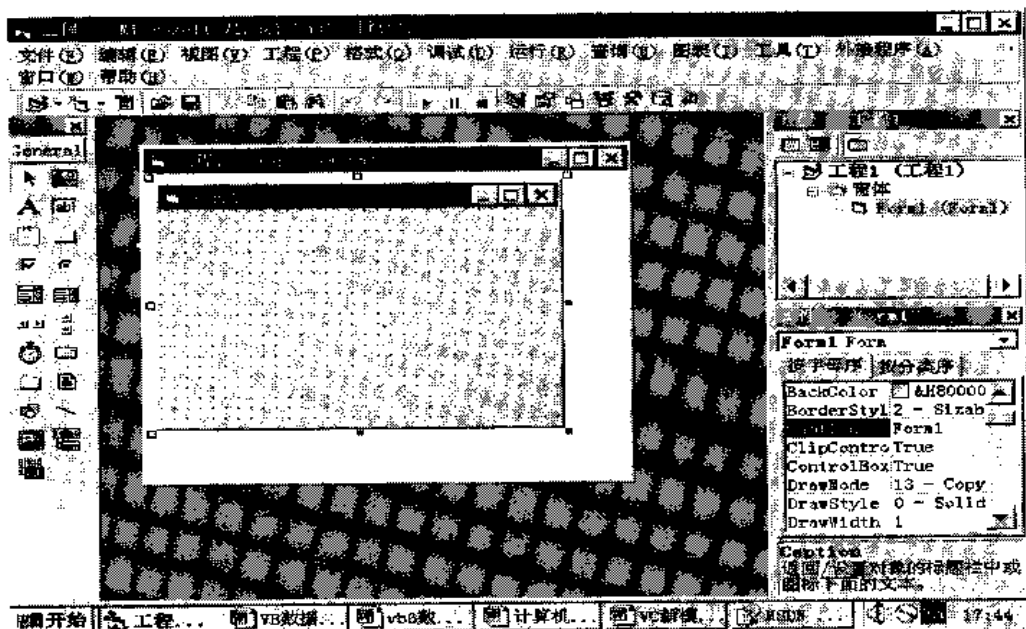


图 1.2 Visual Basic 的集成环境

2. Visual Basic 的菜单

相信读者对菜单应该不会感到陌生,接下来我们将对 Visual Basic 中的菜单功能进行介绍。由于篇幅所限,对那些并不常用或显而易见的菜单功能就不再一一讲述。

(1) “文件”菜单

文件菜单项中主要包括了以下几项:

- “新建工程”项: 新建一个 Visual Basic 的工程, Visual Basic 的工程类型有标准的 EXE、ActiveX EXE、VB 应用程序向导等十多种类型。通常我们都选择标准的 EXE 类型。
- “打开工程”项: 打开一个已经存在的工程文件。在 Visual Basic 中, 工程文件是以 .vbp (Visual Basic Project) 作为扩展名的。
- “添加工程”项: 有时候, 应用程序需要同时具有几个不同的工程, 此时开发者可以在原有工程的基础上给应用程序添加一个新的工程。
- “移除工程”项: 与添加工程相对, 开发者可以从应用程序中移除一个工程。
- “保存工程”项: 当开发者对工程中的某些部分做了改变或添加了某些部分后, 可以选择保存工程来对所做的改变或添加进行保存, 如果在未保存所做的改变或添加之前退出 Visual Basic 环境, Visual Basic 会提示你要不要对所做的改变或添加进行保存。
- “工程另存为”项: 可以把当前的工程存为另一个工程文件。
- “生成 .exe”项: 把当前的工程编译成可执行文件, 使得它可以脱离集成环境而运行。

(2) “视图”菜单

视图菜单项中主要包括了以下几项:

- “代码窗口”项：显示当前所取得焦点的控件的代码。你也可以在焦点控件处单击鼠标右键，从而在弹出的菜单中选择“查看代码”项。
- “对象浏览器”项：显示所有工程的对象结构。如图 1.3 所示，你可以在对象浏览器中的下拉框里选择一个工程，然后你就可以看到在此工程下的所有对象和变量。

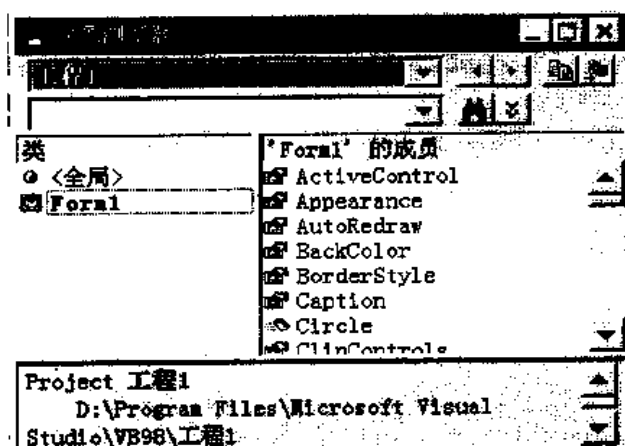


图 1.3 对象浏览器

- “立即窗口”项：当在集成环境中调试或运行应用程序时，你可以打开立即窗口，在立即窗口中你可以观察到在程序运行过程中变量的值，这对于开发者调试所编写的应用程序是极为有用的。在立即窗口中观察变量值的方式为：? 变量名称。
- “工程资源管理器”项：结构化地显示工程中的窗体 ■ 模块 ■ 数据报表(Data Report) 等内容。
- “属性窗口”项：显示在工程中得到焦点的控件属性。你可以在属性窗口中改变焦点控件的属性值。
- “数据视图”项：显示数据视图窗口。关于数据视图我们将在第三章用 Visual Basic 开发数据库中进行详细地讨论。

(3) “工程”菜单

- “添加窗体”项：在当前工程中添加一个窗体，添加的窗体类型为 SDI(单文档界面)。
- “添加 MDI 窗体”项：在当前工程中添加一个 MDI(多文档界面)窗体。在一个工程中只允许有一个 MDI 窗体。
- “添加模块”项：在当前的工程中添加一个模块(Module)。在用 Visual Basic 建立一个大的项目时，可能会有多次使用的代码。因此，通常要在自己的模块中存储那些要重复使用的过程和函数，而不让它们与窗体关联。
- “添加类模块”项：在当前的工程中添加一个类模块(Class)。Visual Basic 是一种面向对象的可视化开发工具，它完全支持对象封装的概念。
- “添加用户控件”项：在当前的工程中添加用户自定义的控件。
- “添加 Data Environment”项：在当前的工程中添加一个数据环境(Data Environment)。关于数据环境我们将在第三章用 Visual Basic 开发数据库中进行详细地讨论。
- “添加 DHTML Page”项：在当前的工程中添加动态的 HTML 页。
- “添加 Data Report”项：在当前的工程中添加一张数据报表(Data Report)。关于数据报

表我们将在第三章用 Visual Basic 开发数据库中进行详细地讨论。

- “添加文件”项：在当前的工程中添加一个外部文件，此文件应该是以 .frm; .ctl; .pag; .dsr; .bas; .cls; .res 结尾的文件类型之一。
- “引用”项：允许选择另外的应用程序的对象，通过设置对那个应用程序对象库的引用，可以在代码中使用它。引用窗口如图 1.4 所示。

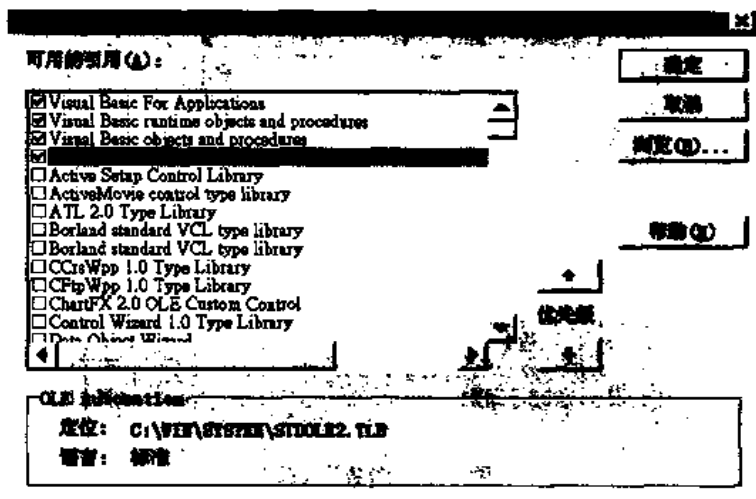


图 1.4 引用窗口

- “部件”项：添加一个控件、设计器或可插入的对象，部件窗口如图 1.5 所示。

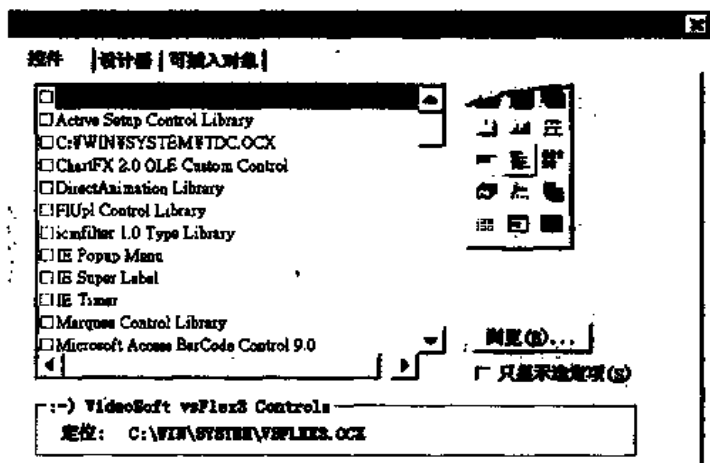


图 1.5 部件窗口

- “工程属性”项：在工程属性窗口(如图 1.6 所示)的通用选项卡中包括以下几个主要部分：(a)工程类型。可以将工程类型改成另一种工程类型，当改变一个工程类型时，所有与它相关的部分将会自动修改。如果某个项目在新的工程类型中是无效的，则它仍然是工程的一部分，但是就不能添加此工程的新的实例。如果当前工程中有某些属性在新工程中无效，则会收到一个信息，建议不要做此改变。(b)启动对象。设定当前工程中的某个窗体或 Sub/Main 最先执行。(c)工程名称。在 Windows 注册区及“对象浏览

器”中确认的部件,名称是否唯一是很重要的。(d)帮助文件名。显示此工程的帮助文件名称。(e)工程帮助上下文标识符。列出帮助主题代码。(f)工程描述。设定工程对象浏览器底部的显示在帮助窗格帮助文本。(g) ActiveX 控件升级。如果从您上次打开该工程后,您又在机器上安装了 ActiveX 控件的新版本,那么在选中 ActiveX 控件后,工程自动更新它们。如果有可用的 ActiveX 控件的新版本而此选项却没有选中,则会出现一个对话框提示您更新控件,该选项的默认值是选中的。

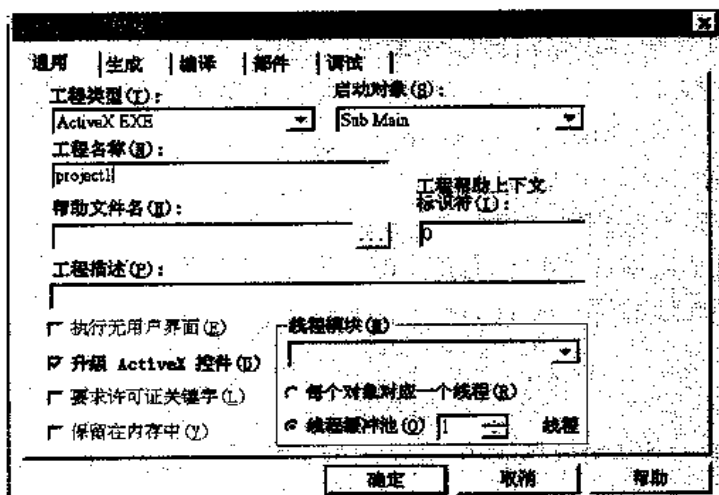


图 1.6 工程属性窗口

(4) “调试”菜单

- “逐语句”项：单步执行调试程序,快捷键为 F8。
- “逐过程”项：逐过程地执行调试程序。
- “添加监视”项：在程序调试中,用来输入监视表达式。此表达式可以是一个变量、属性、函数调用或是其他任何正确的 Basic 表达式。监视表达式会在每次进入中断模式时,或在执行立即窗口中的每个语句后,自动在监视窗口中更新。添加监视窗口如图 1.7 所示。

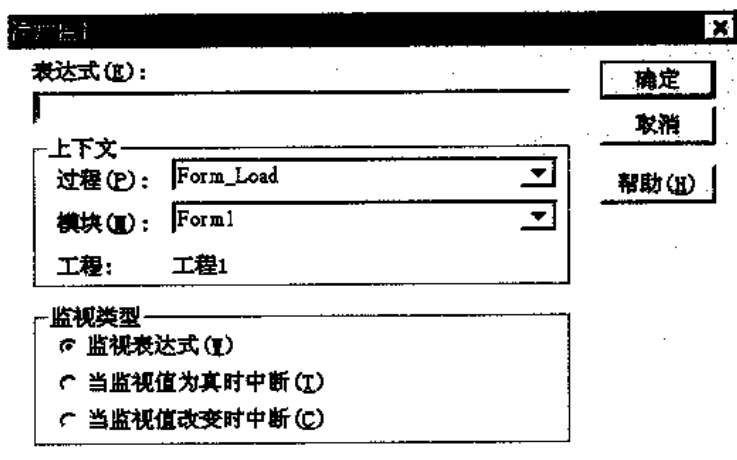


图 1.7 添加监视窗口

- “编辑监视”项：用来删除或编辑监视表达式的内容或类型。
- “切换断点”项：在光标的所处位置上加入断点。

(5) “工具”菜单

- “菜单编辑器”项：用来设计应用程序的菜单，Visual Basic 支持多级菜单。
- “选项”项：显示“选项”对话框，可以从中选取选项卡来设置 Visual Basic 编程环境的属性。

3. Visual Basic 的工具箱

工具箱包含有开发应用程序用的各种控件，在图 1.2 所示的 Visual Basic 集成环境中，工具箱位于屏幕的左边，工具条的下边。工具箱如图 1.8 所示，如果你在集成环境中看到的工具箱里控件与图 1.8 不尽相同，你可以往工具箱中添加或删除控件。有多少种控件可用，取决于用户所使用的 Visual Basic 的版本以及用户是否为 Visual Basic 增加了其它工具。图 1.8 中显示的是在 Visual Basic 中最为常用的那些控件，接下来我们就要对这些控件进行介绍。

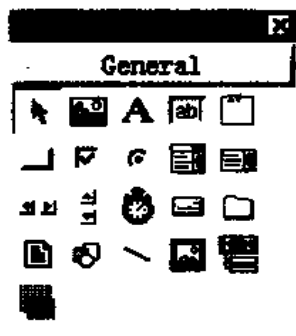


图 1.8 Visual Basic 工具箱

(1) 指针

这是工具箱中唯一不绘制控件的项。在选定指针后只能改变窗体中绘制的控件的大小，或移动这些控件。

(2) 图片框(PictureBox)

显示图形图像(装饰或者活动图片)，该控件作为接受来自图形方法的输出容器，或作为其它控件的容器。PictureBox 控件可以显示来自位图、图标或者元文件，以及来自增强的元文件、JPEG 或 GIF 文件的图形。如果控件不足以显示整幅图像，则裁剪图像以适应控件的大小。

(3) A 标签(Label)

允许有不被修改的文本，例如一个图形下的标题。

(4) 文本框(TextBox)

保存可以输入或修改的文本。TextBox 控件有时也称作编辑字段或者编辑控件，显示设计时输入的、用户输入的或运行时在代码中赋予控件的信息。

(5) 框架(Frame)

允许从图形方面或在功能上对控件分组。为了将控件分组，首先要绘制框架，然后在框架中画出控件。Frame 控件为控件提供可标识的分组。Frame 可以在功能上进一步分割一个窗体。例如，把 OptionButton 控件分成几组。

(6) 命令按钮(CommandButton)

创建按钮，选择它来执行某项命令。CommandButton 控件可以开始、中断或者结束一个进程。选取这个控件后，CommandButton 显示按下的形状，所以有时也称之为下压按钮。

(7) 复选框(CheckBox)

创建一个复选框，用它很容易指出某事件的真假，有多个选择时，也可用它显示这些选择。选择 CheckBox 控件后，该控件将显示×，而清除 CheckBox 控件后，×消失。该控件可用来提供 True/False 或者 Yes/No 选项。组中可以使用 CheckBox 控件显示多项选择，从而可选择

其中的一项或多项。也可以通过对 Value 属性编程设置 CheckBox 的值。

(8) 选项按钮(OptionButton)

允许显示多个选项,但只能从中选择一项。在选项组中用 OptionButton 显示选项,用户只能选择其中的一项。在 Frame 控件、PictureBox 控件或者窗体这样的容器中绘制 OptionButton 控件,就可以把这些控件分组。

(9) 组合框(ComboBox)

允许绘制一个组合列表框和文本框。使用时可从下拉列表中选择一项,也可在文本框中输入值。ComboBox 控件将 TextBox 控件和 ListBox 控件的特性结合在一起,既可以在控件的文本框部分输入信息,也可以在控件的列表框部分选择一项。

(10) 列表框(ListBox)

用于显示项的列表,可从这些项中选择一项。如果包含的项太多而无法一次显示出来,则可滚动列表框。ListBox 控件显示项目列表,从其中可以选择一项或多项。如果项目总数超过了可显示的项目数,就自动在 ListBox 控件上添加滚动条。

(11) 水平滚动条(HScrollBar)

水平滚动条是一个图形工具,可快速移动很长的列表或大量信息,可在标尺上指示当前位置,可以作为输入设备,或作为速度或数量的指示器。

(12) 垂直滚动条(VScrollBar)

与水平滚动条类似,垂直滚动条是一个图形工具,它可以快速引导一个很长的列表或大量信息,可以在标尺上指示当前位置,可以作为输入设备,或作为速度和数量的指示器。

(13) 定时器(Timer)

在指定的时间间隔内产生定时器事件。该控件在运行时不可见。通过引发 Timer 事件,Timer 控件可以有规律地隔一段时间执行一次代码。

(14) 驱动器列表框(DriveListBox)

显示有效的磁盘驱动器。在运行时,由于有 DriveListBox 控件,所以可选择一个有效的磁盘驱动器。该控件用来显示用户系统中所有有效磁盘驱动器的列表。可以创建对话框,通过它从任一可用驱动器的磁盘文件列表中打开文件。

(15) 目录列表框(DirListBox)

显示目录和路径。在运行时,DirListBox 控件显示目录和路径。这个控件可以显示分层的目录列表。例如,可以创建对话框,在所有可用目录中,从文件列表打开一个文件。

(16) 文件列表框(FileListBox)

显示文件列表。在运行时,在 Path 属性指定的目录中,FileListBox 控件将文件定位并列举出来。该控件用来显示所选择文件类型的文件列表。例如,可以在应用程序中创建对话框,通过它选择一个文件或者一组文件。

(17) 形状(Shape)

在设计时,允许在窗体上绘制多种形状的图形。可在其中选择矩形、圆角矩形、正方形、圆角正方形、椭圆形或圆形。

(18) 行(Line)

在设计时用来在窗体上绘制各种样式的线。

(19) 图像(Image)

在窗体上显示位图、图标、或元文件中的图形图像。Image 控件中显示的图像可以仅是装饰性的,与 PictureBox 相比,它使用的资源要少一些。因为 Image 控件使用较少的系统资源,所以重画起来比 PictureBox 控件要快,但是它只支持 PictureBox 控件的一部分属性、事件和方法。

(20) 数据(Data)

通过窗体上被绑定的控件来访问数据库中的数据。Data 控件允许从一个记录移动到另一个记录,并显示和操纵来自被连结的控件的记录的数据。如果没有 Data 控件或等价的数据源控件,比如 RemoteData 控件,窗体上的被连结的控件不能自动访问数据。

(21) OLE

允许把其它应用程序的对象链接和嵌入到 Visual Basic 应用程序中。

1.1.4 Visual Basic 的联机帮助系统

Visual Basic 为开发人员提供了功能相当强大的在线帮助。笔者现在假定用户已经在系统中安装了联机帮助系统,事实上,作为一个 Visual Basic 的使用者,你也应该这样做。笔者很难想象开发人员在—个没有联机帮助系统的环境下去开发 Visual Basic 的应用程序。联机帮助系统包含了几乎所有 Visual Basic 语言参考手册中的信息内容,另外还包含着数以百计的程序示例。联机帮助系统有一个非常有用的特性:使用上下文相关的帮助信息,这就是说用户在任何时候都可以按下 F1 键就能直接得到所需要的信息。图 1.9 就显示了 Visual Basic 的联机帮助系统。

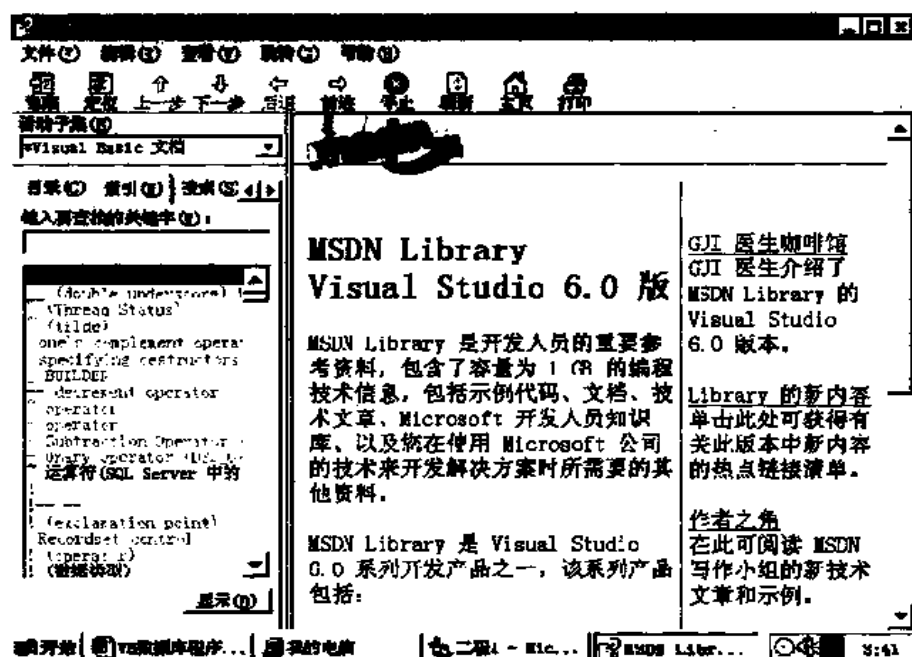


图 1.9 Visual Basic 的联机帮助系统

在如图 1.9 所示的窗口上,用户可以查看完整的帮助主题索引。当把所要查找的内容输入,按显示按钮后,用户就可以得到关于此内容完整的帮助信息。Visual Basic 提供的联机帮助

系统是一个超文本系统。这就意味着用户可以从本帮助内容出发转到其它用户感兴趣的内容上去。

在结束本节之前,总结一下 Visual Basic 开发一个应用程序时所应该采取的步骤:

- (1) 建立一个工程(Project)。
- (2) 在工程下定制用户所希望的窗体(Form)。
- (3) 确定窗体的控件都识别哪些事件(Event)。

(4) 为这些事件编写相应的事件过程(Produce),如果需要则应该编写子过程来满足事件过程工作。

1.2 Visual Basic 编程初步

现在,相信读者已经对使用 Visual Basic 开发 Windows 下的应用程序有了初步的认识。但为了使能够尽快开始编写自己的 Visual Basic 应用程序,读者必须了解构成 Visual Basic 中的更为卓越的编程语言。

Visual Basic 编程语言是从最早的 Basic 语言一步一步演化发展起来的。现在,已经发展到了 6.0 版本,无论从形式上还是从内容上来看,它与最早期的那种解释型的 Basic 语言已经有了相当大的差异。如果读者对 QBasic、C、Pascal 或 Quick Basic 有所了解的话,那么在读本章的过程中就会感到简单易学,学习起来当如轻车熟路。如果读者对以上这些语言都不太了解的话,也不要着急,只要你仔细地阅读本章的内容,很快你就会掌握 Visual Basic 的编程语言。

1.2.1 Visual Basic 程序剖析

Visual Basic 代码的编写与传统意义上的结构化程序设计有着本质上的区别,这些本质上的差异会使那些对结构化程序设计有着丰富经验的程序员感到极不习惯。所以,在开始讲述编程语言之前,我们必须对 Visual Basic 的程序机理有一个极为清晰的了解,这样才能在开始编写 Visual Basic 的应用程序时不会感到无所适从。

为 Visual Basic 应用程序编写的所有代码都是用来响应事件的,也就是说 Visual Basic 程序是事件驱动的。所谓事件驱动就是说应用程序都是由一些独立的代码段组成,这些代码段从结构上来看并不存在着优先问题,它的执行与否完全是由用户的操作来决定的。当用户进行了某一种操作(驱动了某种事件)时,Visual Basic 内置的消息机制才会把这种操作信息发送给系统,然后由系统找到相对应的事件并执行为本事件所编写的代码。如果读者把在一些旧的编程语言中养成的习惯强加到 Visual Basic 中去,将会遇到意想不到地麻烦。所以,读者应该在使用 Visual Basic 的一开始就有一种事件驱动的概念,这样才能在接下来的学习过程中渐入佳境。

Visual Basic 的编程总是在代码(Code)窗口中进行的,图 1.10 显示了 Visual Basic 的代码窗口。代码窗口左边的下拉框显示了当前工程中的对象,用户可以从中学选取某一对象;右边的下拉框显示了所选取对象的各种事件,当用户选定某一事件后就可以为该事件编写代码了。

要打开代码窗口,用户可以在无论什么时候双击控件或窗体。同样,用户可以从“视图”菜单中选取“代码窗口”菜单项,或者按下 F7 键来打开代码窗口。当处理多窗体或较为复杂的程序代码时,用户还可以打开多个代码窗口。

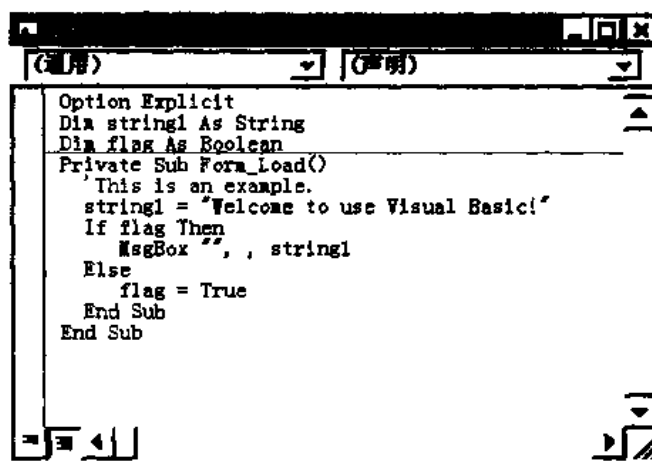


图 1.10 Visual Basic 的代码窗口

1.2.2 Visual Basic 语言介绍

1. 常量和变量

(1) 常量

对于那些使用过其他任何一种编程语言的读者而言,常量不应该是一个陌生的名词。常量的引入使得程序有更好的可读性并且更加易于调试。在程序的编写过程中更为常见的是,为一个常数命名。如果在一段程序代码中要大量地用到数值 3.1415926,那么定义一个常数 $\text{Pie} = 3.1415926$,然后在接下来凡是要用到 3.1415926 的代码中都用 Pie 来代替,这种做法显然要比在程序中敲上无数多个繁琐的数字简洁明了得多。

用户可以使用关键字 `Const` 来创建一个常数,关键字 `Const` 后面紧跟着常数名,一个等号,然后是一个数值。例如:

```
Const Pie=3.1415926
```

在 Visual Basic 当中,习惯上对常量全部使用大小写字母混写的方式,尽管这并不是必须的。另外,对常量的引用也不必拘泥于字母的大小写。

(2) 变量

在 Visual Basic 中,无论何时使用一个变量,都会在计算机的内存中为它开辟一个区域来存放它的信息。Visual Basic 的变量名最多可以由 255 个字符组成,其首写字符必须为一个字母,后面可以包含任何字母、数字和下划线的组合。在变量名中,对字母的大小写并不敏感,例如 `VAR1` 和 `var1` 是同一个变量。

① 变量声明

在使用变量之前先进行变量说明,这是一个很好的编程习惯。虽然 Visual Basic 提供了不预先声明就直接使用变量的功能,但是笔者并不建议读者这样做,这样一来将会导致程序的混乱性和可读性的降低。要保证在使用变量之前预先声明,就应该在声明变量之前键入 `Option Explicit`。这样,在接下来的代码编写过程当中,如果用户键入了一个预先没有声明的变量,Visual Basic 就会给出错误信息告诉用户键入的变量名没有预先声明。

Visual Basic 对变量的声明是使用 `Dim` 语句来完成的。由于单字符的标识符很容易被人

忽略,所以使用 Dim 语句能增强程序的可读性。变量声明的语法由如下的形式实现:

```
Dim 变量名 As 变量类型
```

下面给出一个变量声明的例子。

```
Option Explicit
Dim op1 As String
Dim op2 As String
Private Sub Form_Load()
    Dim year As String
    Dim month As String
    Dim date As String
    Dim count As Integer
    Dim flag As Boolean
End Sub
```

用户还可以在一行之中组合多个声明,例如:

```
Dim year As String, month As String, date As String
```

但是,如下的声明却是错误的:

```
Dim year, month, date As String
```

最后,如果读者想要使用可变数据类型的变量,可以只使用 Dim 语句,而不使用 As 子句或标识符:

```
Dim Notype_var 'make Notype_var have the variant data type
```

用户还可以这样使用:

```
Dim Notype_var As Variant 'more explicit easier to read
```

② 变量类型

Visual Basic 可以处理十多种标准的变量类型,另外还可以定义用户自己的变量类型。接下来我们简单介绍一下几种常用的变量类型:

- 字符串型:字符串型变量用以存放字符。
- 整型:整型变量用以存放较小的整数值(从-32768 到 32767)。
- 长整型:长整型变量能够存放从-2147483648 到+2147483647 间的整数。
- 单精度型:单精度型变量用以存放数字的近似值,可以是小数,但只能精确到 7 位数。
- 双精度型:双精度型变量能够存储 16 位精度的数值。
- 货币型:它是为了避免由二进制数转换到十进制小数时产生的某些问题而设计的,货币型变量在小数点右边可以有 4 个数字,而在小数点左边有 14 位数字。
- 日期型:日期型变量能够方便地存储日期和时间信息,其范围从 January 1,100 的午夜到 December 31,9999 的午夜。如果用户没有在日期型变量中加入时间,Visual Basic 会假定它为午夜。
- 字节型:字节型变量是 Visual Basic 4.0 中新加的类型,其值在 0 到 255 整数范围内。
- Boolean 型:当变量只有 True 和 False 两个选择时,可以使用 Boolean 型变量。

(3) 变量的作用域

当希望能使变量在程序的某一部分至另一部分为有效时,开发人员应该为变量指定一个

作用域。在 Visual Basic 中,实现此问题的方法是在过程内分离变量,除非有特殊的措施,否则在一个过程内改变变量的值不会影响到另一个过程中同名变量的值。也就是说,变量对于过程而言是局部的。

2. 数字

(1) 数值

在任何编程语言的代码书写过程中,数字随处可见。在 Visual Basic 中,如果要给一个变量赋一个数值,应该在赋值语句的右侧放置该数值。如果给一个整型变量赋予一个带有小数点的数值,它将被自动四舍五入。如果给一个特定的变量赋予一个超出其限制范围的较大的数值时,Visual Basic 在运行期间会显示一条错误信息。如果用户要进行大数值的运算时,用户就会发现 Visual Basic 在大数值的显现过程中使用的是科学记数法。

(2) 数值的运算

有了数值,数值之间的运算就成为理所当然的事。Visual Basic 为用户提供了许多种算术运算符来处理数值之间的运算。以下给出七种最常用的算术运算符:

- +: 进行数值之间的加法运算。
- -: 进行数值之间的减法运算。
- *: 进行数值之间的乘法运算。
- /: 进行数值之间的除法运算。
- ^: 幂运算。
- \: 整除运算。
- Mod: 整除后取余数的运算。

3. 字符串

字符串变量是保持字符信息的基本数据类型。字符串变量可以包含大约 65535 字节 (64K),可以为定长,也可以为变长,每个字节包含一个字符。定长字符串用一个具体长度来说明。例如,定义一个 10 个字符长度的字符串变量:

```
Dim TenString as String * 10
```

变长字符串可以有不超过 64K 的任何长度,减少存储开销。

在字符串的使用过程中,更为重要的一点是字符串的连接问题。为了把两个字符串连接在一起,可以使用一个加号(+)来实现,在使用加号来连接字符串时,要注意连接时的顺序问题。下面是一个字符串连接的例子:

```
Dim NameVar as String
Dim EventVar as String
Dim ObjectVar as String
Dim Sentence as String
NameVar = "We"
EventVar = "use"
ObjectVar = "Visual Basic."
```

```
Sentence = NameVar + EventVar + ObjectVar 'Sentence=We use Visual Basic.
```

4. 简单的语句介绍

(1) 赋值语句

为变量赋值是 Visual Basic 中最为常见的任务之一。Visual Basic 使用等号来实现赋值语句,例如:

```
NameVar = "We"
```

当然,用户也可以使用原来的 Let 方法:

```
Let NameVar = "We"
```

同为变量赋值相对应,给属性赋值在 Visual Basic 中也是经常要用到的。如果用户要为一个 Visual Basic 的对象设置或改变其属性,那么在对象名后紧跟一点号,然后书写属性名,并把它们放在等号的左边,把要赋予的值放在等号右边。这样,用户就可以在程序中根据自己的需要随心所欲地通过代码来改变属性设置。例如,如果用户希望在程序运行中改变一个名为 Command1 的命令按钮的标题,可以在事件过程中添加如下行:

```
Command1.Caption = "New Caption"
```

(2) 注释语句

注释语句的书写在任何的程序编写当中都是非常重要的。读者可千万不要忽略了注释语句的书写,一个好的程序员往往会为自己所编写的代码附上大量的注释。在开发一个大的项目过程中,如果没有注释语句,那么就会使得他人接手你的工作变得几乎不可能。

在 Visual Basic 中,有两种方法来说明一个注释语句。现在最常用的是使用一个单引号('),在单引号的后面加上所要注释的话。例如:

```
Private Sub Form1_Load()  
    ' This is a commend line  
End Sub
```

用户还可以使用老式的 Rem 关键字。程序会从该 Rem 语句下面的第一条可执行语句继续执行。如果在其它语句行后使用 Rem 关键字,则必须使用冒号(:)与语句隔开。例如:

```
Private Sub Form_Load()  
    Dim I  
    I = 4 ' This is a commend line  
    I = 6 ;Rem This is a commend line  
End Sub
```

(3) End 语句

End 语句用来结束一个过程或块,它有以下几种形式:

- End: 停止执行。不是必要的,可以放在过程中的任何位置关闭代码执行、关闭以 Open 语句打开的文件并清除变量。
- End Function: 必要的,用于结束一个 Function 语句。
- End If: 必要的,用于结束一个 If...Then...Else 语句块。
- End Property: 必要的,用于结束一个 Property Let、Property Get 或 Property Set 过程。
- End Select: 必要的,用于结束一个 Select Case 语句。
- End Sub: 必要的,用于结束一个 Sub 语句。
- End Type: 必要的,用于结束一个用户定义类型的定义(Type 语句)。
- End With: 必要的,用于结束一个 With 语句。

5. 函数和过程

我们知道,用 Visual Basic 开发应用程序的主要任务是给各种事件编写相应的代码(因为 Visual Basic 是事件驱动的)。系统为控件和窗体所提供的种种事件事实上就是一个一个过程(函数),如图 1.11 所示,只是在用户开始着手编写代码之前这些过程(函数)是以模版的形式提供给用户的,它并没有真正的程序代码。

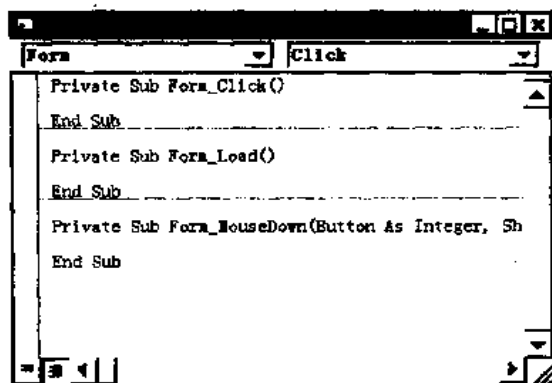


图 1.11 代码窗口中的过程模板

图 1.11 所示的代码窗口中含有三个过程的模板,用户如果加入其它的过程,可以从左边的下拉框中选取。这三个过程分别为 `Form_Load()`, `Form_Click()`, `Form_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As Single)`。与之相应的事件分别为窗体载入,窗体单击和在窗体上按下鼠标。如果用户在 `Form_Click()` 过程中加入代码,那么在程序运行期间,当用户在窗体上单击鼠标左键时就会执行这段代码。

如果某一事件过程过于冗长,那么从程序可读性和调试上来说,应当考虑把它分成几个子程序。子程序包括函数和过程两种。函数和过程最重要的区别是函数有返回值而过程没有。用户设计子程序时应该根据需要来选择使用函数还是过程。

(1) 函数

一个完整的函数示例如下:

```
Private Function Foo( X As Integer ) As Integer
    Dim Y As Integer
    Y = X+1
    If Y>0 Then
        Foo = Y
    Else
        Foo = X
    End If
End Function
```

函数 `Foo` 定义中的 `X` 是传入函数的参数,当然你也可以传入多个参数,每个参数定义之间用逗号相隔,到底要使用多少个参数是根据具体的需要来定的。我们知道,函数的参数传递有两种类型:值传递和地址传递。如果没有特别的说明,在 Visual Basic 中的函数参数传递为地址传递。这就意味着我们在使用参数时要特别地小心,在函数体内,对参数所做的任何改变

都会影响原始变量。如果一定要使参数为按值传送,那么就应该在参数的定义前加上 ByVal。

最后,如果函数声明为 Private,则可以通过连接它的窗体中的代码看到。如果声明为 Public,项目的其他部分可以使用窗体名和函数名来调用此函数。在这一点上过程与之类似。

(3) 过程

正如上述的那样,函数有返回值而过程没有。在实际的程序开发中,我们用得更多的是那些无需有返回值的子程序,这时候我们就应该用到过程(Sub)。想要调用过程有两种方式,一种是直接调用,在直接调用时,不允许在过程名后面加括号,就算是此过程含有参数也应该这样做。另一种做法是使用 Call 关键字,在使用 Call 关键字时,必须对参数表使用括号,下面是一个使用 Call 关键字调用过程的例子:

```
Call Temp_Procedure(Argument1,Argument2,Argument3,...)
```

如果不用 Call 关键字来调用,则应该如下:

```
Temp_Procedure Argument1,Argument2,Argument3,...
```

当然,在过程调用时的参数传送也和函数的参数传送一样,如果没有特别的说明,都应该按地址传送来对待。

1.3 Visual Basic 程序控制流

1.3.1 条件判定

条件判定在 Visual Basic 的程序设计中是极为常用的,它根据判定值的真或者假来选择程序分支。条件判定语句的标准形式如下:

```
If condition Then
    Do Something
Else
    Do Otherthing
End If
```

在程序执行中,条件判定语句根据 condition 的值来选择程序分支。如果 condition 的值为真(True),则执行紧跟着 Then 之后的语句 Do Something;否则,如果 condition 的值为假(False),就执行跟在 Else 之后的语句 Do Otherthing。其中 Else 分支是可选的。与其他的程序设计语言相同,Visual Basic 中的条件判定语句也允许嵌套。

1.3.2 循环

循环也就是重复操作,它在程序执行过程中重复固定或不固定次数的操作。只要程序的执行进入循环体中,它就一直继续,直到达到预定的目标时才退出循环体,或者一直继续,直到最终改变了某种初始条件时才退出循环体。在 Visual Basic 中一共有三种循环形式,根据重复次数的固定或不固定,可以把循环分成两类:确定循环和不确定循环。

1. 确定循环

确定循环就是在执行过程中循环的次数是确定的。Visual Basic 中只有一种循环形式是确定循环,那就是 For-Next 循环。下面是一个 For-Next 循环的例子:

```
For I=1 To 20
```

```
MsgBox " ", , "This is an example."
```

```
Next I
```

在上述的例子中,I 是一个循环变量,它从初始值开始执行(在本例子中初始值为 1),到终止值(在本例子中为 20)为止结束循环。每执行一次循环体,循环变量就自动加一。一般来说,循环变量应该为整数,这样不仅能使程序清晰易读,而且能够让 Visual Basic 用尽可能少的时间来加速循环。循环体就是位于 For 和 Next 之间的语句,在实际的代码编写中,用户可以根据自己的需要来书写循环体的内容。

2. 不确定循环

不确定次数的循环在程序代码的编写过程中也是相当常用的。顾名思义,不确定循环是循环次数并不预先固定的循环。它并不是用循环变量来控制循环体执行的次数,而是根据某种初始条件的改变来控制循环的次数。

(1) Do 循环

用 Do 循环重复执行一语句块,且重复次数不定。Do...Loop 语句有几种演变形式,但每种都计算数值条件以决定是否继续执行。如同在 If...Then 中,condition 必须是一个数值或者值为 True(非零)或 False(零)的表达式。

- Do While condition

```
Statements
```

```
Loop
```

当 Visual Basic 执行这个 Do 循环时会首先测试 condition。如果 condition 为 False,则跳过所有语句。如果 condition 为 True,则 Visual Basic 执行语句,然后退回到 Do While 语句再测试条件。因此,只要 condition 为 True,循环可以随意执行多少次。如果 condition 一开始便为 False,则不会执行语句。

- Do

```
Statements
```

```
Loop While condition
```

Do...Loop 语句的一种演变形式是先执行语句,然后在每次执行后测试 condition。这种形式保证 statements 至少执行一次。

其它两种演变形式类似于前两个,所不同的是,只要 condition 为 False 而不是 True,它们就执行循环。

- Do Until condition

```
Statements
```

```
Loop
```

- Do

```
Statements
```

```
Loop Until condition
```

(2) While 循环

在 Visual Basic 中还有另一种循环形式,这是为了与先前的 Basic 版本兼容。

```
While condition
```

```
Statements
```

Wend

1.3.3 跳转

与大多数的编程语言类似,为了同先前的版本保持兼容,在 Visual Basic 中保留了跳转(Goto)语句。虽然笔者不建议读者在编写代码时使用 Goto 语句,过多的使用 Goto 语句会使程序的结构变得极其混乱。

要在 Visual Basic 中使用 Goto 语句,必须对行加标号,标号必须以字母开始,并用冒号结束,而且必须在第一列中(显然,应当尽可能使用描述性的标号)。例如:

```
ErrorInput:
    ' We can goto here.
For I=1 To 5
    ImportData = InputBox("Data Import", "Enter Data,Ok to end", "")
    If ImportData = "Ok" Then
        Goto ErrorInput
    Else
        Exit Sub
    End If
Next I
```

1.4 Visual Basic 的内部函数

Visual Basic 提供了成百上千的内部函数,这些内部函数考虑到了开发者在开发程序过程中会遇到的几乎所有问题,可以说 Visual Basic 的内部函数是构成 Visual Basic 强大功能的重要支柱之一,下面我们就分类对那些在程序编写过程中经常会遇到的内部函数进行介绍。如果读者想要了解更多的内部函数,可以查看 Visual Basic 的联机帮助文档。

1.4.1 字符处理函数

1. Mid 函数

Mid 函数返回字符串中指定数量的字符。它的使用语法如下:

Mid(string,start[,length])

参数说明:

- String: 这是一个必要的参数,它为一个字符串的表达式。如果 string 中包含 Null,那么它将返回 Null。
- Start: 必要参数,它指出了在字符串中将要被取出部分的字符首位置。如果 start 超过 string 的字符数,Mid 则返回零长度字符串("")。
- Length: 这是一个可选的参数,它指出了要返回的字符数。如果省略或者 length 超过了文本的字符数,将返回字符串中从 start 到尾端的所有字符。

2. Len 函数

Len 函数返回相应字符串的长度。它的使用语法如下:

Len(string | varname) “|”表示或者。

参数说明:

- String: 任何有效的字符串表达式。如果 string 包含 Null, 会返回 Null。
- Varname: 任何有效的变量名称。如果 varname 包含 Null, 会返回 Null。如果 varname 是可变的数据类型, 那么 Len 会把它视为 String 类型, 并且总是返回其所包含的字符数。

3. Left 函数

Left 函数包含字符串中从左边算起指定数量的字符。它的语法如下:

Left(string, length)

参数说明:

- String: 任何有效的字符串表达式。如果 string 包含 Null, 会返回 Null。
- Length: 必要参数, 它指出将返回多少个字符。如果为 0, 返回零长度字符串("")。如果大于或等于 string 的字符数, 则返回整个字符串。

4. Right 函数

Right 函数与 Left 函数类似, 但它返回的是包含字符串中从左边算起指定数量的字符。它的语法如下:

Right(string, length)

参数说明:

- String: 任何有效的字符串表达式。如果 string 包含 Null, 会返回 Null。
- Length: 必要参数, 它指出将返回多少个字符。如果为 0, 返回零长度字符串("")。

5. LTrim, RTrim 与 Trim 函数

返回一个 String 类型的数值, 其中包含所指定字符串的拷贝, 没有前导空白(LTrim)、尾随空白(RTrim)或前导和尾随空白(Trim)。它的语法如下:

LTrim(string); RTrim(string); Trim(string)

参数说明:

- String: 任何有效的字符串表达式。如果 string 包含 Null, 会返回 Null。

6. InStr 函数

InStr 函数返回一字符串在另一字符串中最先出现的位置。它的语法如下:

InStr([start,] string1, string2[, compare])

参数说明:

- Start: 这是一个可选参数, 它为数值表达式, 设置每次搜索的起点。如果省略, 将从第一个字符的位置开始。如果 start 包含 Null, 将发生错误。如果指定了 compare 参数, 则一定要有 start 参数。
- String1: 必要参数。接受搜索的字符串表达式。
- String2: 必要参数。被搜索的字符串表达式。
- Compare: 可选参数。指定字符串比较。如果 compare 是 Null, 将发生错误。如果省略 compare, Option Compare 的设置将决定比较的类型。

7. StrComp 函数

StrComp 函数返回一个整数值, 它为两个字符串比较的结果。它的语法如下:

StrComp(string1, string2[, compare])

参数说明:

- String1: 必要参数。表示任何有效的字符串表达式。
- String2: 必要参数。表示任何有效的字符串表达式。
- Compare: 可选参数。它指定字符串比较的类型。如果 compare 参数是 Null, 将发生错误。如果省略 compare, Option Compare 的设置将决定比较的类型。

1.4.2 数学函数

1. Abs 函数

Abs 函数返回参数的绝对值, 其类型和参数相同。它的语法如下:

Abs(number)

参数说明:

- Number: 这是一个必要的参数, number 参数是任何有效的数值表达式, 如果在 number 参数中包含 Null, 则返回 Null, 如果 number 是未初始化的变量, 则返回 0。

2. Exp 函数

Exp 函数返回一个双精度的数值, 这个值是指定 e (自然对数的底) 的某次方。它的语法如下:

Exp(number)

参数说明:

- Number: 必要参数。参数 number 是双精度型或任何有效的数值表达式。如果 number 的值超过 709.782712893, 则会导致错误发生。常数 e 的值大约是 2.718282。

3. Log 函数

Log 函数返回一个双精度的数值, 这个值指定参数的自然对数值。它的语法如下:

Log(number)

参数说明:

- Number: 必要参数。参数 number 是双精度型或任何有效的数值表达式。

4. Sgn 函数

Sgn 函数返回一个整数值, 这个返回的整数值指出参数的正负号。如果参数值大于 0, Sgn 函数返回 1; 如果参数值小于 0, Sgn 函数返回 -1; 如果参数值等于 0, Sgn 函数返回 0。它的语法如下:

Sgn(number)

参数说明:

- Number: 必要参数。参数 number 是任何有效的数值表达式。

5. Int 函数

Int 函数返回参数的整数部分。如果参数为负数, 则 Int 返回小于或等于 number 的第一个负整数。它的语法如下:

Int(number)

参数说明:

- Number: 必要参数。参数 number 是任何有效的数值表达式。

6. Sqr 函数

Sqr 函数返回一个双精度的数值, 它给出了参数的平方根。它的语法如下:

Sqr(number)

参数说明:

- Number: 必要参数。参数 number 是双精度型数值或任何有效的大于或等于 0 的数值表达式。

7. Rnd 函数

Rnd 函数返回一个包含随机数值的单精度数,这个单精度数的取值范围为小于 1 并且大于等于 0。它的语法如下:

Rnd([number])

参数说明:

- Number: 可选参数。参数 number 是单精度型数值或任何有效的数值表达式。number 的值决定了 Rnd 生成随机数的方式。对最初给定的种子都会生成相同的数列,因为每一次调用 Rnd 函数都用数列中的前一个数作为下一个数的种子。所以从严格的意义上来说,Rnd 函数只能产生伪随机数。要想产生真正的随机数,就应该在使用 Rnd 函数之前先调用无参数的 Randomize 语句来初始化随机数生成器。

8. 三角函数

三角函数返回一个指定参数的三角值。三角函数主要有 sin(),cos(),tan()等,它的语法如下:

sin(number);cos(number);tan(number)

参数说明:

- Number: 必要参数。number 参数是双精度型数值或任何有效的数值表达式,表示一个以弧度为单位的角。

1.4.3 日期和时间函数

1. Time 函数

Time 函数是一个无参函数,它返回系统当前的时间。它的语法为:

Time

2. Date 函数

Date 函数是一个无参函数,它返回系统当前的日期。它的语法为:

Date

3. Now 函数

Now 函数是一个无参函数,它返回系统当前的日期与时间。它的语法为:

Now

4. Timer 函数

Timer 函数是一个无参函数,它返回一个单精度数值,代表从午夜开始到现在经过的秒数。它的语法为:

Timer

5. Hour 函数

Hour 函数返回一个 0 到 23 之间的整数,表示一天之中的某一钟点。它的语法为:

Hour(time)

参数说明:

- Time: 必要参数。time 参数可以是任何能够表示时刻的数据类型、数值表达式、字符串表达式或它们的组合。如果 time 包含 Null, 则返回 Null。

6. Minute 函数

Minute 函数返回一个 0 到 59 之间的整数, 表示一小时中的某分钟。它的语法为:

Minute(time)

参数说明:

- Time: 必要参数。time 参数可以是任何能够表示时刻的数据类型、数值表达式、字符串表达式或它们的组合。如果 time 包含 Null, 则返回 Null。

7. Second 函数

Second 函数返回一个 0 到 59 之间的整数, 表示一分钟中的某秒。它的语法为:

Second(time)

参数说明:

- Time: 必要参数。time 参数可以是任何能够表示时刻的数据类型、数值表达式、字符串表达式或它们的组合。如果 time 包含 Null, 则返回 Null。

第二章 数据库概述

随着现代社会的快速发展,对信息的依赖程度日益增强。作为信息管理的主要工具,数据库技术正在越来越广泛地被应用到社会和生产中的各个方面。数据库技术是管理数据的一种方法,它研究如何组织和存储数据,如何高效地获取和处理数据。根据数据库系统本身所支持的数据模型的特点及所采用的相应数据库系统技术的发展,到目前为止已经经历了三代,即层次网状模型、关系模型和面向对象模型。本章的主要任务是使读者对数据库系统有一个大致的了解,如果读者在这之前已经学过有关数据库方面的知识,那么可以跳过本章内容。

在本章中我们主要讲述以下几部分:

- 数据库系统与模型
- 关系数据库
- SQL 语言

2.1 数据库系统与模型

2.1.1 数据库系统

数据库系统本身是一个较为模糊的概念,它很难用一两句话来将其下一个确切的定义。但是,为了使读者对它有一个在总体上的了解,笔者就试着给出一个关于数据库系统的定义。即数据库系统中的数据库是一个已被规格化和结构化且相互关联的数据集合,这些数据中不存在有害的或无意义的冗余;数据库中数据的组织与存储结构与使用这些数据的程序相互独立;数据库中的数据可同时为多个应用程序服务;数据库中的数据定义、输入、修改和检索等操作均是按一种公用的且可控的方式进行。根据数据库定义和实际应用的具体数据库系统的情况,我们可以认为一个数据库系统实际上是由三个部分组成,它们是数据库、多种应用和数据库管理系统。用户的各种应用通过数据库管理系统和后端的数据库联系,同时后端数据库的数据反馈也是通过数据库管理系统而传递给用户程序的。

一个数据库系统应该具有以下 5 个基本特点:

1. 整体描述性

由于数据库系统是从整体的角度来考虑数据的组织,因此它必须有能力描述反映客观事物及其相互联系的复杂数据模型(关于数据模型,我们将在下一小节中介绍),使用它能够对数据本身及相互间的各种关系进行充分的描述。目前数据库系统提供了三种数据模型,它们是层次网状模型、关系数据模型和对象数据模型。一种类型的数据库系统一般来说只支持一种数据模型。

2. 数据独立性

数据独立性包含两层意思,即数据的物理独立性和数据的逻辑独立性。数据在物理存储设备上的组织结构被称为数据的物理组织,数据在用户或应用程序面前所表现出来的组织结构被称为数据的逻辑组织。所谓数据的物理独立性是指数据物理组织的变化不影响数据的逻辑

组织,因此也就不影响已有的应用程序。数据的逻辑独立性是指当数据的逻辑组织发生变化时,原有应用程序的执行不受影响或影响很小。

3. 数据共享

由于数据库是从整体的角度对数据进行组织的,并且要在保证数据一致性的情形下,使数据库中的数据为尽可能多的用户提供应用服务。所以,这些用户所使用的数据应该是可以共享的。如果数据不能共享,数据库中必然会出现大量的冗余数据,这样一来不仅会造成存储空间的浪费,还会带来数据不一致的隐患。

4. 安全性

数据库的安全控制主要指的是数据库的保密性。并不是每个用户都能够存取数据库中所有数据的,系统管理员和一般工作人员允许掌握的数据操作范围显然是有区别的。数据库系统把用户存取数据的权限分成若干等级,通过对各个用户授予不同的使用权限来确保数据库免遭损害和被非法使用。通常是通过采用口令密码以及数据库中数据访问授权等方法来对使用者操作数据的合法权进行检验,以实现数据库安全性的保护控制。

5. 完整性

所谓完整性包括数据的正确性、有效性和相容性。正确的数据不一定是有效的,数据库系统应提供尽可能多的检验措施,以确保数据库中的数据满足用户的各种约束要求。

数据库系统的核心是数据库管理系统工程 DBMS,在它的控制之下,用户可以建立、使用、修改和维护数据库中的数据。数据库管理系统是建立在操作系统之上的应用软件平台。接下来介绍一下当用户读取数据库中数据时,数据库系统的工作过程,读者从中可以了解到 DBMS 是如何来进行数据库系统的管理工作的。用户访问数据库系统中数据的过程实际上就是用户与 DBMS 进行交互的过程。其具体的步骤如下:

(1) 用户通过应用程序指明它使用的子模式名称,发出数据操作命令,DBMS 通过处理用户的应用程序,接受该操作命令。

(2) DBMS 按照应用程序中的子模式名称,调用相应的子模式,核对该用户的访问权利、操作合法性等,若检查通过则继续执行,否则拒绝执行并报告出错信息。

(3) DBMS 按模式确定子模式中操作所涉及的记录类型,并通过模式到存储模式的映射,找出这些记录类型的相应存储模式。

(4) DBMS 查阅存储模式,确定应从哪个物理文件、存储设备以及调用哪个访问程序去读取所需要的记录。

(5) DBMS 的访问程序找到有关的物理数据地址,向操作系统发出读操作命令。

(6) 操作系统收到 DBMS 发来的命令后,启动系统的输入/输出程序完成读操作,把要读取的数据块送到内存中的系统缓冲区。

(7) DBMS 收到操作系统关于输入/输出操作结束回答后,按模式、子模式的定义,将已读入到系统缓冲区的内容映射为用户程序所要的逻辑记录,并送到用户的工作区中。

(8) DBMS 向应用程序发送反映操作执行结果的状态信息。

(9) 记载 DBMS 系统的工作日志。

(10) 应用程序检查状态信息,如果执行成功,则可以对程序工作区中的数据做正常处理,否则按出错类型决定程序的后续处理。

2.1.2 数据模型

我们知道,所谓数据是信息的一种表达形式。那些成千上万的数据有没有共性呢,如果有的话,我们应该采用什么样的方式来描述它们的共性呢。由此,我们就引入了数据模型的概念。数据模型是成千上万数据的一种抽象,它不是描述某一个或某一种数据,而是描述所有数据的共性。一般来讲,数据模型包含两方面内容,它们是:

(1) 数据的静态特性,这主要包括数据的基本结构、数据间的联系和数据中的相互约束等特性。

(2) 数据的动态特性,这主要包括对数据操作的方法。

在实际的数据库设计中,E-R 模型(实体联系模型)是最为常用的一种。E-R 模型是于 1976 年提出的,它为数据库系统应用的设计人员提供了三个基本的模型描述成分,它们是实体、联系和属性。E-R 模型就是利用这三个成分来描述反映一个客观信息世界内容的数据模型。下面我们就来介绍一下 E-R 模型中的这三个基本成分:

1. 实体

我们知道,数据是用来描述现实世界中各种各样的事物对象的。这些事物对象的特征是可以相互有区别的,否则它们就会被认为是同一种对象。从实体联系模型的角度来看,凡是可以相互区别并可以被我们所识别的事物对象均可以被认为是实体。在实体中的各种具体实体,可以称为该实体的实体值。例如学生的集合是一个实体,一个名叫珊珊的女学生是学生实体中的具体实体。

2. 联系

任何的事物在现实世界中都不可能孤立地存在,与之相对应实体之间也存在着各种相互的联系。联系有不同类型的实体之间的联系和同一类型的实体联系,例如教师和学生之间的联系就是属于不同类型的实体之间的联系,而学生和学生之间的联系则属于同一类型的实体联系。在 E-R 模型中,通常将不同实体之间存在的联系归纳成三种不同类型,它们是:

(1) 一对一的联系

一对一联系的定义为:对于实体 A 中的某一个具体实体,在实体 B 中至多有一个具体的实体与之相对应,反过来对于实体 B 中的某一个具体实体,在实体 A 中至多有一个具体的实体与之对应。通常我们将这种联系记为 1:1 的联系。

(2) 一对多的联系

一对多联系的定义为:对于实体 A 中的某一个具体实体,在实体 B 中可以有若干个具体的实体与之相对应,反过来对于实体 B 中的某一个具体实体,在实体 A 中只能有一个具体的实体与之对应。通常我们将这种联系记为 1:N 的联系。

(3) 多对多的联系

多对多的联系定义为:对于实体 A 中的某一个具体实体,在实体 B 中可以有若干个具体的实体与之相对应,反过来对于实体 B 中的某一个具体实体,在实体 A 中也可以有多个具体的实体与之对应。通常我们将这种联系记为 N:N 的联系。

3. 属性

实体型是对实体集合中所有实体共同描述特征的集合,这些实体所共有的描述特征就称为实体的属性。在一个具体的实体型中,其中的每一个属性都有其取值范围,这一范围称为属

性的值域。实体中的某个属性的取值可以用来唯一区分实体型中的具体实体,如学生实体型中的学号属性的取值就可以用来区分每一个学生。所以,这种属性又可称为该实体型的标识码。

2.2 关系数据库

2.2.1 关系数据模型

关系模型是由 E. F. Codd 在 1970 年引入到数据库领域中来的。在这之后,数据库的专家们又定义了关系数据库的基本概念,引进了规范化的理论,提出了关系代数和关系演算等数据操纵语言,为数据库的发展奠定了坚实的基础。由于关系数据库是建立在较为完善的关系数学理论基础上的,它很好地解决了在数据库逻辑设计中存在的如何构造一个好的数据库逻辑结构问题,并且关系数据库中的规范化理论是数据库设计中的一种理论指南,所以到目前为止,关系数据库系统已成为数据库领域中最为活跃的一个分支。

关系数据模型是以集合论中的关系概念为基础而建立起来的数据模型,以下我们将介绍关系数据模型的基本知识。

1. 关系的数学定义

在这里我们应该先给出笛卡尔乘积的定义。给定一组集合 $D_1, D_2, \dots, D_n$, 则这组集合的笛卡尔乘积是集合: $D = D_1 \times D_2 \times \dots \times D_n \{ (d_1, d_2, \dots, d_n) \mid d_i \in D_i, i=1, 2, \dots, n \}$ 。这其中:笛卡尔乘积集合 D 中的每一个元素称为元组, n 表示参与笛卡尔乘积中的集合个数,又称为笛卡尔乘积的度。

关系数据库中关系的含义和我们平时所见到的关系的含义不尽相同。在数学上,关系的定义为:笛卡尔乘积 $D_1 \times D_2 \times \dots \times D_n$ 的一个子集 R , 被称为定义在集合 $D_1, D_2, \dots, D_n$ 之上的一个关系。集合 $D_1, D_2, \dots, D_n$ 称为关系 R 的域。例如,现有集合 $NUMBER = \{9511007, 9511033, 9511042\}$, 集合 $NAME = \{\text{周}, \text{刘}, \text{张}\}$, 集合 $AGE = \{21, 23\}$, 集合 $SEX = \{M, F\}$, 其笛卡尔乘积的一个子集如图 2.1 所示。

NUMBER	NAME	AGE	SEX
9511007	周	23	M
9511033	张	21	F
9511042	刘	23	M

图 2.1 笛卡尔乘积的一个子集

2. 关系数据模型

关系数据模型是定义在 E-R 模型和关系的数学定义基础上的一种数据模型,它将描述实体和联系的有关属性看成是集合,而将实体和联系认为是建立在这些集合之上的关系。一个关系实际上是由若干元组所构成的。这里的关系与数学上的关系在概念上有所不同,在数学上,关系中元组值是有序的,而在关系模型中,其元组值是无序的。为了形象地描述关系数据模型中各关系,通常是用一张简单的二维表格来描述一个关系的,该二维表分成两个不同部分,一个是表头部分,它描述关系的名称,又称为表名;另一个是表格内容,它描述关系中的具体元组

值。即表格中的每一行对应一个元组值,表格中的每一列对应一个属性。如图 2.2 所示。

学 号	姓 名	性 别	科 目	成 绩
98110001	蒋珊珊	女	数据库	96
98110120	王苗苗	女	程序设计	91
98110013	王建军	男	数据结构	67

图 2.2 学生成绩的二维表格

2.2.2 关系数据库系统

关系数据库系统的应用开始于 20 世纪 80 年代,到目前为止它已成为最为流行的数据库系统。在关系数据库系统中,其数据库中的数据是基于关系进行逻辑组织的。一个关系数据库是由若干个关系(即二维表格)所组成的。通常在关系数据库系统中,这些关系又被称为关系表。在关系数据库系统中,可以通过数据描述语言对其数据库中的各关系表进行描述。这种描述通常包括两个部分,即对关系表本身以及关系表中所包含的各属性集合进行描述。

为了对数据库中的各关系表进行存取,就必须提供基于关系数据模型的操作语言。从关系的角度出发,对关系的操作,理论上讲就是对集合的操作,而对关系进行各种不同操作及运算的总和就是关系代数,自 20 世纪 70 年代提出关系代数以来,它在关系数据操作语言的发展与研究中,一直起着较为重要的作用。这里,我们着重介绍与关系数据库数据操作密切相关的关系代数运算。它们是选择、投影和连接。

1. 选择运算(Selection)

选择运算是将关系作为元组的集合,从中选择出满足一定要求的元组,其运算结果是一个新的关系。简单地说,就是对关系表的行进行选择操作,从中选择出满足一定要求的行组成一个新的关系表。选择运算可以表示为:Select <关系表名> Where <选择条件>。

2. 投影运算(Projection)

投影运算是从构成一个关系的若干基本属性集合出发,从中选取某些基本属性集合,重新构造出另一关系,因此其运算结果也是产生一个新的关系。简单地说,就是对关系表的列进行选择操作,从中选择出若干列组成一个新关系表。投影运算可以表示为:Project<关系表名> On<属性名>。

3. 连接运算(Join)

连接运算是两个关系进行运算,从构成这两个关系的所有基本属性集合中,选择出满足一定要求的基本属性集合,重新构造一个新关系,因此其运算结果也是产生一个新的关系。简单地说,就是对两张关系表的行进行合并操作,从中选择出满足对列一定要求的行,组成一个新关系表。连接运算可以表示为:Join <关系表名 1> And <关系表名 2> Where <连接条件>。

选择运算、投影运算和连接运算是关系代数中最基本的数据操作运算,这些基本运算为建立关系数据库系统的操作语言奠定了基础。无论要检索满足何种条件的数据,其检索过程涉及到几个关系(表),运用这三种运算操作,都可以实现,从而保证了数据库的完备性。目前,实际上应用的关系数据库的数据操作语言都是基于这三种关系代数运算而构造的。

2.2.3 关系数据库理论

本小节主要介绍与建立关系数据库中的关系模式(即关系表结构)密切相关的关系数据库设计的理论基础,也就是关系范式的有关基本知识。范式理论为判断关系模式的好坏提供了理论标准,是数据库设计人员设计数据库的重要依据。

1. 数据库设计

数据库设计是应用系统开发过程中的重要一环,它是由数据库设计人员根据信息需要、处理需要、DBMS 说明书、操作系统及硬件环境说明进行设计。设计过程主要分为四个阶段:需求分析和数据分析阶段、概念设计阶段、逻辑设计阶段和物理设计阶段。

(1) 需求分析和数据分析

需求分析和数据分析主要是指对将要处理的对象进行调查,从调查中获得用户对数据库的信息要求、处理要求以及安全性和完整性要求。

(2) 概念设计

概念设计是指在需求分析和数据分析的基础上,将要处理对象的有关信息表示成用户容易理解的形式。

(3) 逻辑结构设计

逻辑结构设计是指将数据库的概念结构转换成所选用的 DBMS 所支持的数据库逻辑结构。数据库逻辑结构设计的结果并不是唯一的,要对数据库结构进行优化,可利用规范化理论进行数据库逻辑设计。

(4) 数据库物理设计

数据库的物理结构主要指数据库在物理设备上的存储结构和存取方法。对于给定的数据库逻辑结构选择一个最适合应用环境的物理结构的过程,称为数据库的物理设计。

2. 函数依赖

在实际的数据库设计当中,除了实体集之间存在着联系以外,在属性之间(即数据项之间)还存在着一定的依赖关系,由此就引入了属性间的函数依赖。函数依赖是数据库设计中的一个重要概念,下面我们就给出函数依赖的定义。

设 $R(U)$ 是属性集 U 上的一个关系模式, X, Y 是 U 的子集。若对于 $R(U)$ 中任意可能关系 r (即对于每一时刻的数据库中对应于关系模式 R 的内容), r 中不可能有两个元组在 X 的属性分量相等,而同时在 Y 的属性分量值却不等,则称 X 函数决定 Y , 或称 Y 函数依赖于 X 。记作 $X \rightarrow Y$ 。通俗地说,就是对于 r 中的属性或属性组 X 的每一个值, r 中 Y 只有一个值与之对应。

3. Armstrong 公理

在 1974 年, W. W. Armstrong 首先提出了函数依赖的公理系统,称为 Armstrong 公理。对于一组已知的函数依赖,利用该公理可以导出所蕴含的函数依赖。

Armstrong 公理:

设 U 为属性总体集, F 为 U 上的一组函数依赖,对于关系模式 $R(U, F)$ 有下列推理规则:

(1) 自反律

若 Y 包含于 X , X 包含于 U , 则 $X \rightarrow Y$ 为 F 所蕴含。

(2) 增广律

若 $X \rightarrow Y$ 为 F 所蕴含, 且 Z 包含于 U , 则 $X * Z \rightarrow Y * Z$ 为 F 所蕴含。

(3) 传递律

若 $X \rightarrow Y, Y \rightarrow Z$ 为 F 所蕴含, 则 $X \rightarrow Z$ 为 F 所蕴含。

4. 规范化设计方法

满足一定条件的关系模式, 称为范式。在 1971 年至 1972 年, E. F. Codd 系统地提出了第一范式(1NF)、第二范式(2NF)和第三范式(3NF)的概念, 1974 年 Codd 和 Boyce 共同提出了 BCNF, 1976 年 Fagin 提出了 4NF, 后来又有人提出了 5NF。一个低级范式的关系模式, 通过分解方法转换为若干个高一级范式的关系模式的集合, 称为规范化。

- 第一范式: 如果一个关系模式, 它的每一个分量是不可分的数据项, 则此关系模式为第一范式。
- 第二范式: 若关系模式 $R \in 1NF$, 且每个非主属性完全函数依赖于码, 则 $R \in 2NF$ 。
- 第三范式: 若关系模式 $R(U, F)$ 为第一范式, 不存在非主属性对码的传递依赖, 则称 $R(U, F)$ 为第三范式。
- BCNF 范式: BCNF 是修正的第三范式, 有时也称为第三范式。BCNF 消除了 3NF 中可能存在的主属性对码的部分函数依赖和传递函数依赖。

在范式中, 常用的是 3NF 和 BCNF。在设计数据库时要综合地考虑, 因为尽管分解关系模式可以减少数据冗余, 也可以克服数据操作容易出现异常等问题, 但随之而来的是数据库操作的复杂性被提高, 系统在进行多关系表操作时所付出的开销也更大。

2.3 SQL 语言

结构化查询语言(SQL)的极大普及是当今计算机工业中最引人瞩目的趋势之一, 目前 SQL 语言已经发展成为标准的关系数据库语言。在本节中将介绍关系数据库中的 SQL 语言的基本知识。

2.3.1 SQL 语言概述

SQL 语言是一种用于和关系数据库进行交互通信的计算机语言。当用户想检索数据库中的数据时, 就可以用 SQL 语言发出此请求, DBMS 对 SQL 请求进行处理, 检索到所要求的数据, 并将其返回给用户。这个向数据库请求并得到数据的过程称为数据库查询, 这就是 SQL 语言(Structure Query Language)这一名称的由来。如今, SQL 语言已不仅仅是一个查询工具, 它已成为可以对关系数据库中的数据进行组织、管理和检索的主要工具。

现在, SQL 用于控制 DBMS 提供给用户的所有功能, SQL 语言已成为用户与基于 SQL 的 DBMS 的接口。它包括以下几个主要的功能:

- 数据定义: SQL 语言可用于定义被存放数据的结构和组织, 以及各数据项间的相互关系。
- 数据检索: SQL 语言能使用户或应用程序从数据库中检索数据并使用这些数据。
- 数据操作: 用户或应用程序通过 SQL 语言可以更改数据库内容, 如增加新数据、删除旧数据或修改已存入的数据等。
- 存取控制: SQL 语言可用来限制用户检索, 增加和修改用户操作数据的权限, 以保护

所存储的数据不被非法存取。

- 数据共享：SQL 语言可用于数据操作控制，以便让并发用户共享同一数据，并保证用户之间不受影响。
- 数据完整性：使用 SQL 语言能对数据库内容的完整性条件作出规定，以使其不会因为修改混乱或系统出错而被破坏。

由此可见，SQL 是一种能控制数据库管理系统并能与之交互的综合性语言。但是，SQL 却不是一种像 Basic 和 C 那样的完整的程序设计语言。在 SQL 语言中，没有用于条件测试的 IF 语句，也没有用于循环的 For 或 While 语句。确切地讲，SQL 只是一种数据库子语言。但是，SQL 可以被嵌入到另一种语言中去（例如 Basic 语言），从而使其相应的程序设计语言具有数据库的存取功能。

SQL 是数据库管理系统的一个重要组成部分，是用户与数据库引擎进行通信的语言和工具。数据库引擎(DataBase Engine)是 DBMS 的核心，它负责数据的实际组织、存储和检索等工作，并可以接受从其它 DBMS 部分发送而来的 SQL 请求，还可以接受用户应用程序，甚至是其它计算机数据库系统所发送的 SQL 请求，并加以处理。以下是 SQL 语言的主要特点：

- SQL 是一种交互式查询语言。用户可以通过键入 SQL 命令来检索数据，并将其显示在屏幕上。这是一种简单易用的数据查询方法。
- SQL 是一种数据库编程语言。程序员可以将 SQL 命令嵌入到应用程序中去，以存取数据库中的数据。用户程序和数据库应用程序都采用这种方法进行数据操纵的。
- SQL 是一种数据库管理语言。数据库管理员可以利用 SQL 来定义数据库组织结构、控制数据存取等，从而实现对大型数据库系统的管理。
- SQL 是一种客户/服务器语言。个人计算机利用 SQL 与存放有共享数据的服务器通过网络进行交互通信。目前有许多应用都采用这种客户/服务器模式，以减轻网络的拥挤状况，使 PC 机和服务器各显其能。
- SQL 是一种分布式数据库语言。分布式 DBMS 利用 SQL 将数据分配给多台通过网络连在一起的所构成的分布式计算机系统。每台计算机上的 DBMS 都用 SQL 和其它计算机通信，发送数据存取请求。
- SQL 是一种数据库网关语言。在混用不同 DBMS 产品的网络中，SQL 通常被用来作网点，以使这些 DBMS 间能相互通信。

2.3.2 SQL 的元素

1. 模式对象

模式为数据逻辑结构或模式对象(schema object)的集合，它为一个数据库用户所占有，模式名与该用户名同名。模式对象是由 SQL 建立和操纵的，它包含下列类型。对这些元素的理解对有效地使用 SQL 语言很重要。

- 表(table)：表为数据库中数据存储的基本单位，其数据按行、列存储。每个表具有一表名和列的集合。每一列有一个列名、数据类型、宽度或精度、比例。一行是对应单个记录的列信息的集合。
- 视图(view)：一个视图是由一个或多个表(或其它视图)中的数据的一种定制的表达，是用一个查询定义，所以可认为是一个存储的查询或是一个虚表。在许多使用表的地方也同时使用视图，但视图与表不同，一个视图不分配任何存储空间，也就是说，视图

并不真正地包含数据。

- 聚集(cluster): 聚集是存储表数据的可选择的方法。一个聚集是一组表,它将具有同一公共列值的行存储在一起,并且它们经常一起使用。
- 数据库链(database link): 数据库链是一个命名的对象,说明从一数据库到另一数据库的一条路径。在分布式数据库中,对全局对象名引用时,数据库链被隐式地使用。
- 索引(index): 索引是与表和聚集相关的一种选择结构。索引是为提高数据检索的性能而建立的,利用它可快速地确定指定的信息。
- 序列生成器(sequence generator): 序列生成器产生序列号。在多用户环境下该序列生成器特别有用,可生成和返回序列号而不需要磁盘 I/O 或事务封锁。
- 同义词(synonym): 一个同义词为任何表、视图、序列、过程等的别名,其定义存储在数据字典中。
- 数据库触发器(database trigger): 数据库触发器类似于存储的过程,它包含 SQL 语言和其它的存储过程。

2. 非模式对象

在数据库中,还有一些其它类型的对象,可用 SQL 建立和操纵,但是它们并不包含在模式中。这些对象是:

- 用户环境文件(profile):
- 角色(role): 角色为相关特权的命名组,可授权给用户和角色。所谓特权是一种特殊类型的 SQL 语句或存取另一用户的对象的权利。
- 回滚段(rollback segment): 回滚段是数据库使用的一种对象,用于存储要撤消的事务所作的修改所需的数据。在一个表空间中可以有多个回滚段,一般多个回滚段可改进性能。
- 表空间(tablespace): 一个数据库划分为一个或多个逻辑单位,该逻辑单位称为表空间。

2.3.3 SQL 语言

在讲述 SQL 语言之前,我们先定义一个小型销售公司关系数据库的内容,除非有特殊的说明,否则本小节中的所有示例均是基于这个数据库的。该数据库所包含的关系表有:

- OFFICE 表。该表的列有: OFFICE, CITY, REGION, MGR, TARGET 和 SALES; 分别表示销售处编号,销售处所在的城市名,销售处所在的地区名,销售处经理编号,销售处的目标和销售处的销售额。
- SALESREPS 表。该表的列有: EMPL_NUM, NAME, AGE, REP_OFFICE, TITLE, HIRE_DATE, MANAGER, QUOTA 和 SALES; 分别表示销售人员编号,销售人员姓名,销售人员年龄,销售人员所在的销售处编号,销售处名称,雇佣日期,销售人员负责人编号,销售人员定额和销售人员的销售额。
- ORDERS 表。该表的列有: ORDER_NUM, ORDER_DATE, CUST, REP, MFR, PRODUCT, QTY 和 AMOUT; 分别表示订单编号,定单日期,顾客编号,销售处编号,制造商编号,产品型号,产品数量和订单金额。

1. 数据查询

查询是 SQL 语言的核心,SQL 语言只提供唯一一个用于数据库查询的语句,即 SELECT

语句。用于表达 SQL 查询的 SELECT 语句是功能最强也是最复杂的 SQL 语句,它提供了很多选项和使用方法。SELECT 语句的命令格式如下:

```
SELECT [ALL|DISTINCT] <关系表字段(表达式)列表| * >
FROM [关系表名(别名)列表]
[WHERE 查询条件]
[GROUP BY 分组要求]
[HAVING 分组搜索条件]
[ORDER BY 排序要求]
[INTO <新关系表名>]
```

SELECT 查询语句是由七个子句构成,其中 SELECT 和 FROM 子句是一个完整 SELECT 查询语句必须要有的,其它的子句可以根据具体需要任选。上述的每个子句功能说明如下:

(1) SELECT 子句

列出所有要求 SELECT 语句进行检索的数据项。这些项可能取自数据库中关系表的列,也可以是 SQL 在执行查询时需要计算的表达式。这里的 ALL 和 DISTINCT 选项,表示查询出的结果中是否容许有内容重复的行出现,缺省时是 ALL 选项,表示容许有内容重复的行出现。而 * 则表示查询出所指定关系表中所有的列。

(2) FROM 子句

FROM 子句列出包含所要查询的数据关系表。

(3) WHERE 子句

WHERE 子句告诉 SQL 只查询某些关系表中满足一定要求的行的数据,查询要求由 WHERE 子句中的查询条件确定。

(4) GROUP BY 子句

GROUP BY 子句指定当前查询是汇总查询,即不是对根据每行产生一个查询结果,而是对相似的行进行分组,然后再对每组产生一个汇总查询结果。

(5) HAVING 子句

HAVING 子句告诉 SQL 只对由 GROUP BY 所得到的某些行组的结果进行过滤,选择出满足分组条件的分组。

(6) ORDER BY 子句

ORDER BY 子句确定是否将查询出的结果按一列或多列中的数据进行排序。缺省时是不排序的。

(7) INTO 子句

INTO 子句确定是否将查询出的结果存入一张新的关系表中,缺省时只将查询出现的结果显示在屏幕上。这是非标准 SELECT 语句中的子句,但目前绝大多数实际应用的 SQL 数据库系统的 SQL 语言提供了这一选项。

下面是一些使用 SELECT 查询语句进行数据库查询的示例。

【例 1】 列出销售额超过 6000 元的销售人员的姓名、销售目标和超过销售目标的销售额。

```
SELECT NAME,QUOTA,(SALES-QUOTA)
```

```
FROM SALESREPS
WHERE SALES>6000
```

【例 2】 查出 1999 年最后一个季度的订单情况。

```
SELECT ORDER_NUM,ORDER_DATE,PRODUCT,AMOUNT
FROM ORDERS
WHERE ORDER_DATE BETWEEN "01-OCT-99"AND "31-DEC-99"
```

【例 3】 查出名称是以 ABC 开头的产品订单情况。

```
SELECT ORDER_NUM,ORDER_DATE,PRODUCT,AMOUNT
FROM ORDERS
WHERE PRODUCT LIKE "ABC%"
```

【例 4】 列出所有的销售处,按区域名字母顺序排列。

```
SELECT CITY,REGION
FROM OFFICE
ORDER BY REGION ASC
```

在这里 ASC 表示升序,如果想表示降序可用 DESC。

【例 5】 列出每个销售人员以及他们工作的城市和区域的情况。

```
SELECT NAME,CITY,REGION
FROM SALESREPS,OFFICE
WHERE REP_OFFICE = OFFICE
```

【例 6】 查出有多少销售员的销售额超过了其目标额,以及他们的销售额总和。

```
SELECT COUNT(NAME),SUM(SALES)
FROM SALESREPS
WHERE SALES>QUOTA
```

读者要注意的是在本例中使用了函数,与前面介绍的查询不同之处在于:它不仅仅是从数据库中查出某些数据,并且对查出的数据进行多种计算处理,并将处理的结果作为查询结果输出。SQL 语言提供了 6 种处理函数,帮助完成不同的处理工作。这 6 种处理函数是:

- SUM(),用于计算表中某一列的总和。
- AVG(),用于计算表中某一列的平均值。
- MIN(),用于选择表中某一列的最小值。
- MAX(),用于选择表中某一列的最大值。
- COUNT(),用于计算表中某一列中值的个数。
- COUNT(*),用于计算某张表的行数。

【例 7】 对于每两个或更多的销售员的销售点,计算其中所有销售员的总销售目标值和总销售员金额。

```
SELECT CITY,SUM(QUOTA),SUM(SALESREPS.SALES)
FROM OFFICE,SALESREPS
GROUP BY CITY
HAVING COUNT(*) >=2
```

【例 8】 列出哪些所有的销售员已销售了销售点指标 50%的销售点和指标。

```
SELECT CITY,TARGET
FROM OFFICE
WHERE (0.5 * TARGET) < ALL( SELECT SALES
                             FROM OFFICE
                             WHERE REP_OFFICE = OFFICE )
```

至此,我们介绍完了有关 SELECT 查询语句所有的基本使用方法,作为总结,下面我们给出 SELECT 查询处理的工作过程描述:

- 形成 FROM 子句所指定的目标表(表的乘积)。如果 FROM 子句仅指定一个表,则此表为目标表。
- 如果有 WHERE 子句,则将其搜索条件用于目标表中的每一行,保留使搜索条件为 TRUE 的行,剔除使搜索条件为 FALSE 或 NULL 的行。若 WHERE 子句中包含另一个 SELECT 查询语句,先去完成另一个 SELECT 查询语句操作,然后将其查询结果用到本次查询的搜索条件中,进行检测。
- 若有 GROUP BY 子句,则对目标表中所保留的行进行分组,以使得每组中的各行具有相同的分组列的列值。
- 若有 HAVING 子句,则将其搜索条件作用于每个行组,保留那些使搜索条件为 TRUE 的组,剔除使搜索条件为 FALSE 或 NULL 的行。若 HAVING 子句中包含另一个 SELECT 查询语句,先去完成另一个 SELECT 查询语句操作,然后将其查询结果用到本次查询的 HAVING 子句搜索条件中,进行检测。
- 对于所保留的每一行(或行组),计算由 SELECT 子句所指定的每项值,对于简单的列指定,取当前行(或行组)中该列的值;对于列函数,若有 GROUP BY 子句,则用当前行组作为其参数;否则,用所有行作为其参数;剔除 SELECT 子句中没有指定的各列。
- 若 SELECT 子句后选择了 DISTINCT,则将所生成结果中的所有内容重复的行剔除掉。
- 若有 ORDER BY 子句,则按相应的排序设置对结果进行排序。

2. 数据添加

SQL 提供了两种用于向数据库关系表中添加新行的方法,它们是单行 INSERT 语句和多行 INSERT 语句。下面我们简单地介绍这两种语句的功能和使用方法。

(1) 单行 INSERT 语句

单行 INSERT 语句用于向一关系表中添加一行新数据。其使用格式如下:

```
INSERT INTO <表名> (列名列表)
VALUES (列值列表)
```

其中,列值列表中的各数值顺序与列名列表中的各列名相互对应。

(2) 多行 INSERT 语句

多行 INSERT 语句用于向一关系表中添加若干行新数据。其使用格式如下:

```
INSERT INTO <表名> (列名列表)
(SELECT 语句)
```

【例9】 将订单表中1999年1月1日前的订单编号、日期和订购数量保存到另一关系表 OLDORDERS 中。

```
INSERT INTO OLDORDERS
(ORDER_NUM,ORDER_DATE,AMOUT)
SELECT ORDER_NUM,ORDER_DATE,AMOUNT
FROM ORDERS
WHERE ORDER_DATE < "01-JAN-99"
```

这里需要说明的是,在多行 INSERT 语句中,其数据来源是一个 SELECT 查询语句的结果。即多行 INSERT 语句与数据库内容的复制功能类似。其中 SELECT 查询语句的结果中列的顺序应与列名列表中各列名相互对应。

3. 数据删除

SQL 语言中用于删除数据库中某个关系表中的数据的语句是 DELETE。它可用于完成删除一关系表中一行或若干行,甚至全部行的操作。其使用格式如下:

```
DELETE FROM <表名>
[WHERE <搜索条件>]
```

这里 WHERE 子句中的搜索条件,同样容许含有另一个 SELECT 查询语句。

4. 数据修改

SQL 语言中用于修改数据库中某个关系表中数据的语句是 UPDATE。它可用于完成修改一关系表的选定行中的一列或若干列的值。其使用格式如下:

```
UPDATE <表名>
SET 列名 = 表达式,...,列名 = 表达式
[WHERE <搜索条件>]
```

这里 WHERE 子句中的搜索条件,同样容许含有另一个 SELECT 查询语句。

【例10】 给当前尚无定额的销售员分派6000元定额。

```
UPDATE SALESREPS
SET QUOTA = 6000
WHERE QUOTA IS NULL
```

【例11】 让接受1999年1月1日之后的订单总额超过9000元的销售员直接向 John(编号是104)汇报。

```
UPDATE SALESREPS
SET MANAGER=104
WHERE 9000 < ( SELECT SUM(AMOUT)
                FROM ORDERS
                WHERE ORDER_DATE > "01-01-99"
```

5. 数据库结构操作

SQL 语言提供了三个基本的用于数据库关系表定义的语句,它们是 CREATE, DROP 和 ALTER 语句。下面我们讲述这三种语句的功能和使用方法。

(1) CREATE 语句

SQL 语言中的 CREATE 语句可以用于定义和生成数据库中的关系表。其使用格式如下:

```
CREATE TABLE <表名>
( 列定义 | 主关键字定义 | 外部关键字定义 | 唯一性约束 | 校验约束 )
```

下面是一个 CREATE 语句使用的例子:

```
CREATE TABLE ORDERS
( ORDER_NUM INTEGER NOT NULL,
  ORDER_DATE DATE NOT NULL,
  REP INTEGER,
  MGR CHAR(3) NOT NULL,
  PRODUCT CHAR(5) NOT NULL,
  QTY INTEGER NOT NULL,
  AMOUNT MONEY NOT NULL,
  PRIMARY KEY (ORDER_NUM),
  FOREIGN KEY TAKENBY(REP)
    REFERENCES SALESREPS
    ON DELETE SET NULL)
```

(2) DROP 语句

SQL 语言中的 DROP 语句可以用来删除数据库中某张关系表。其使用格式如下:

```
DROP TABLE <表名>
[CASCADE|RESTRICT]
```

DROP 语句中的 CASCADE 或 RESTRICT 选项,用于说明删除父表时对其子表的处理方法。

(3) ALTER 语句

SQL 语言中的 ALTER 语句可以用于修改数据库中某张关系表的结构定义。其使用格式如下:

```
ALTER TABLE <表名>
( ADD 列定义 |
  ALTER 列名 SET DEFAULT/DROP DEFAULT |
  DROP 列名 CASCADE/RESTRICT |
  ADD 列名 主关键字定义/外部关键字定义/唯一性约束/校验约束 |
  DROP CONSTRAINT 约束名 CASCADE/RESTRICT)
```

6. 视图操作

数据库中关系表中定义了表的结构和编排方式。然而,SQL 语言又提供另外一种数据组织方法,通过它,用户可以按其它组织形式对原来表中的数据进行重新组织。这种方法就是视图。视图的结构和内容是通过 SQL 查询获得的,它有一个视图名,并可以永久地保存在数据库中。用户通过 SQL 查询语句,可以像其它普通关系表一样,对视图中的数据进行查询,仿佛它们就是数据库中的“真实表”。在下面,我们将对视图的创建和删除这两种操作的 SQL 语句的功能和使用方法做一介绍。

(1) 视图的创建

SQL 语言所提供的视图创建语句是 CREATE VIEW,其使用格式如下:

```
CREATE VIEW <视图名>
[列名列表] AS ( SELECT 查询语句)
```

从视图的创建语句内容,可以看出视图的内容是来自一个 SQL 查询的结果。列名列表选项如果存在,则视图中各列将由它来确定,否则由查询的结果所包括的列来确定。

(2) 视图的删除

SQL 语言所提供的视图删除语句是 DROP VIEW,其使用格式如下:

```
DROP VIEW <视图名>
[CASCADE|RESTRICT]
```

视图的删除与普通关系表的删除最大的不同点就是,删除视图仅仅是删除了视图的组织结构,用户以后就不能再用这个视图来进行操作,但组成视图内容的数据并没有删除,它们仍然保存在原来的关系表中。

7. 安全操作

下面,我们介绍 SQL 语言所提供的有关安全性操作的语句,主要是 GRANT 和 REVOKE 这两种。

(1) GRANT 语句

该语句被用来对特定的用户授予关于数据库对象安全性的权限。通常 GRANT 语句由表或视图的拥有者来使用,以授权其它用户对相应表或视图中的数据进行存取操作。其使用的格式如下:

```
GRANT [SELECT (列名序列)] / [,INSERT (列名序列)] /
[,DELETE (列名序列)] / [,UPDATE (列名序列)]
| ALL PRIVILEGES ON <表名> TO 用户标识符| PUBLIC
[WITH GRANT OPTION]
```

GRANT 语句包括一组被授予的权限、一个权限所作用的表、以及一个被授予权限的用户标识符。这里需要说明的是 SELECT(列名序列)、INSERT(列名序列)、DELETE(列名序列)和 UPDATE(列名序列)可以结合在一起进行授权。与它们相对应的是 ALL PRIVILEGES 选项,它是指前面所有的四种授权,且作用于表中所有列。PUBLIC 选项表示对所有的用户进行授权。而 WITH GRANT OPTION 选项,是说明其它被授权的用户是否可以将他们所获得的权限转授给其它别的用户。

(2) REVOKE 语句

通常在 SQL 数据库中,用 GRANT 语句已经授权的权限都可以用 REVOKE 语句来进行取消。REVOKE 语句的使用格式如下:

```
REVOKE [SELECT] / [,INSERT] / [,DELETE] /
[,UPDATE]| ALL PRIVILEGES
ON <表名> FROM 用户标识符序列| PUBLIC
```

REVOKE 语句与 GRANT 语句结构相似,它对一个特定的数据库目标,规定了一组特定的将被取消的权限,以及将被取消权限的若干用户标识符。REVOKE 语句可以帮助取消你以前授予每个用户标识符的所有权限或某些特权。

8. 事务处理

数据库的更新通常都是由客观世界的所发生的事件引起的。为保证数据库内容的一致,就要将数据库的一组操作作为一个整体来进行,要么全部成功完成,要么全部失败退出。如果由于故障或其它原因而使一组操作中有一些完成,有一些未完成,则必然会使得数据库中的数据

出现不一致,从而使得数据库的完整性受到破坏。因此,更新操作序列必须作为一个整体在 DBMS 执行时出现,即“要么全做,要么全不做”。SQL 提供了事务处理的机制,来帮助 DBMS 实现上述的功能。

事务处理(Transaction)是由一个或多个 SQL 语句序列结合在一起所形成的一个逻辑处理单元。事务处理中的每个语句都是完成整个任务的一部分工作,所有的语句组织在一起能够完成某一特定的任务。DBMS 在对事务处理中的语句进行处理时,是按照下面的约定来进行的,这就是“事务处理中的所有语句被作为一个原子工作单位,所有的语句即可成功地被执行,也可以没有任何一个语句被执行”。DBMS 负责完成这种约定,即使在事务处理中应用程序异常退出,或者是硬件出现故障等各种意外情况下,也是如此。在任何意外情况下,DBMS 都负责确保在系统恢复正常后,数据库内容决不会出现“部分事务处理中的语句被执行完”的情况。

SQL 语言为事务处理提供了两个重要的语句,它们是 COMMIT 和 ROLLBACK 语句。它们的使用格式是:

COMMIT WORK

ROLLBACK WORK

COMMIT 语句用于告诉 DBMS,事务处理中的语句被成功执行完成了。被成功执行完成后,数据库内容将是完整的。而 ROLLBACK 语句则是用于告诉 DBMS,事务处理中的语句不能被成功执行。这时候,DBMS 将恢复本次事务处理期间对数据库所进行的修改,使之恢复到本次事务处理之前的状态。

第三章 Visual Basic 数据库开发

相信用 Visual Basic 来开发近乎完美的数据库应用程序是广大 Visual Basic 使用者梦寐以求的事,这也正是我们讲述本书的目的。与其它专用的数据库开发工具类似,在用 Visual Basic 开发数据库应用程序中,会遇到许多关于 Visual Basic 数据库方面的专用概念。这些概念会贯穿于整个 Visual Basic 的开发过程。但是,这些专用概念对那些没有使用过 Visual Basic 开发数据库应用程序的人来说是相当陌生的。所以,在开始我们的实际数据库应用程序设计之前,有必要对这些概念进行说明。

在本章中我们主要讲述以下几个部分:

- 客户/服务器结构
- ActiveX 数据对象简介
- ADO 对象
- Visual Basic 对数据和数据源的访问

3.1 客户/服务器结构

3.1.1 客户/服务器的概念

Visual Basic 是一种用来开发客户机/服务器数据库应用程序的理想工具。使用客户机/服务器模式是当今数据库开发的主流。客户机/服务器模式是从模块化程序设计的基础上发展起来的,但是它把以往那种基于模块的设计思想更推进了一步;它允许模块可以不在同一存储空间中运行。在这种体系结构中,调用模块就成为客户(client)的一个请求,而被调用的模块就成为服务器(server)提供的服务。

客户机/服务器的逻辑扩展是让客户程序和服务器各自运行于相应的硬件和软件平台上,它们各负其责,相互协同地为同一个应用服务。在实际的应用设计中,应当把数据库的前端应用放到客户机上,而后端的数据库管理系统(DBMS)放在服务器上。处于前端的客户机通常用来管理整个系统的用户接口,检查用户输入数据的有效性并向后端的服务器发送请求,有时还执行一些逻辑上的运算。而位于后端的服务器则接受客户端的请求,执行数据库的查询和更新等操作,集中地管理数据,并对客户端的请求及时地作出响应。

客户机/服务器结构的基本特征如下:

- 它是由一个与用户交互的前台客户和一个与共享资源交互的后台服务器构成。客户进程提供了特定问题的解答逻辑,并且提供了用户与其他应用系统的接口。服务器进程作为软件引擎工作,它管理着共享资源。
- 前台任务和后台任务对计算资源的需求有本质上的区别。
- 这是一个典型的多机种、多供应商的环境。客户和服务器的硬件和操作系统平台通常都是不同的。客户和服务器进程通过一组严格定义的应用程序编程接口(API)来进行相互通信。

- 客户机/服务器系统的重要特征之一是可伸缩性。它既可以水平放缩也可以垂直地放缩。水平放缩是指可以添加或删除一些客户工作站而只对性能有轻微的影响。垂直放缩是指可以迁移到更大更快的服务器或多个服务器上。

3.1.2 客户/服务器体系结构

客户机/服务器结构有两种不同的类型——基于两层客户机/服务器结构和基于三层客户机/服务器结构。本小节对两种体系结构都将进行介绍。

1. 两层客户机/服务器体系结构

两层客户机/服务器体系结构是一个可以很好地满足应用程序处理需求的体系结构和技术。它用一个数据库服务器代替网络中的文件服务器,网络就可以响应用户在关系数据库(而不是文件系统)上的查询处理。这种方法的好处之一是可以显著地减少网络流量,而且在数据库服务器的管理下可以很容易地实现多用户更新。两层数据库结构如图 3.1 所示:

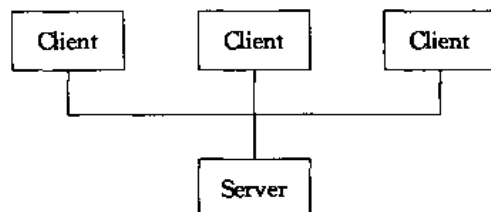


图 3.1 两层客户机/服务器体系结构

2. 三层客户机/服务器体系结构

一般地,在一个较大的客户机/服务器体系结构的企业环境中,两层体系结构的性能随着在线用户的上升而下降,这主要是因为数据库服务器的连接处理。数据库服务器需要为每一个连接到服务器的客户创建一个进程。两层方法的另一个问题是在当前实现中没有提供“既成事实分割”的余地,一旦一个应用程序开发成功,将它的某个功能模块从一个服务器移到另一个服务器是很不容易的,需要手工地重新生成过程代码。而在较新的三层方法中,提供了将应用程序代码模块拖放到不同计算机上的能力。

工业上已经对两层体系结构的限制作出了响应,在客户与服务器之间又添加了一个中间层。此中间层可以完成许多不同的功能——排队、程序执行和数据库分级等。在客户机/服务器技术中应用中间层已证明可以比两层的方法提供更好的性能和弹性。应用中间层的优点之一是,如果中间层提供了排队功能,那么两层方法中的同步处理就变成了异步处理。也就是说,客户可以将请求传输到中间层,然后就脱离,只要确定得到合适的响应就行了。另外,中间层还为要处理的工作安排日程和优先级。带有中间层的体系结构就称为三层或多层体系。三层数据库结构如图 3.2 所示:

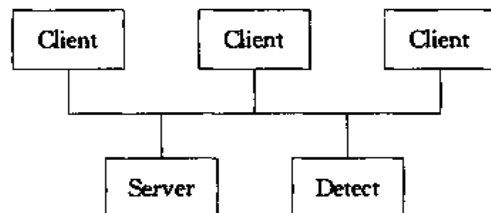


图 3.2 三层客户机/服务器体系结构

3.2 ActiveX 数据对象简介

3.2.1 COM 和 DCOM 概述

1. 组件对象模型

组件对象模型(Component Object Model) 是一种面向对象的结构,它使得对象可以从一个应用程序到另一个应用程序被重复地使用。这不仅使得程序员的工作量大大地减少,软件的可重复利用率明显地提高,也进一步使得用不同开发工具开发的应用程序之间的通信成为可能,例如你可以利用 COM 的规范编写一些对象,从而使这些对象在 Visual Basic 和 Visual C++ 之间进行通信。COM 之所以有这种能让广大程序员们兴奋不已的能力,是因为它为对象连接提供了一个标准的协议,当那些用不同的语言设计的对象进行连接时,对象之间通过一个标准的接口协议来互相通信。应该说,COM 是当今软件的发展方向之一。

长期以来,Microsoft 公司一直在致力于开发这种基于组件的应用程序的开发模型。在早期的 VB 版本中带有一种称为 VBX 的 VB 控件,它提供了可以嵌入到 VB 应用程序中去的定制的 Windows 控件。OLE 和 VBX 控件是 Microsoft 公司迈向面向对象的应用程序组件的第一步,后来发展到了 OLE 控件(OCX),OLE 控件是一种基于 OLE 的定制控件。在 OCX 之后,技术发展到了今天的 ActiveX,ActiveX 与 OCX 最主要的区别是它可以跨越各种平台,而不再是仅仅局限于 Windows。

OLE DB 是一种使用 OLE 组件对象模型(COM)来从各种不同的数据源中提取和操作数据的接口。OLE DB 接口对不同的数据源提供了统一访问数据的方式。OLE DB 是新的低层接口,它介绍了一种“通用的”数据访问范例。也就是说,OLE DB 并不局限于 ISAM、Jet 甚至关系数据源,但是它能够处理任何类型的数据,而不考虑它们的格式和存储方法。在实际应用中,这种多样性意味着可以访问驻留在 Excel 电子数据表、文本文件甚至邮件服务器,诸如 Microsoft Exchange 中的数据。

COM 使得程序开发人员可以创建一些应用程序组件,然后把它们组合成新的应用程序,这些组件可以用不同的语言独立开发,而且可以很容易地组合在一起构成新的应用程序或应用程序的一部分。应用程序的组件化有助于加速并简化应用程序的开发进程。也可以使用 COM 组件来实现应用程序之间的通信。与其它的模型不同的是 COM 体系的目的就是应用程序彼此通信开发的一种方法。在通信建立之后,应用程序就可以直接互相通信。

COM 模型的要点如下:

- 组件必须严格地遵循一套规则,这与以前提供的函数库和基于 Windows 的组件模型的通信方法(如 DLL)不同。
- 天生独立的组件。COM 组件可以用不同的程序设计语言开发而不需要任何特殊的接口来连接应用程序间的通信。这就使所有的 COM 组件有天生的兼容性。
- 真正的面向对象的组件。COM 组件定义了一套有固定接口集的封装对象,从而对组件的使用者隐藏了实现细节。
- 提供了封装对象生命周期的机制。这是通过参照计数来实现的。
- 全局唯一的标识码(GUID)。它用来唯一地标识一个 COM 对象,COM 使用 128 位的标识码,以确保每个对象标识的唯一性。

- 安全性继承。使用户享有基本的操作系统权限以确保他有权执行一组件。
- 支持分布式模型。这使得组件可以应用于网络环境,你可以访问或执行远程系统上的组件。

2. 分布式的组件对象模型

分布式的组件对象模型(COM)是使 ActiveX 组件能通过网络直接互相通信的对象协议。DCOM 与语言无关,因此任何可以生成 ActiveX 组件的语言都可以生成 DCOM 应用程序。读者如果理解了组件对象模型的概念,也就很容易理解分布式组件对象模型。

3.2.2 ActiveX

ActiveX 实际上是在 OLE 和 COM 基础上扩展起来的一套技术。COM 为 OLE 和 ActiveX 控件提供了底层的对象访问体系。ActiveX 与 OLE 组件非常类似,但是 ActiveX 组件比 OLE 组件更小而且更快,它十分适合于嵌入到网页中去,在这一点上,有过 Java 开发经验的读者应该深有体会。

当前,ActiveX 组件被所有的 Windows 平台以及 Macintosh 平台支持,而且最近又被移植到 UNIX 上。ActiveX 组件包括控件、代码部件和 ActiveX 文档。采用这些方案的某几种或者全部,能为业务需求创建出高性能的 Internet 解决方案。

1. ActiveX 控件

ActiveX 控件可以使 HTML 页面变得更加有生气,并具备更强的功能。除了创建自己的控件以外,还可以使用在 Visual Basic 专业版和企业版中提供的 Internet 控件来扩展含 Internet 技术的标准应用程序。举例来说,可以在 Web 页面上使用 ActiveX 控件向 Web 页面中增加自定义的菜单,增加滚动标题以显示广告或重要的公告信息,或者是用来增加动画效果。

2. ActiveX 代码部分

可以使用 ActiveX 代码部件(.dll 或 .exe 文件)向客户端或服务端的 HTML 页面中增加功能。客户端的代码组件可以提高速度,因为用户的命令不再需要通过寻径返回服务器了。服务器组件还具有能够显示用户界面元素的优点。

3. ActiveX 文档

ActiveX 文档是和 Internet 或 Intranet 站点中的其它元素紧密结合在一起的。在使用与 Visual Basic 的 Internet 应用程序无关的 DHTML 应用程序、IIS 应用程序或者 HTML 页面时,可以同时使用 ActiveX 文档。适当地安排这些组件,可以使用户能透明地在 ActiveX 文档、应用程序或 Web 站点的其它页面之间漫游。

3.3 ADO 对象

3.3.1 数据存取模式

Visual Basic 一共支持三种不同的面向对象的数据存取 API:数据存取对象(DAO)、远程数据对象(RDO)和 ActiveX 数据对象(ADO)。在本小节中我们来分别讨论这三种不同的数据存取模式。

1. DAO 简介

DAO 是 VB 最初、最古老的数据访问模式,它能用于两类不同的数据库环境,第一类是单

一索引序列数据库;第二类是客户机/服务器型的 ODBC 数据库。DAO 是 DLL 的集合,它是以 Microsoft 的 Jet 数据库引擎为核心而发展起来的一组动态链接库。它提供了两种类型来支持数据库:(1)通过 Microsoft 的 Jet 数据库引擎来操作本地的数据库,如 Foxbase, FoxPro 等。(2)通过使用 ODBC Direct,使得使用 DAO 来存取 ODBC 数据库成为可能。

2. RDO 简介

RDO 提供了一个抽象的层面,直接与 ODBC API 相连接,也就是 ODBC Direct 接口。RDO 使用 ODBC API 和数据服务驱动器创建数据库服务连接,创建查询和连接表来操纵结果数据组,并且执行依赖数据服务器的复杂程序。DAO 是为查询处理和连接管理而使用 Jet Database Engine,而 RDO 则是直接与数据库服务器连接。由于 RDO 直接与数据库服务器联系,所以它比较适合于客户机/服务器方式。

3. ADO 简介

ADO 是另一种用于开发访问 OLE DB 数据源应用程序的 API。有多种程序设计语言都支持 ADO,如 Visual Basic, Visual C++, Visual J++ 等。ADO 提供了较为高级并容易被理解的访问 OLE DB 数据源的机制,它结合了 DAO 和 RDO 的优点。并提供了 OLE DB 数据源的入口。OLE DB 是一种数据库体系结构,它为企业网络提供了普遍的数据综合能力(从大型机到桌面上的数据,而不管数据的具体类型是什么),与 ODBC 相比,OLE DB 在数据通道方面更具有普遍性和有效性,这是因为它允许同建立在 COM 基础上的更多的数据类型联系和由于 ADO 是为开发客户机/服务器应用程序而专门设计的,所以在本书中我们都将采用 ADO 数据存取模式。图 3.3 是典型 ADO 应用程序的结构。

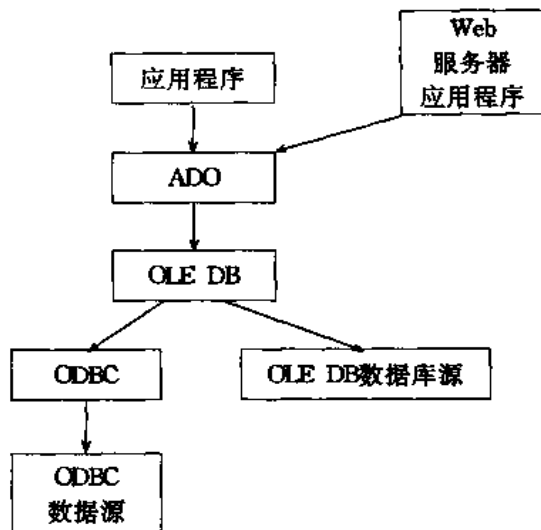


图 3.3 典型 ADO 应用程序的结构

ADO 的主要特征如下:

- 支持成批更新。即对更新的多个记录进行缓冲,并同步传输到服务器。
- 支持所有类型的光标。包括只能前移型,关键值集型以及动态和静态型。
- 支持服务器端的存储过程。这些过程可以大大提高应用程序的通用性。
- 支持返回多个记录集的查询。
- 支持查询目的。包括限制返回的记录个数,激活返回记录的过滤和预先语句。

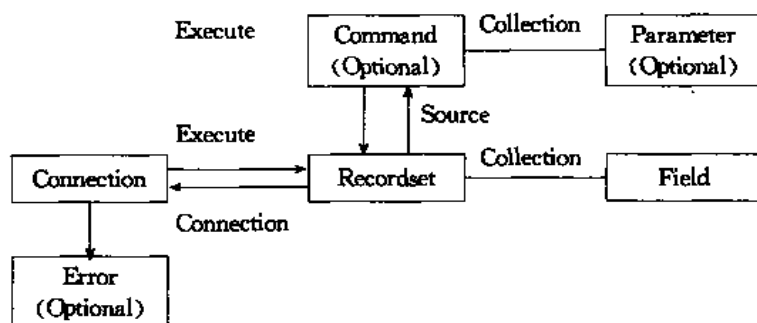


图 3.4 ADO 对象层次

ADO 是由以下对象组成的: Command, Connect, Error, Field, Parameter 和 RecordSet。图 3.4 给出了 ADO 对象的层次结构。除了 Error 和 Field 之外的所有 ADO 对象都是可以单独创建的,而 Command, Error 和 Parameter 对象是可选的。

3.3.2 ADO 对象介绍

1. Connection 对象

Connection 对象封装了到数据源的连接而且允许使用 Execute() 方法来执行 SQL 命令, Execute() 命令返回一个 RecordSet 对象。可使用 Connection 对象配置一个数据库的链接、定义脱机等级、执行 SQL 命令并对 SQL 语句的执行进行控制和管理。图 3.5 很好地显示了 Connection 对象同其它对象的关系。

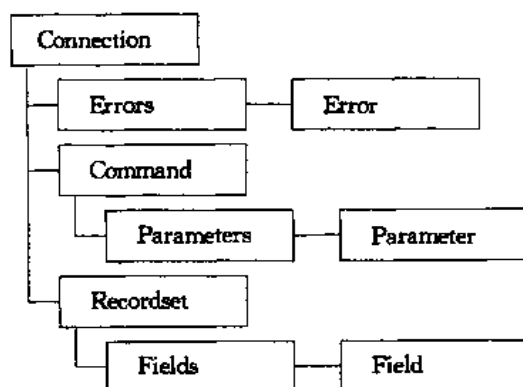


图 3.5 Connection 对象与其它对象的关系图

2. Command 对象

Command 对象定义了对数据源执行的指定命令。使用 Command 对象查询数据库并将查询所得到的记录集返回给 RecordSet 对象,以便执行大量操作或处理数据库结构。Command 对象的主要功能有:

- 使用数据管理命令创建 RecordSet 对象,完成成批更新和使用数据定义命令修改数据源结构。
- 打开到数据源的连接,指明一个命令,执行存储过程和创建语句。
- 与 Parameter 对象联合创建参数化的命令。应用程序可以往各参数集中添加参数,而

不需要数据提供者预先填充参数集。

3. Error 对象

Error 对象封装了从数据源返回的错误。如果 Error 对象可用,可使用它来获得错误描述、错误代码、创建错误的对象、参考帮助文件和当前的 SQL 状态。Errors 集合中包含了为响应涉及提供者的单个错误而创建的所有 Error 对象。任何涉及 ADO 对象的操作都可能产生一个或多个提供者错误。产生错误时,可以将一个或多个 Error 对象置于 Connection 对象的 Errors 集合中。其他 ADO 操作产生错误时,将会自动清空 Errors 集合,并且将新的 Error 对象置于 Errors 集合中。

4. Field 对象

Field 对象封装了 RecordSet 对象中的一列。可以这样说,一个 RecordSet 对象是由一组 Field 对象组成的。可以使用 Field 对象来访问列的名称、列值、类型、精度、范围和列的大小。Field 对象只能从 RecordSet 对象中来访问。

5. Paramater 对象

Paramater 对象封装了一个命令参数,此参数可以是输入型、输出型或者输入/输出型。许多数据源提供者都支持参数化的命令。需要进行的操作在这些命令中只定义一次,但可以使用变量(或参数)来改变命令的某些细节。例如,SQL SELECT 语句可使用某个参数定义 WHERE 子句的匹配条件,而使用另一个参数来定义 SORT BY 子句的列的名称。

6. RecordSet 对象

RecordSet 对象是 ADO 的核心,可以单独地创建。RecordSet 对象表示的是来自基本表或命令执行结果的记录全集。任何时候,RecordSet 对象所指的当前记录均为集合内的单个记录。在 RecordSet 对象中有一个默认的指针,用户程序可以通过这个默认指针的移动来访问记录集中的单个记录。RecordSet 对象封装了一组可以访问列级数据的 Field 对象。

3.4 Visual Basic 对数据和数据源的访问

3.4.1 Visual Basic 对数据的访问

在 Visual Basic 数据库编程中,不可避免地会有对数据的访问。如果在一个数据库应用程序中没有牵涉到对具体数据的访问,那是一件多么让人匪夷所思的事。接下来,我们将简略地介绍一下在 Visual Basic 中对数据的访问。也许一开始读者会对这些感到陌生,那没有关系,这里我们只是想让读者对此有一个大概的了解,具体的内容会放到后面的实例章节中讲述。

1. ActiveX 数据对象

在 Visual Basic 的三个版本中都提供了 ActiveX 数据对象(ADO),这项新的数据访问技术的特性包括:更简单的对象模型;与其它 Microsoft 和非 Microsoft 的技术更好的集成;为本地和远程数据提供的通用接口;可远程访问的和断开的记录集;用户可访问的数据绑定接口;以及层次结构的记录集。

2. 数据环境

数据环境设计器提供了一个创建 ADO 对象的交互式的设计环境,它包括在专业版和企业版中。这些可作为数据源供窗体或报表上的数据识别对象使用,或者在程序中作为显露在 Data Environment 对象之外的方法和属性进行访问。数据环境设计器不仅支持 Visual Basic

的 UserConnection 设计器的所有功能,还支持一些附加的功能,包括拖放、层次结构、分组以及合计等。数据环境设计器如图 3.6 所示。

要添加一个数据环境设计器对象到一个新的 Visual Basic 工程,请按照以下步骤执行:

- 在“工程”菜单中,单击“引用...”菜单项。
- 从“引用...”对话框中,选择“Data Environment 1.0”,然后单击“确定”。
- 从“新建工程”菜单的“新建”选项卡中,选择“标准 EXE”工程,然后单击“打开”。
- 从“工程”菜单中,选择“添加 Data Environment”菜单项。

一旦数据环境设计器被添加到你的 Visual Basic 工程中,数据环境设计器窗口出现,并且会自动地将一个 Connection 对象添加到你的数据环境。一旦有了 Connection 对象,你就可以对它添加 Command 对象了。

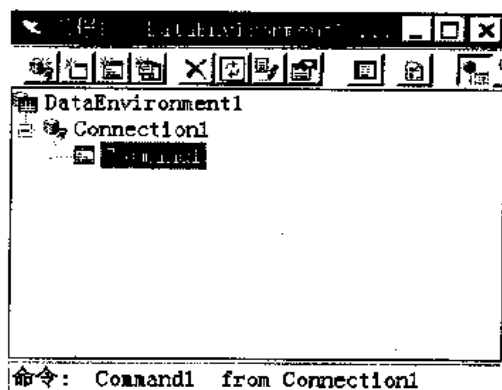


图 3.6 数据环境设计器

3. ADO Data 控件

在 Visual Basic 的三个版本中都提供了 ADO Data 控件,这是一种新的、OLE DB 识别的数据源控件,其功能与 Visual Basic 内部的数据控件和 Remote Data 控件十分相似,通过这种方式允许你使用最少的代码来创建你的数据库应用程序。

4. 增强的数据绑定

包括在专业版和企业版中。在 Visual Basic 以前的版本中,只能在窗体上绑定控件。在 Visual Basic 6.0 中,已经可以将任何 ADO/OLE DB 数据源绑定到任何 ADO/OLE DB 数据使用者上。在运行时,你可以设置控件的 DataSource 属性来将控件动态地绑定到数据源。你可以创建用作数据源和数据使用者的类,并且通过新的 BindingsCollection 对象将这些类绑定在一起。你可以创建与 ADO 数据控件相似的、用作数据源的用户控件。你还可以创建与 DataGrid 控件相似的、复杂绑定的用户控件。

5. OLE DB 支持

在 Visual Basic 的三个版本中都提供了这种支持。OLE DB 是一组 COM 接口,它为应用程序访问保存在不同信息源中的数据提供了统一的访问方法,这些信息源可以是关系型的,也可以是非关系型的。这些接口支持大量可应用于数据源的 DBMS 功能,使数据源能够共享其数据。ADO 是编程人员访问 OLE DB 的方法。所有新的数据绑定控件、数据环境和数据报表设计器都是 OLE DB 识别的。

6. 可视化数据库工具集成

包括在企业版中,利用集成的可视化数据库工具能够可视地创建和修改数据库结构和查询;例如创建 SQL Server 和 Oracle 数据库表等,利用拖放来创建视图、以及自动更改列的数据类型。

(1) 数据视图

“数据视图”窗口在一个或者多个数据库连接上显示一个视图,提供一个特定连接的数据库的整体结构的访问。“数据视图”窗口提供使用 Microsoft 可视化数据库工具(查询设计器和数据库设计器)的方法,来可视地操作数据库的结构。“数据视图”提供了一种图形化方法来组织数据库对象,这些数据库对象包括数据库图表、表、视图、触发器和存储过程。通过双击、拖放以及使用单击右键等特性,可以打开、创建和编辑数据库对象。要打开数据视图,如图 3.7 所示。

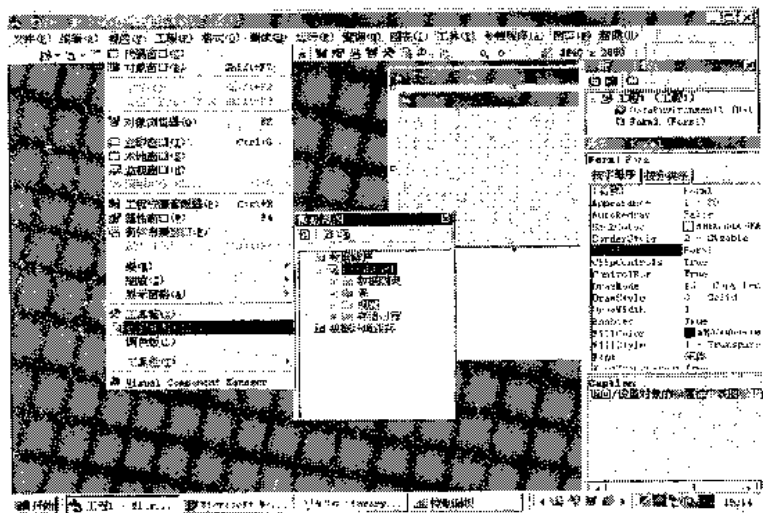


图 3.7 打开数据视图

(2) 数据库设计器

数据库设计器是一种可以在图型环境下进行数据库设计的可视化工具。图 3.8 显示了数据库设计器。如果读者所使用的不是 Visual Basic 6.0 企业版,则无法使用数据库设计器。使用此设计器可以:

- 创建和修改 Microsoft SQL Server 和 Oracle 数据库的结构。
- 创建、修改或删除例如表、关系、索引和约束等数据库对象。
- 在不影响数据库的情况下试验数据库设计,直到选择保存新的设计。
- 使用简单的界面执行复杂的 DDL 操作。例如,通过从下拉列表中选择不同的数据类型可以更改一列的数据类型。

可以利用数据库图表与服务器数据库进行交互。数据库图表以图形方式显示数据库中的表。这些表显示了它们包括的列、表之间的关系以及附加在表上的索引和约束。当要建立一个数据库的连接时,你可以利用数据库图表:查看数据库中的表和它们之间的关系;执行复杂操作以改变数据库的物理结构。你可以随意地更改数据库图表而不影响基层数据库。当通过

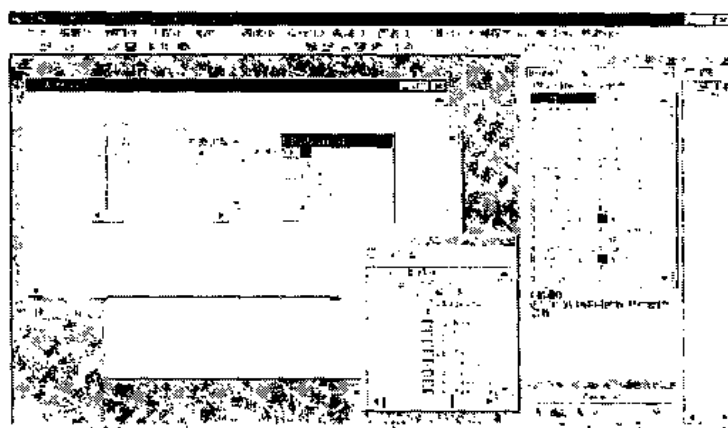


图 3.8 数据库设计器

数据库图表修改数据库对象时,所做的修改只有在保存表或数据库图表时才会被保存。这样,你就可以在数据库设计中利用“如果—就会”应用方案来做试验,而不永久性地影响现有的设计或数据。

(3) 查询设计器

查询设计器为你提供了四种不同的方法构造并执行针对任何遵从 ODBC 的数据库的查询。你可以:使用可视的查询图表将表拖动到查询中,并使用图形控件操作查询定义。使用准则网格指定搜索准则、排序次序、输出列等等。使用 SQL 窗格输入 ANSI-SQL 语句,或者让查询设计器为选择、插入、更新和删除查询生成 SQL 语句。在数据库表中浏览并编辑数据的当前视图。查询设计器如图 3.9 所示。考虑到大多数的用户不可能使用企业版,所以对查询设计

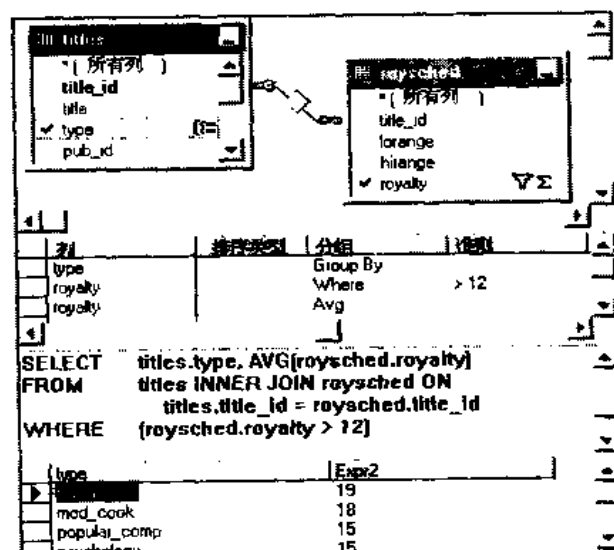


图 3.9 查询设计器

器的具体操作我们就不进行展开说明了,如果读者需要使用到关于这方面的具体操作,请自行参考 Visual Basic 的联机帮助文档。

7. 数据报表

数据报表设计器允许您利用拖放技术来快速地从任何记录集(包括层次结构的记录集)来创建报表,它包括在 Visual Basic 的专业版和企业版中,如图 3.10 所示。数据报表设计器由如下对象组成:

(1) DataReport 对象

DataReport 对象与 Visual Basic 窗体类似。DataReport 对象同时具有一个可视的设计器和一个代码模块。可以使用设计器创建报表的布局。也可以向设计器的代码模块添加代码,以采用编程方式调整设计器中包含的控件或部分的格式。

(2) Section 对象

数据报表设计器的每一个部分由 Sections 集合中的一个 Section 对象表示。设计时,每一部分由一个可以单击以选择页的标头和可以放置和定位控件的窗格表示。可以使用对象及其属性在报表生成之前对其进行动态重新配置。

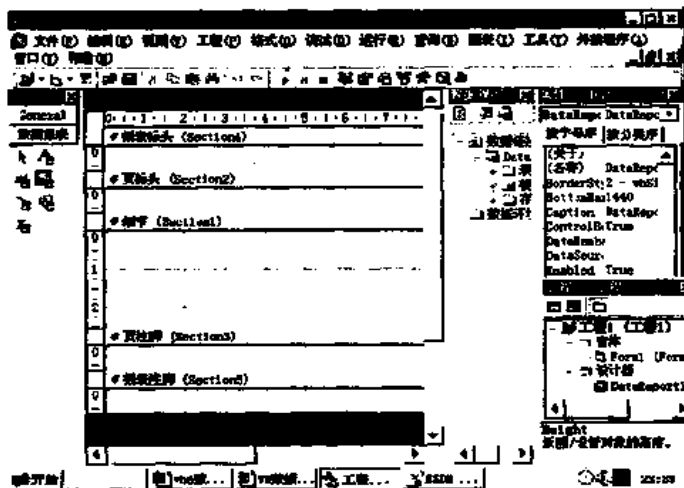


图 3.10 数据报表设计器

(3) DataReport 控件集

DataReport 控件集仅包含可在数据报表设计器上工作的特殊控件。(注意:不能在数据报表设计器上使用 Visual Basic 的内部控件或任何 ActiveX 控件)。这些控件可以在 Visual Basic Toolbox 中找到,但是它们被放置在一个独立的名为“数据报表”的选项卡上。

下面我们来介绍一下数据报表设计器中的 DataReport 控件集。当一个新的数据报表设计器被添加到一个工程时,下列控件将自动地被放置在名为 DataReport 的“工具箱”选项卡中:

- Label 控件(RptLabel)

Label 控件允许在报表上放置标签、标识字段或其它部分。

- TextBox 控件(RptTextBox)

TextBox 控件允许规定文本格式,或指定一个 DataFormat。

- Image 控件(RptImage)

Image 控件使用户能在报表上放置图形。注意,该控件不能被绑定到数据字段。

- Line 控件(RptLine)

Line 控件使用户能在报表上绘制标尺,以进一步区分每个部分。

- Shape 控件(RptShape)

Shape 控件使用户能在报表上放置矩形、三角形或圆形(椭圆型)。

- Function 控件(RptFunction)

Function 控件是一个特殊的文本框,用于报表生成时计算数值。

8. DataRepeater 控件

DataRepeater 控件包括在 Visual Basic 的专业版和企业版之中,它与 Access 的窗体类似,用户只要将 UserControl 插入到 DataRepeater 中就可以创建数据库的一个自定义视图。UserControl 可以包含 TextBox、CheckBox、DataGrid 或其他绑定到数据字段的控件。

9. Format 对象

在 Visual Basic 的所有版本中都包括。Format 对象在数据库和绑定控件之间提供双向的数据转换:当从数据库中读取一个值时,Format 对象将添加适合于绑定控件的格式。

10. DataGrid 控件

DataGrid 控件包括在 Visual Basic 的所有版本中,该控件可以让用户快速生成一个应用程序来查看和编辑记录集。它也支持新的 ADO Data 控件。

3.4.2 对数据源的访问

本小节要开始讨论使用 Visual Basic 开发数据库应用程序时几种经常访问的数据库。与其它数据库开发工具相同(例如 PowerSoft 公司的 PowerBuilder),使用 Visual Basic 开发数据库应用程序所完成的只是数据库的前端以及同数据库后端的接口的连接。而后端相应的数据源必须使用专门的数据库来进行管理。这也正好与客户机/服务器的数据库开发模式相对应。

1. Access 数据库

Access 是 Microsoft Office 这套功能强大的办公室软件包中所带的数据库组件,它以 Jet 数据库引擎来作为数据访问引擎。以下我们将对 Access 做较为详细的介绍。图 3.11 显示了 Access 2000 的界面。

(1) Access 概述

Microsoft Office 的 Access 数据库具有一个典型的关系数据库管理系统(DBMS)所具有的一切特征。Access 本身具有一种内置的语言,它是 Visual Basic 的一个子集(VBA),通过它,Access 可以创建应用程序。Access 包含一组丰富的数据库向导,通过使用这组向导,用户可以完成创建表和查询,定义表单和报表等多种数据库功能。让人感到鼓舞的是,由于 Microsoft 一开始所作出的关于兼容性方面的考虑,使得用 Access 可以访问其它格式的数据库文件,包括 dBase、Paradox、FoxPro 以及其它的 ODBC 数据源。

(2) Jet 数据库引擎

Jet 数据库引擎提供了对 Access 的支持。DAO 访问方法为访问 Jet 数据库做了优化。DAO 是组成 Jet 数据库引擎核心的一组动态链接库,它支持两种不同类型的数据库环境:

- 使用 Microsoft Jet 数据库引擎时,DAO 为访问本地的 ISAM(Index Sequential Access

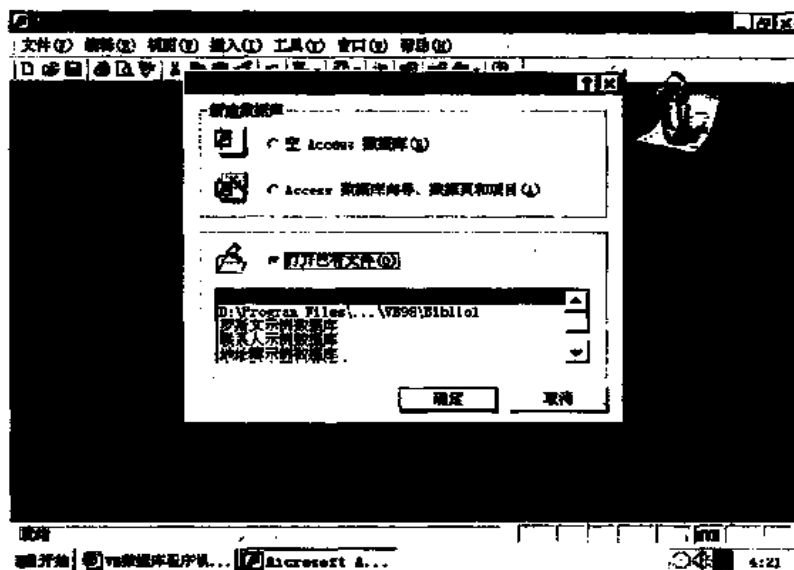


图 3.11 使用 Access 2000

Method,索引顺序存取法)数据库——像 dBase,Paradox,FoxPro 等做了优化。

- 使用 ODBC Direct 时,DAO 为访问 ODBC 数据库做了优化。

DAO Jet 数据库引擎不是一个真正的客户机/服务器模式的数据库引擎。用 Jet 数据库引擎开发的每个程序,都需要有 DAO DLL 文件的一个本地拷贝才能访问数据库,即使程序员正在运行多个客户机。然而,通过连接至 ODBC 数据库源,就能用 DAO 创建客户机/服务器程序。如果是在自然模式下使用 Jet 数据库引擎,DAO 创建的数据库可被 Microsoft Access 直接访问。Jet 数据库源有的是用 DAO API 创建的,有的是用 Microsoft Access 或 Microsoft Excel 创建的。

(3) Access 的使用

Access 2000 的对象中包括以下几个主要部分:

- 表:表是任何一个关系型数据库系统的关键部分,Access 表的字段支持文本、数字、日期、货币等多种数据类型。
- 查询:以 Access 表单的形式显示给用户,用来查看、修改和分析多种格式的数据,还可以用作表单和报表的数据源。
- 窗体:用户可以通过向导来创建窗体,窗体事实上是一种显示方式,它把数据库中的数据根据用户的需要显示出来。
- 报表:用于以打印的方式显示数据。Access 报表使你能够控制报表中的任何东西的任何特性,包括大小和外观。报表功能的使用必须在系统安装了打印机之后才能实现,如果用户还未安装打印机而又想使用报表功能,Access 会提示你先要安装打印机。
- 页:Web 上的功能项,对于页功能的使用首先必须安装 IE 5.0 或更高的版本。
- 宏和模块:宏是执行特定任务的一系列操作的集合——类似于 Visual Basic 中的子程序。这里的模块类似于 Visual Basic 中的模块,它包含存储在一起的声明和程序的集合。

Access 数据库比较适合用于小型数据库事务的开发,目前市场上大多数的第三方工具都

能够协同 Access 来共同管理你的小型事务。当你将要开发一个小型的数据库系统时,你可以使用 Access。Access 数据库及 Jet 数据库引擎并不是为人数过多的开发组设计的。如果你不能确定你的开发产品中的数据量,那么就请你最好使用 Microsoft 的 SQL Server 或者 Oracle 数据库。

2. SQL Server

当用户要使用 Visual Basic 来开发一个大型的数据库系统而又发现 Access 并不能完全胜任时,你应该考虑使用 Microsoft 的另一产品——SQL Server。与 Access 数据库不同,SQL Server 并不提供单独的、完全自给自足的应用程序开发环境。也就是说,用户能够使用 Access 自身来开发那些纯数据访问的应用程序,而 SQL Server 则是完全地作为后端来管理和运行数据库。到目前为止,SQL Server 已经发展到了 7.0 版本。

(1) SQL Server 的主要特点

SQL Server 7 是一个灵活性高、完全关系的、高性能的多用户数据库服务器,它主要是应用于大中型企业,为客户机/服务器应用程序管理大量的数据。SQL Server 7 的主要特点如下:

- 支持多用户:此能力可以为大中型企业提供对同时发生的数据库使用的支持。
- 高度灵活性:此能力得益于内建的 Windows NT 灵活性——包括支持多处理器和以元字节大小方式支持数据库。
- SQL-92 标准:Transact SQL 与 ANSI SQL-92 标准是完全兼容的。
- 完整的管理和开发工具集:SQL Server 7 提供各种工具来管理单独的和多 SQL 用户,包括为集成管理服务的 Microsoft 管理平台(MMC)。
- 并行数据库备份和恢复:能够备份并恢复数据库和事务日志,并能自动地从介质、用户和服务器死机等错误中恢复。
- 数据复制:从一个数据库到另一个数据库的表及事务的复制。
- 数据仓库:能够管理并存储大量的数据。SQL Server 7 可以打破元字节界限而达到数据库大小。
- 分布式查询:能够自动地在多 SQL Server 上运行查询,从而提高性能。
- 分布式事务:能够创建、管理并协同使用两个或多个的 SQL Server 的事务。
- 支持 Internet:与 Internet 信息服务器(IIS)和 Visual InterDev 的完全集成。
- 集成 Windows NT 安全性:能够使用 Windows NT 的域安全性——包括用户和组。
- 集成邮件:与 Microsoft Exchange 服务器的完全集成,为的是自动发布警告和错误信息。

(2) SQL Server 数据库引擎

SQL Server 数据库引擎能非常有效地使用多线程和多处理器,SQL Server 可以配置为适应某个分配给指定处理器的线程,或禁止分配给该处理器。从操作系统的角度来看,每个线程可以得到由单一进程所共享的内存。由于 SQL Server 使用本地 Windows NT 的多线程,它将错误更多地分离开,并使任务之间的切换更加平滑和有效。

为了有效地使用多处理器,SQL Server 引进了对称服务器结构和本地线程级多处理方式。如下所述:

- 对称服务器结构:SQL Server 运行为单一的多线程处理,这就帮助减少了内存使用和系统负载。

- 本地线程级多处理方式: SQL Server 通过使用多线程来使用多处理器,这也是 Windows NT 中的一个关键特点,是因为它包含了对多处理器最有效的使用。多线程操作系统最有效地利用了多处理器,这主要是因为处理器不用花很长时间来等待下一个任务。

(3) 数据存取模型的选择

所有的数据存取模型我们在前面章节中都已经做了介绍,包括 DAO、RDO 和 ADO,这些数据存取模型可以用来存取 SQL Server 数据库源。因为 Microsoft SQL Server 是一个服务器端数据库源,所以 RDO 和 ADO 更适合用来开发数据存取应用程序。另外,Microsoft SQL Server 7 是第一个主要数据供应商,它提供本地 OLE DB 数据存取,这就使得 ADO 成为越来越重要的存取 SQL Server 7 数据库的方法。当然,使用 DAO ODBC Direct 模型也可以优化来存取 SQL Server 7 数据库。如果用户需要最佳性能的话,可以直接存取 ODBC 接口。

3. Oracle 数据库

Oracle 系统是由以 RDBMS 为核心的一批软件产品构成,在 Oracle RDBMS 中带有过程数据库选项、分布式数据库选项和并行服务器选项。Oracle 的协同开发环境(简称 CDE)提供了新一代集成的软件生命周期开发环境,可用以实现高生产率、大型事务处理及客户/服务器结构的应用系统。

Oracle 公司是第一个推出基于 SQL 标准的关系数据库产品的公司。它推出的 Oracle 数据库系统支持多种硬件平台及操作系统。用户的 Oracle 应用可以很方便地从一种计算机配置移至到另一种计算机配置上。Oracle 的分布式结构可将数据和应用驻留在多台计算机上,而相互间的通信是透明的。Oracle 数据库系统具有以下的特点:

- 支持大数据库、多用户的高性能的事务处理。Oracle 支持最大数据库,其大小可达到几百千兆,可以充分地利用硬件设备。支持大量用户同时在同一数据上执行各种数据应用,并使数据争用最小、保证数据一致性。系统维护具有高的性能,Oracle 每天可连续 24 小时工作,正常的系统操作不会中断数据库的使用。可控制数据库数据的可用性,可在数据库级或在子数据库级上控制。
- Oracle 遵守数据存取语言、操作系统、用户接口和网络通信协议的工业标准。所以它是一个开放系统,保护了用户的投资。
- 实施安全性控制和完整性控制。Oracle 为限制和监控数据存取提供系统可靠的安全性。Oracle 实施数据完整性,为可接受的数据指定标准。
- 支持分布式数据库和分布处理。Oracle 为了充分利用计算机系统和网络,允许将处理分为数据库服务器和客户应用程序,所有共享的数据管理由数据库管理系统的计算机处理,而运行数据库应用的工作站集中于解释和显示数据。通过网络连接的计算环境,Oracle 将存放在多台计算机上的数据组合成一个逻辑数据库,可被全部网络用户存取。分布式系统像集中式数据库一样具有用户透明性和数据一致性。
- 具有可移植性、可兼容性和可连接性。由于 Oracle 可在许多不同的操作系统上运行,以致在 Oracle 上所开发的应用可以移植到任何操作系统,只需很少修改或不需要修改。Oracle 软件同工业标准相兼容,包括许多工业标准的操作系统,所开发应用系统可在任何操作系统上运行。可连接性是指 Oracle 允许不同类型的计算机和操作系统通过网络可共享信息。

第四章 配置数据源实例

数据源是数据的提供者。存在数据库中的数据要提供给用户的应用程序,必需要通过一个合理化的接口。在用户数据和放在数据库服务器上的数据库的通信过程中,数据源就起到了这个作用。当然,我们不能把数据源作为一个标准接口的规范来看待,因为事实上,数据源同数据库一样,是一个比较模糊的概念。我们很难用一个确切的定义来描述它,实际上也没有这种确切的定义存在。在这里我们可以从两个层次上来认识数据源:从用户的角度来看,数据源是数据的直接提供者,用户的应用程序是通过数据源来存取数据库中的数据的,所以应用程序同数据库的连接实际上是同数据源的连接;从数据库的角度来看,数据源为数据库中的数据提供了一种和用户的应用程序交互的方法。

在本章中我们主要讲述以下几个部分:

- ODBC 数据源介绍
- Access 数据源的配置
- SQL Server 数据源的配置

4.1 ODBC 数据源介绍

ODBC(开放式数据库的连接)是一种应用程序的接口(API)。这种接口提供了独立于任何的数据库管理系统编写应用程序的能力。ODBC 通过 ODBC 驱动程序提供了对不同的数据库供应商的一组应用程序接口来给特殊的数据库管理系统(DBMS)。用户的应用程序使用这组 API 来调度 ODBC 驱动程序。然后驱动程序通过 SQL 语句同 DBMS 发生联系。

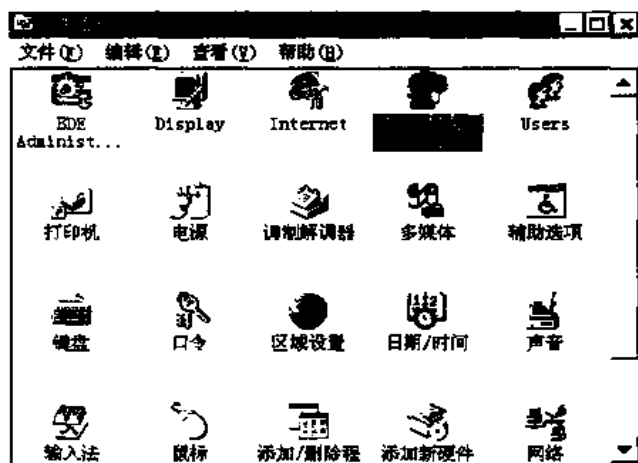


图 4.1 控制面板

下面我们就来介绍 ODBC 数据源。首先,用户必须保证在系统中已经安装了 32 位的 ODBC 数据源,如图 4.1 所示,如果用户未在系统中安装 ODBC 数据源,请先安装之。安装完

ODBC 数据源之后,双击打开 ODBC 数据源管理器,如图 4.2 所示。

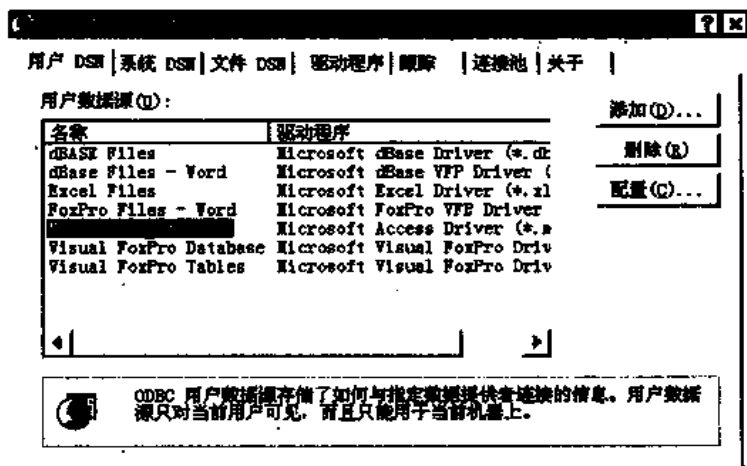


图 4.2 ODBC 数据源管理器

ODBC 数据源管理器为配置、添加、删除各种不同的数据源名称提供了最为简洁的方法。它使得用户可以轻松地完成数据源的配置工作,用户也可以删除原有的 DSN(数据源名称),或对先前配置的数据源进行修改。ODBC 数据源管理器的具体功能如下:

- 用户 DSN: ODBC 用户数据源存储了如何与指定数据提供者连接的信息。用户数据源只对当前的用户可见,而且只能应用在本机上。
- 系统 DSN: ODBC 系统数据源存储了如何与指定数据提供者连接的信息。系统数据源对当前机器上的所有用户可见。
- 文件 DSN: ODBC 文件数据源允许用户连接数据提供者。文件 DSN 可以由安装了相同驱动程序的用户共享。
- 驱动程序: ODBC 驱动程序允许那些支持 ODBC 的程序通过 ODBC 数据源获取信息。如果安装新的驱动程序,要使用其安装程序。
- 跟踪: ODBC 跟踪允许创建调用 ODBC 驱动程序的日志,以供技术支持人员查看;也可以辅助调试应用程序。Visual Studio 跟踪启动 Microsoft Visual Studio 的 ODBC 跟踪。
- 连接池: 连接池允许应用程序重新打开连接句柄,此操作将往返过程存入服务器。

4.2 Access 数据源的配置

Access 数据库所带的 Jet 数据库引擎具有很强的专用性。在早期的 Visual Basic 版本中,Access 的 Jet 数据库引擎是 Visual Basic 的专用引擎。虽然,Visual Basic 也可以通过其它的数据库引擎来和数据源相连接,但是由于 Jet 引擎的特殊性,使用它来和数据库相连在一定的程度上会提高系统的效率,使得数据库的存取更加方便和有效。对于那些使用 Visual Basic 来开发非客户机/服务器模式应用程序的用户来说,使用 Access 数据库无疑是一个明智的选择。下面我们讲述 Access 数据源的配置。

(1) 打开控制面板中的 ODBC 数据源管理器,如图 4.2 所示。在用户数据源项中选择 MS Access Database 选项。你可以删除或配置一个已有的用户数据源,在这里我们将要添加一个数据源。

(2) 单击添加按钮,系统将准备在 MS Access Database 下添加一个用户数据源。为了安装数据源,会弹出创建新数据源窗口,如图 4.3 所示。

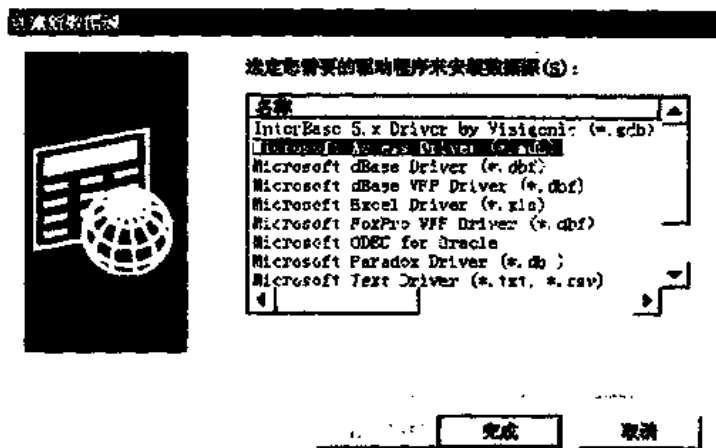


图 4.3 创建新数据源

(3) 在创建新数据源窗口中,我们将选择 Microsoft Access Driver 来安装数据源。

(4) 在图 4.3 中选择之后,单击“完成”按钮,进入 ODBC Microsoft Access 安装窗口,如图 4.4 所示。

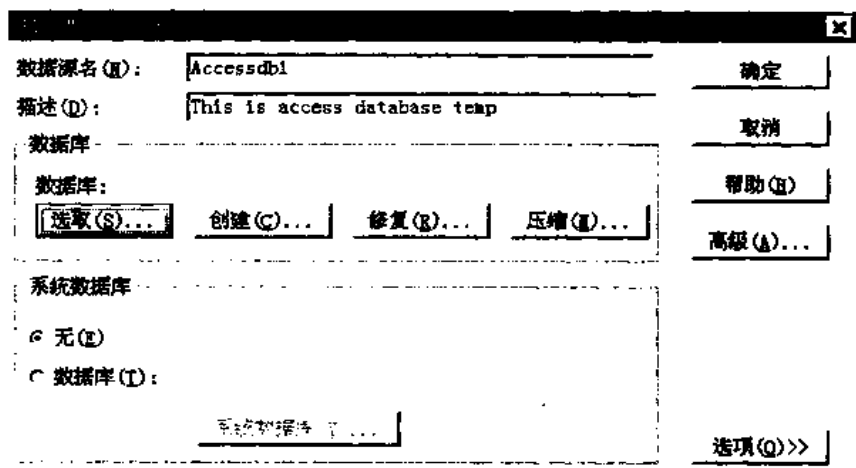


图 4.4 ODBC Microsoft Access 安装窗口

- 在“数据源名”项中键入数据源的名称。在这里我们键入 Accessdb1 作为数据源名。
- 在“描述”项中键入对数据源的描述。在这里我们键入 This is access database temp 的字样,当然,描述项并不是非要不可的。

- 在“数据库”项中可以选取、创建、修复或者压缩数据库。在这里我们选择“选取”按钮来选取相应的 Access 数据库。如果用户没有自己建立的 Access 数据库,那么用户可以到安装 Access 的目录下面选取 Access 自带的数据库。注意,Access 数据库是以.mdb 作为扩展名的。
- 最后,单击“确定”按钮。至此,一个新的 Access 数据源已经添加完成,如图 4.5 所示。

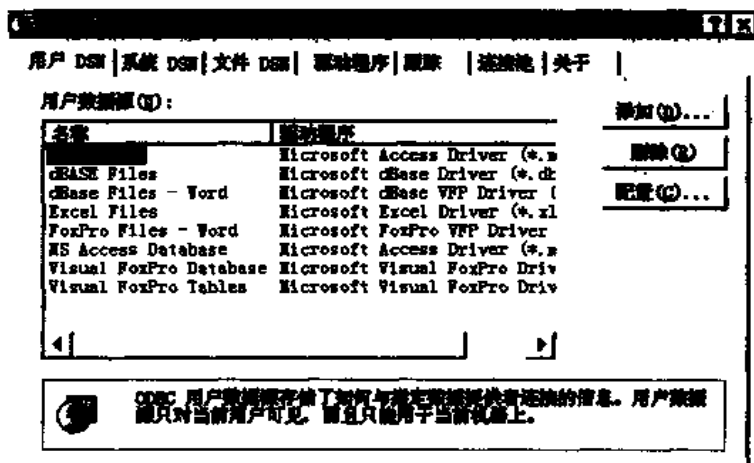


图 4.5 添加用户数据源完成

在这里要做一点说明,如果用户希望所创建的数据源名被本机中的多个用户所使用,请用用户添加系统 DSN。在实际情况中,开发一个单机单用户的数据库系统并没有什么意义,所以一般都需要对系统 DSN 进行配置。系统 DSN 的添加过程同用户 DSN 添加过程类似,读者可以动手到计算机上试一试。

4.3 SQL Server 数据源的配置

事实上,如果用户想要开发客户机/服务器模式的大型数据库应用程序,那么 Access 就会显得无能为力了。Visual Basic 可以通过 ADO 对象的桥梁作用和许多大型数据库相连接,如 Oracle、Sybase 等,当然也可以通过 ODBC 和其它的一些数据库相连。但是,作为 Microsoft 的系列产品,SQL Server 无疑是一个很好的选择。SQL Server 的高灵活性、高性能和多用户性决定了它作为 Visual Basic 后端数据库的可能性。因为 Microsoft 的 SQL Server 是一个服务器端的数据库源,所以 RDO 和 ADO 更适合于用来开发数据存取应用程序。另外,SQL Server 是第一个主要数据供应商,它提供本地的 OLE DB 数据存取,这就使得 ADO 成为越来越重要的存取 SQL Server 数据库的方法。下面,我们就来讲述如何来配置 SQL Server 数据源。

(1) 打开控制面板中的 ODBC 数据源管理器,如图 4.2 所示。单击“添加”按钮来添加一个数据源。

(2) 单击添加按钮后,系统将准备添加一个用户数据源。为了安装数据源,会弹出创建新数据源窗口,如图 4.6 所示。由于 Visual Basic 和 SQL Server 的系列性,所以即使用户并没有安装 SQL Server,在创建新数据源窗口中也列有 SQL Server 这一项。这正从一个侧面说明了我们当时为何要选用 SQL Server 来作为 Visual Basic 的后端数据库。当然,这里的列表项只

是起了一个标识作用,如果用户尚未安装 SQL Server,那么就请先安装上然后在进行以下的步骤。

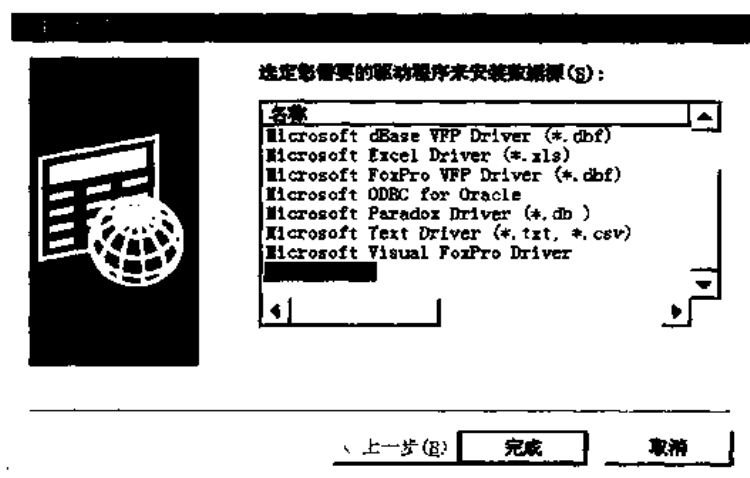


图 4.6 创建 SQL Server 数据源

(3) 在图 4.6 中选择 SQL Server 之后,单击“完成”按钮,进入“创建新的数据源到 SQL Server”窗口,如图 4.7 所示。

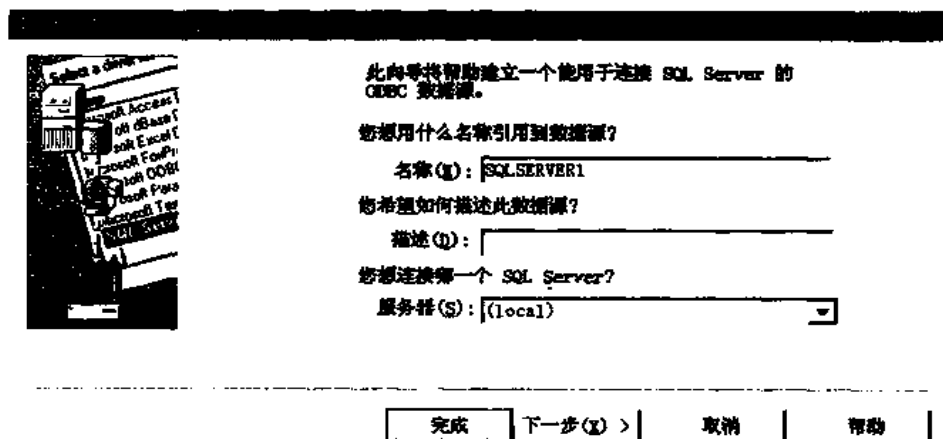


图 4.7 为 SQL Server 建立新的数据源

- 在“名称”项中键入新的数据源名,这里我们键入 SQLSERVER1 作为新的数据源名称。
- 在“描述”项中键入对数据源的描述。在这里我们并未键入任何描述性文字,因为描述项并不是非要不可的。
- 在“服务器”项中键入想连接的 SQL Server 服务器。如果要连接的 SQL Server 是安装在本地机上,那么就下拉框中选取 local,local 表示的是连接到本地的服务器。如果要连接的 SQL Server 是安装在其它的服务器上,那么就应从下拉框中选取相应的选项。如果从列表中选择服务器名称,则不需要更多的配置。如果输入一个不存在的服务器的名字作为 SQL Client Configuration Utility 中的一项,可在向导的下一个屏幕上

为新名字创建一个服务器别名。

(4) 单击“完成”按钮完成新数据源的配置。单击“下一步”按钮进行下一步的配置工作,如图 4.8 所示。

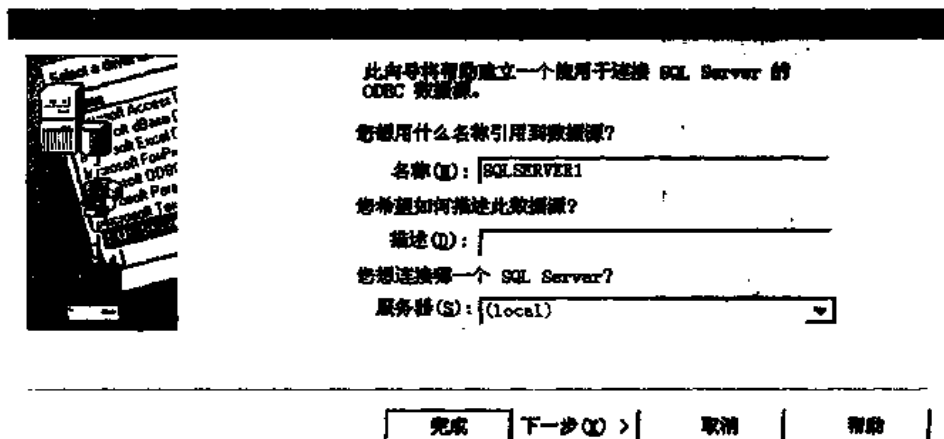


图 4.8 配置新的数据源

- 在“SQL Server 应该如何验证登录标识号的真伪?”下面有两个单选项供用户选择。
- 如果选定“使用用户输入登录标识号和密码的 SQL Server 验证”,请指定连接到 SQL Server 时 SQL Server 驱动程序使用的登录标识号。这仅适用于确定服务器默认配置的连接,而不适用于使用已创建数据源进行的后续连接。
- 如果选定“使用用户输入登录标识号和密码的 SQL Server 验证”,请指定连接到 SQL Server 时 SQL Server 使用的密码。这只适用于确定服务器默认配置的连接,而不适用于使用新数据源进行的后续连接。

(5) 单击“下一步”按钮将进行连接的验证。如果正确,则连接成功;如果不正确,系统会指出具体的错误,用户应该重新验证输入的正确性。

第五章 连接数据库实例

上一章我们已经学习了如何配置数据源。在配置完了数据源之后,摆在我们面前的就是怎么和一个后端数据库相连接的问题了。在 Visual Basic 中数据的操作共有三种方式:DAO、RDO 和 ADO,由于 ADO 几乎具有了前两种数据操作所共有的优点,所以在本书中我们一律采取 ADO 方式来访问数据库中的数据。读者可能已经想起,ADO 中包括 Connection、RecordSet、Command、Parameter、Field 和 Error 等对象。其中,Connection 对象为一个到数据源的连接提供上下文。

在本章中我们主要讲述以下两个部分:

- Connection 对象介绍
- 连接到数据库的实例

5.1 Connection 对象介绍

5.1.1 Connection 对象的属性

Connection 对象支持许多属性,可以利用这些属性来操作当前的连接状态或者获取一些基本的 Connection 对象的信息。有一些属性是只读的,而另一些属性是可读写的。下面我们就来介绍 Connection 对象的各种属性,在各种属性的介绍过程中,我们有时会插入一些简短的例子。

1. Attributes 属性

Attributes 属性设置或返回一个整型值,它用来指示对象的一项或多项特性。对于 Connection 对象,Attributes 属性为读/写,并且其值可能为以下任意一个或多个 XactAttributeEnum 值的和(默认为零)。

- AdXactCommitRetaining: 执行保留的提交,即通过自动调用 CommitTrans 启动新事务。并非所有提供者都支持该常量。
- AdXactAbortRetaining: 执行保留的中止,即通过自动调用 RollbackTrans 启动新事务。并非所有提供者都支持该常量。

2. CommandTimeout 属性

CommandTimeout 属性设置或返回长整型值,该值指示等待命令执行的时间(单位为秒)。默认值为 30,指示在终止尝试和产生错误之前执行命令期间需等待的时间。使用 Connection 对象或 Command 上的 CommandTimeout 属性,允许由于网络拥塞或服务器负载过重产生的延迟而取消 Execute 方法调用。如果在 CommandTimeout 属性中设置的时间间隔内没有完成命令执行,将产生错误,然后 ADO 将取消该命令。如果将该属性设置为零,ADO 将无限期等待直到命令执行完毕。

3.ConnectionString 属性

ConnectionString 属性设置或返回字符串值,这个字符串值包含用来建立到数据源的连

接的信息。使用 `ConnectionString` 属性,通过传递包含一系列由分号分隔的 `argument = value` 语句的详细连接字符串可指定数据源。ADO 支持 `ConnectionString` 属性的四个参数,任何其他参数将直接传递到提供者而不经 ADO 处理。

- **Provider** 参数:指定用来连接的提供者名称。
- **File Name** 参数:指定包含预先设置连接信息的特定提供者的文件名称。
- **Remote Provider** 参数:指定打开客户端连接时使用的提供者名称。
- **Remote Server** 参数:指定打开客户端连接时使用的服务器的路径名称。

【例 5-1】 使用 Connection 对象的 ConnectionString 属性

```
Public Sub ConnectionString_Example()  
    Dim connection1 As ADODB.Connection  
    Dim connection2 As ADODB.Connection  
    Dim connection3 As ADODB.Connection  
    Dim connection4 As ADODB.Connection  
    ' 不使用数据源名 (DSN) 打开连接。  
    Set connection1 = New ADODB.Connection  
    connection1.ConnectionString = "driver={SQL Server};" & _  
        "server=Myserver;uid=sa;pwd=password;database=pubs"  
    connection1.ConnectionTimeout = 30  
    connection1.Open  
    ' 使用 DSN 和 ODBC 标记打开连接。  
    Set connection2 = New ADODB.Connection  
    connection2.ConnectionString = "DSN=Pubs;UID=sa;PWD=password;"  
    connection2.Open  
    ' 使用 DSN 和 OLE DB 标记打开连接。  
    Set connection3 = New ADODB.Connection  
    connection3.ConnectionString = "Data Source=Pubs;User ID=sa;Password=password;"  
    connection3.Open  
    ' 使用 DSN 和单个参数而非连接字符串打开连接。  
    Set connection4 = New ADODB.Connection  
    connection4.Open "Pubs", "sa", "pwd"  
End Sub
```

注意,如果没有复选“引用”对话框中的 Data Access Object,程序就会出现错误。在上面的程序段中,先定义了 4 个 Connection 对象的变量。这 4 个变量分别对应于 4 种不同的连接方式。语句 `Set connection1 = New ADODB.Connection` 是创建一个新的 Connection 对象,然后再调用 Connection 对象的 `Open` 方法来打开数据库连接。关于 Connection 对象的方法我们将在下一小节中介绍。

4. ConnectionTimeout 属性

`ConnectionTimeout` 属性设置或返回指示等待连接打开的时间的长整型值(单位为秒)。其默认值为 15,指示在终止尝试和产生错误前建立连接期间所等待的时间。如果由于网络拥塞或服务器负载过重导致的延迟使得必须放弃连接尝试时,请使用 Connection 对象的 `ConnectionTimeout` 属性。如果打开连接前所经过的时间超过 `ConnectionTimeout` 属性上设置的时间,将产生错误,并且 ADO 将取消该尝试。如果将该属性设置为零,ADO 将无限等待直到

连接打开。

5. DefaultDatabase 属性

DefaultDatabase 属性可设置或返回指定 Connection 对象上默认数据库的名称。

【例 5-2】 使用 Connection 对象的 DefaultDatabase 属性。

```
Public Sub DefaultDatabase_Example()  
    Dim connection1 As ADODB.Connection  
    Set connection1 = New ADODB.Connection  
    connection1.ConnectionString = "driver={SQL Server};" & _  
        "server=Myserver;uid=sa;pwd=password"  
    cnn1.Open  
    cnn1.DefaultDatabase = "pubs"  
End Sub
```

6. IsolationLevel 属性

IsolationLevel 属性指出 Connection 对象如何处理对象。

7. Mode 属性

Mode 属性设置或返回以下某个 ConnectModeEnum 的值,指示用于更改在 Connection 中的数据的数据的可用权限。

- AdModeUnknown: 默认值。表明权限尚未设置或无法确定。
- AdModeRead: 表明权限为只读。
- AdModeWrite: 表明权限为只写。
- AdModeReadWrite: 表明权限为读/写。
- AdModeShareDenyRead: 防止其他用户使用读权限打开连接。
- AdModeShareDenyWrite: 防止其他用户使用写权限打开连接。
- AdModeShareExclusive: 防止其他用户打开连接。
- AdModeShareDenyNone: 防止其他用户使用任何权限打开连接。

8. Provider 属性

Provider 属性指出当前数据提供者的名字,或者是使用 Open() 方法时没有指定名字的情况下所使用的提供者名。但是,调用 Open 方法时如果在多处指定提供者可能会产生无法预料的后果。如果没有指定提供者,该属性将默认为 MSDASQL (Microsoft OLE DB Provider for ODBC)。

【例 5-3】 使用 Connection 对象的 Provider 属性。

```
Public Sub Provider_Example()  
    Dim connection1 As ADODB.Connection  
    Set connection1 = New ADODB.Connection  
    connection1.Provider = "Microsoft.Jet.OLEDB.3.51"  
    connection1.Open "C:\Samples\northwind.mdb", "admin", ""  
End Sub
```

9. State 属性

State 属性对所有可应用对象都可用,它用来说明其对象状态是打开的或关闭的。可以随时使用 State 属性来确定指定对象的当前状态。该属性是只读的,并返回下列常量之一的长整

数型值。

- AdStateClosed: 默认值, 指示对象是关闭的。
- AdStateOpen: 指示对象是打开的。
- AdStateConnecting: 指示 Recordset 对象正在连接。
- AdStateExecuting: 指示 Recordset 对象正在执行命令。
- AdStateFetching: 指示 Recordset 对象的行正在被读取。

【例 5-4】 使用 Connection 对象的 State 属性。

```
Public Sub State_Example()  
    Dim connection1 As ADODB.Connection  
    Dim statestring AS String  
    Set connection1 = New ADODB.Connection  
    connection1.ConnectionString = "DSN=Pubs;UID=sa;PWD=password;"  
    connection1.Open  
    Select Case connection1.State  
        Case adStateClosed  
            statestring = "adStateClosed"  
        Case adStateOpen  
            statestring = "adStateOpen"  
    End Select  
    ' 显示连接的状态。  
    MsgBox "connection1.State: ", , statestring  
End Sub
```

5.1.2 Connection 对象的方法

Connection 对象的方法用来管理事务、执行命令、打开和关闭连接。有一点读者应该注意, 即 ADO 对象所支持的方法是独立于当前所使用的数据源的。例如一个 OLE DB 数据源不必支持 OLE DB 规范的全部功能。下面我们就来介绍 Connection 对象的各种方法, 与属性介绍相类似, 在方法的介绍过程中, 我们也会插入一些具体的使用例子。

1. BeginTrans 方法

BeginTrans 方法开始一个连接到数据源的新事务, 即启动一个新的事务。与之相应的还有 CommitTrans 和 RollbackTrans 方法。CommitTrans 保存所有的更改并结束当前事务, 它也可以用来启动一个新的事务。RollbackTrans 取消当前事务中所做的任何更改并结束事务, 同 CommitTrans 类似, 它也可以用来启动一个新的事务。它们的语法为:

```
Object.BeginTrans  
Object.CommitTrans  
Object.RollbackTrans
```

需要注意的是, 并非所有提供者都支持事务。需验证提供者定义的属性 Transaction DDL 是否出现在 Connection 对象的 Properties 集合中, 如果在则表示提供者支持事务。如果提供者不支持事务, 调用其中的某个方法将返回错误。一旦调用了 BeginTrans 方法, 在调用 CommitTrans 或 RollbackTrans 结束事务之前提供者将不再立即提交所做的任何更改。

【例 5-5】 使用 Connection 对象的 BeginTrans、CommitTrans 和 RollbackTrans 方法。

```
Public Sub BeginTrans_Example( )
    Dim cnn1 As ADODB.Connection
    Dim rstTitles As ADODB.Recordset
    Dim strCnn As String
    Dim strTitle As String
    Dim strMessage As String
    ' 打开连接。
    strCnn="Provider=sqloledb;Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    Set cnn1 = New ADODB.Connection
    cnn1.Open strCnn
    ' 打开 Titles 表。
    Set rstTitles = New ADODB.Recordset
    rstTitles.CursorType = adOpenDynamic
    rstTitles.LockType = adLockPessimistic
    rstTitles.Open "titles", cnn1, ., adCmdTable
    rstTitles.MoveFirst
    cnn1.BeginTrans
    ' 在记录集中循环并询问是否想要更改指定标题的类型。
    Do Until rstTitles.EOF
        If Trim(rstTitles!Type) = "psychology" Then
            strTitle = rstTitles!Title
            strMessage = "Title: " & strTitle & vbCr & "Change type to self help?"
            ' 更改指定雇员的标题。
            If MsgBox(strMessage, vbYesNo) = vbYes Then
                rstTitles!Type = "self_help"
                rstTitles.Update
            End If
        End If
        rstTitles.MoveNext
    Loop
    ' 询问用户是否想提交以上所做的全部更改。
    If MsgBox("Save all changes?", vbYesNo) = vbYes Then
        cnn1.CommitTrans
    Else
        cnn1.RollbackTrans
    End If
    ' 打印记录集中的当前数据。
    rstTitles.Requery
    rstTitles.MoveFirst
    Do While Not rstTitles.EOF
        Debug.Print rstTitles!Title & " - " & rstTitles!Type
        rstTitles.MoveNext
    Loop
    ' 恢复原始数据,因为这只是演示。
    rstTitles.MoveFirst
    Do Until rstTitles.EOF
```

```
    If Trim(rstTitles!Type) = "self-help" Then
        rstTitles!Type = "psychology"
        rstTitles.Update
    End If
    rstTitles.MoveNext
Loop
rstTitles.Close
cnn1.Close
End Sub
```

该例子更改了数据库中 Titles 表里所有心理学书籍的类型。在 BeginTrans 方法启动事务将所有对 Titles 表的更改隔离后,CommitTrans 方法将保存更改。可使用 Rollback 方法撤销用 Update 方法保存的更改。

2. Open 方法

Open 方法打开一个数据源的连接。它的语法如下:

```
connection.Open ConnectionString,UserID,Password,OpenOptions
```

参数说明:

- ConnectionString: 这是一个可选的字符串,它包含了连接信息。
- UserID: 可选的字符串,它包含了建立连接时所使用的用户名称。
- Password: 可选的字符串,它包含了建立连接时所使用的密码。
- OpenOptions: 可选的字符串,它包含了建立连接时所使用的连接方式。

3. Close 方法

Close 方法关闭一个到数据源的连接。在关闭的同时,此连接所使用的任何资源都会被释放。它的语法如下:

```
Object.Close
```

【例5-6】 使用 Connection 对象的 Open 和 Close 方法。

```
Public Sub OpenClose_Example()  
    Dim conn As ADODB.Connection  
    ' Open the Connection object,omit the password  
    conn.Open "Data Source = MyDatabase","john"  
    ' Use the Connection object here...  
    ' Close the Connection  
    conn.Close  
End Sub
```

4. Execute 方法

Execute 方法用来执行查询或由数据源支持的其他命令,并且返回一个 RecordSet 对象。它的语法如下:

```
Set recordset = connection.Execute (CommandText,RecordsAffected,Options)
```

参数说明:

- CommandText: 这是一个包含要执行的 SQL 语句、表名、存储过程或特定提供者的文本的字符串。
- RecordsAffected: 这是一个可选的长整型变量。提供者向其返回操作所影响的记录数

目。

- Options: 这是一个可选的长整型变量。它指示提供者应如何为 CommandText 参数赋值。

【例5-7】 使用 Connection 对象的 Execute 方法。

```
Public Sub Execute_Example()  
    Dim cnn As ADODB.Connection  
    Dim MyRecordset As ADODB.Recordset  
    Dim Mysql As String  
    Set cnn = New ADODB.Connection  
    Set MyRecordset = New ADODB.Recordset  
    cnn.Open "Data Source = MyDatabase","john"  
    Mysql = "Select * From Books "  
    MyRecordset = cnn.Execute Mysql  
    cnn.Close  
    MyRecordset.Close  
End Sub
```

5. OpenSchema 方法

OpenSchema 方法从提供者中获取数据库纲要信息。它的语法如下:

```
Set recordset = connection.OpenSchema (QueryType, Criteria, SchemaID)
```

参数说明:

- QueryType: QueryType 参数指出所要运行的纲要查询类型。
- Criteria: 这是一个可选的参数,它指出每个 QueryType 选项的查询限制条件数组。
- SchemaID: OLE DB 规范没有定义提供者纲要查询的 GUID。如果 QueryType 设置为 adSchemaProviderSpecific,则需要该参数,否则不使用它。

【例5-8】 使用 Connection 对象的 OpenSchema 方法。

```
Public Sub OpenSchemaX()  
    Dim cnn1 As ADODB.Connection  
    Dim rstSchema As ADODB.Recordset  
    Dim strCnn As String  
    Set cnn1 = New ADODB.Connection  
    strCnn="Provider=sqloledb;Data Source=srv;Initial Catalog=pubs;User Id=sa;Password= ;"  
    cnn1.Open strCnn  
    Set rstSchema = cnn1.OpenSchema(adSchemaTables)  
    Do Until rstSchema.EOF  
        Debug.Print "Table name: " & rstSchema!TABLE_NAME & vbCr & "Table type: " &  
            rstSchema!TABLE_TYPE & vbCr  
        rstSchema.MoveNext  
    Loop  
    rstSchema.Close  
    cnn1.Close  
End Sub
```

该范例使用 OpenSchema 方法显示 Pubs 数据库内每个表的名称和类型。

5.2 连接到数据库的实例

下面我们就给出一个连接到具体数据库的例子。在这个例子中,我们所用到的数据库为 Microsoft Office 系列中的 Access 2000。首先,我们打开 Access 2000,在新建数据库中选择“空 Access 数据库”项,如图 5.1 所示:



图 5.1 新建 Access 数据库

在选择了“空 Access 数据库”项以后,Access 会为之创建一个新的 Access 数据库,如图 5.2 所示,用户应该在“文件名”项后键入给新数据库起的名称。在这里,我们给新的数据库命名为 Access_db,它代表这是一个 Access 数据库。

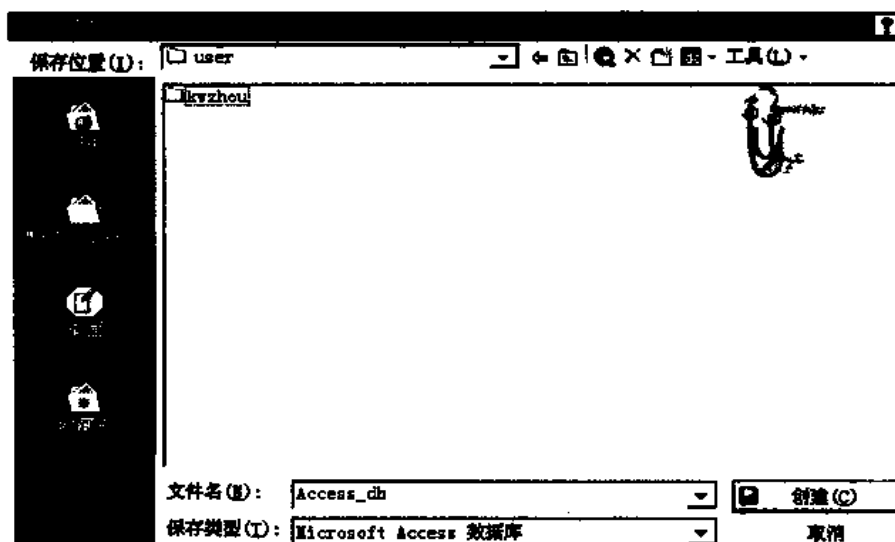


图 5.2 保存 Access 数据库

在单击“创建”按钮后,系统就生成了一个新的 Access 空数据库。在空数据库中,还没有任何用户所创建的表。在接下来的步骤中,我们就要给 Access_db 数据库建立表。在 Access 2000

中,共有三种创建表的方式,它们分别为使用设计器创建表、使用向导创建表、通过输入数据创建表。如图5.3所示,用户可以根据自己的喜好来选择其中的任何一种方式。在下面我们将采取第一种方式即使用设计器来创建表。

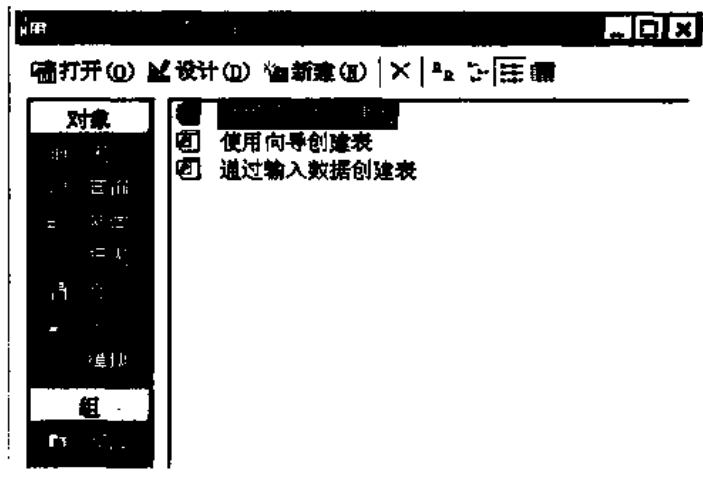


图5.3 给 Access 数据库创建表

下面,我们就要为 Access_db 创建一个名为“电话号码”的表。该表是由以下的6个表项所组成。

(1) 电话号码表项

- 数据类型: 文本。
- 字段大小: 50。
- 有效性规则: 无。
- 有效性文本: 无。
- 必填字段: 否。
- 允许空字符串: 否。
- 索引: 有(无重复)。

(2) 姓名表项。姓名表项的具体定义规则同电话号码表项。

(3) 地址表项。地址表项的其它具体定义规则同电话号码表项,只把其中的索引一项改为“无”。

(4) 安装时间表项。安装时间表项的具体定义规则同地址表项。

(5) 权限表项。权限表项的其它具体定义规则同地址表项,只把其中的必填字段一项改为“是”。

(6) 备注表项。

- 数据类型: 备注。
- 有效性规则: 无。
- 有效性文本: 无。
- 必填字段: 否。
- 允许空字符串: 是。

在输入完了电话号码表的各个表项之后应如图5.4所示。关闭表设计器,这时 Access 会提

示你是否要进行保存表,选择“是”。然后给表起名为“电话号码”。如果没有定义主键(Primary Key),Access 会提示你给表加上一个主键。

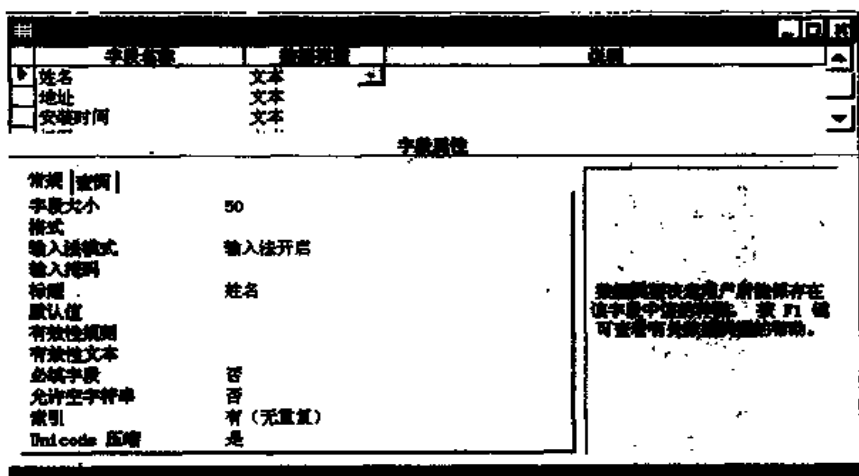


图5.4 电话号码表

在完成了表的设计之后,我们就会看到在 Access_db 数据库中出现一个名为“电话号码”的表,如图5.5所示。有了数据库和表之后,我们就可以按照上一章所述的具体步骤来配置 ODBC 数据源了。只有在配置完成了数据源之后,在 Visual Basic 的应用程序中才可以通过 ADO 对象来进行具体数据的操作。这一点请读者要千万注意,如果在建立了数据库之后没有对数据源进行配置或者对数据源的配置工作出现错误,那么我们在此之前所做的一切努力都是白费的。因为它根本就无法同数据库进行正确的连接工作,也就谈不上通过 Visual Basic 的应用程序来对数据库中的数据进行操作了。从中我们可以看到,连接数据库和上一章讲述的对数据源的配置之间的关系是多么的密切,两者之中任何一个出现问题,都会导致最后无法正确操作数据库中的数据。

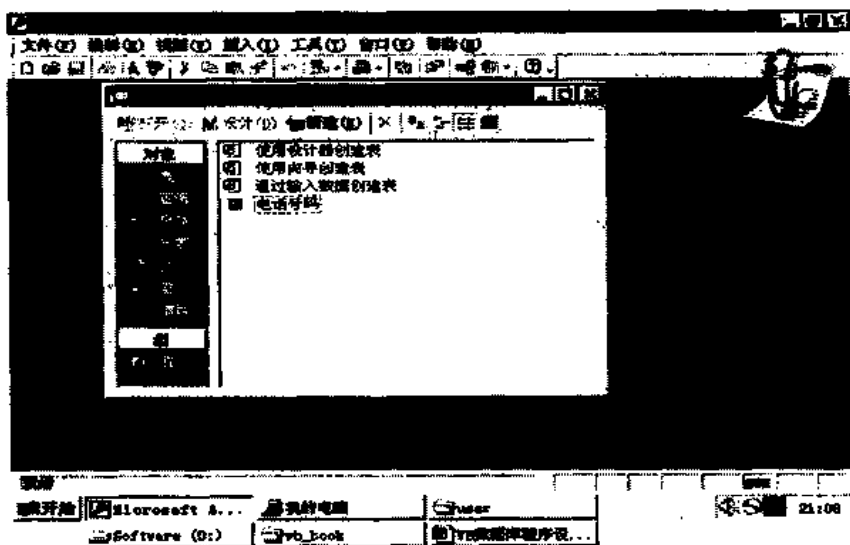


图5.5 建立了电话号码表的 Access db 数据库

接下来,我们就要进入 Visual Basic 的开发环境中去具体地连接到 Access.db 数据库。进入 Visual Basic 开发环境后,新建一个标准的 Exe 工程。在工程的 Form1 中我们键入如下的代码:

```
Private Sub Form_Load()  
    Dim cnn As ADODB.Connection  
    Dim my_recordset As ADODB.Recordset  
    Dim connect_string As String  
    Dim statestring As String  
    Set cnn = New ADODB.Connection  
    Set my_recordset = New ADODB.Recordset  
    ' 连接 Access 数据库。  
    connect_string = "DSN=Access.db;UID=;PSW=" <img alt="A small graphic of a person with arms raised in a celebratory gesture." data-bbox="485 325 515 355"/>  
    cnn.Open connect_string  
    Select Case cnn.State  
        Case adStateClosed  
            statestring = "adStateClosed"  
        Case adStateOpen  
            statestring = "adStateOpen"  
    End Select  
    ' 显示连接的状态。  
    MsgBox "connection1.State: ", , statestring  
    ' 对电话号码表进行查询操作。  
    my_recordset.Open "Select * from 电话号码", cnn  
    my_recordset.Close  
    cnn.Close  
End Sub
```

连接数据库的代码是写在 Form_Load() 子过程中的,这就说明了对数据库的连接工作是在程序一开始,即窗体的加载过程中进行的。在 Form_Load() 中,首先定义了一个 Connection 对象 cnn 和一个 RecordSet 对象 my_recordset,以及两个字符串类型的变量 connect_string 和 statestring。接下来的两个 Set 语句是为了生成一个 Connection 对象 cnn 和一个 RecordSet 对象 my_recordset。然后是连接 Access 数据库,先把连接字符串“DSN=Access.db;UID=;PSW=”赋给字符串变量 connect_string。连接字符串中的具体各项内容是根据 ODBC 数据源的配置而定的,不同的配置方式和配置内容就会有不同的连接字符串。cnn 的 Open 事件用来执行与数据库的连接工作。为了验证与数据库的连接工作是否正确,在这里我们读取了 cnn 的 State 属性。正如本章中的第一节所讲到的,State 属性可以用来说明其对象状态是打开的或关闭的。在使用了 State 属性之后,通过系统函数 MsgBox 把与数据库的连接状态显示给用户。在显示之后的语句 my_recordset.Open “Select * from 电话号码”,cnn 则示范了如何从数据库表中查询所需要的数据。

在这里读者需要注意的是,在程序中每当用一个 Set 语句来生成一个 Connection 或 RecordSet 对象时,最后使用完毕之后应该对每一个 Connection 对象或 RecordSet 对象进行关闭操作。关闭操作是使用对象的 Close 方法。

第六章 使用 Data 控件

Visual Basic 内在的 Data 控件同 Microsoft Access 所用的数据库引擎相同,它是通过使用 Microsoft 的 Jet 数据库引擎来实现数据访问。这一技术使用户可以无缝地访问很多标准的数据库格式,而且使用户无需编写任何代码就可以创建数据库的应用程序,例如数据库表中数据的显示。这种内在的 Data 控件最适合一些小型的数据库,诸如 Access 和 ISAM 数据库等。除了 Data 控件之外,在 Visual Basic 中还有 ADO Data 控件和 Remote Data 控件。这三种不同类型的控件却具有很强的相似性,它们都是将一个数据源连接到一个数据绑定控件的“数据控件”。并且,从控件的外观上来看,它们也非常相似。为了相互比较起见,在本章中我们就分别讲述这三种控件。

在本章中我们主要讲述以下几个部分:

- Data 控件的实例
- ADO Data 控件的实例
- RemoteData 控件

6.1 Data 控件的实例

Data 控件是 Visual Basic 的内部控件之一,即 Visual Basic 运行以后在开发环境的工具箱内就能见到并直接使用 Data 控件。正如前面所讲的,利用 Data 控件我们可以做到不写任何程序代码或者写少量的代码就能够实现对数据库中的数据进行查询、添加、删除、更新等一系列基本的数据库操作,如图6.1所示。

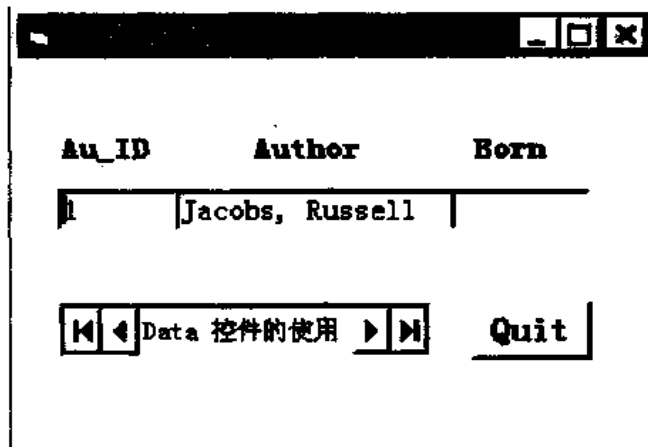


图6.1 Data 控件的使用

在下面这个 Data 控件使用的实例中,我们所用到的是系统自带的数据库 Biblio.mdb。这个 Access 数据库中一共有四个表,它们分别是 Authors、Publishers、Title Author 和 Titles。在本例中我们将用到的只是 Authors 表。

此程序是使用 Data 控件来检索 Biblio 数据库 Authors 表中的具体数据。首先,我们新建一个标准的 Exe 工程,如图 6.2 所示,并把 Form1 的 Caption 属性改为使用 Data 控件。

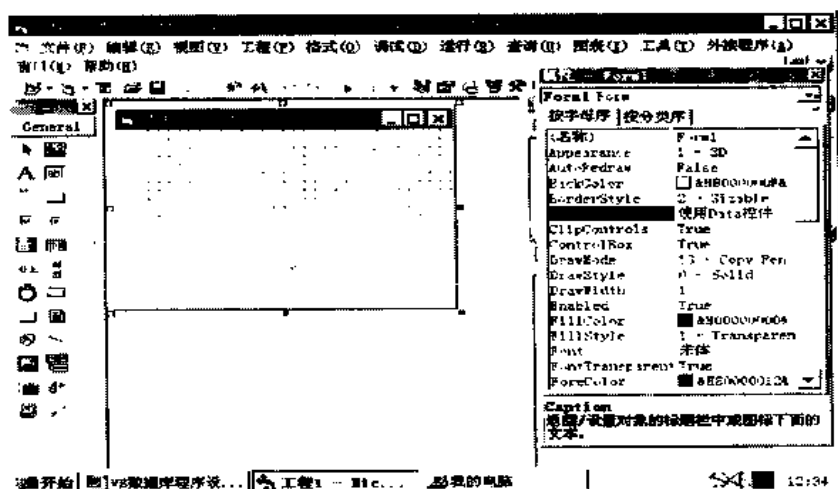


图 6.2 新建标准 Exe 工程

运行该工程,你将看到一个名为“使用 Data 控件”的窗体。该窗体自动地含有 Windows 标准的极大、极小和退出按钮。从这里读者应该认识到,即使什么工作都不做,Visual Basic 也会为应用程序创建一个标准的 Windows 窗口。

在 Form1 中加入一个 Data 类型的控件,调节 Data 控件的大小使它如图 6.3 所示。默认时,Data 控件的 Caption 属性为 Data1,在这里我们把它改为使用 Data 控件。接着,要改变以下三个属性的值。

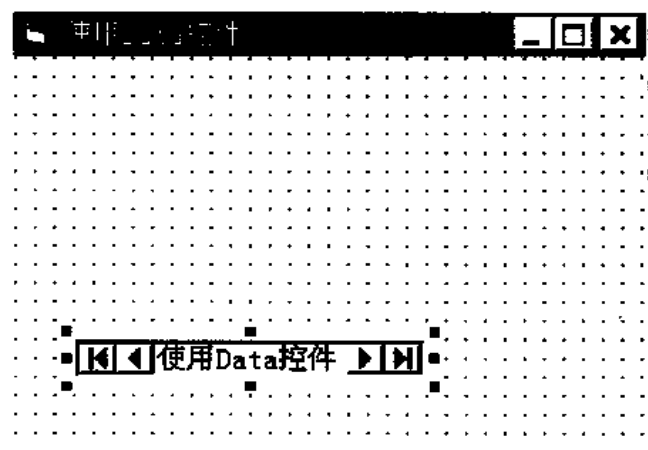


图 6.3 加入一个 Data 控件

(1) Connect 属性

Data 控件的 Connect 属性指示打开的数据库源、浏览查询中使用的数据库或附加表。在 Connect 属性的下拉框中我们可以见到有许多种的数据库类型,在这里我们选择 Access,表示连接的是 Access 的数据库。

(2) DatabaseName 属性

DatabaseName 属性设置一个数据控件的数据源的名称和位置。用户应该在这个属性后面输入所要用的数据库的具体位置。在这里我们输入 D:\Program Files\Microsoft Visual Studio\VB98\Biblio.mdb, 表示 Biblio 数据库的确切存储位置。

(3) RecordSource 属性

RecordSource 属性设置一个数据控件的底层表、SQL 语句或 QueryDef 对象。打开 RecordSource 的下拉框, 我们会发现它包含了 Biblio 数据库中的所有表, 在这里我们选择 Authors 表, 表示本应用程序所要用到的数据来源于 Authors 表中。

在 Form1 中加入一个 TextBox 控件和一个 Label 控件, 调节控件的大小并且使它们与 Data 控件对齐。改变 Label 控件的 Caption 属性为 Au_ID, 并且清空 TextBox 控件的 Text 属性。完成操作后如图 6.4 所示。

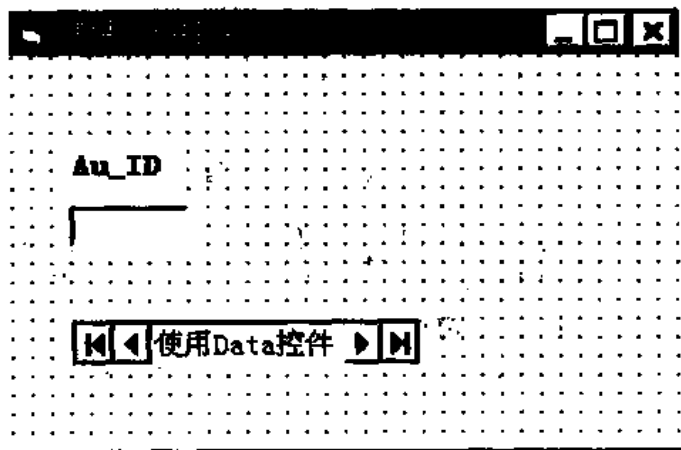


图6.4 加入 TextBox 和 Label 控件

要使得 TextBox 控件通过 Data 控件与具体的数据库中的数据相联系, 就要改变 TextBox 控件的以下两个属性的值。

(1) DataSource 属性

DataSource 属性用来设置一个数值, 它指出数据控件通过它将当前控件绑定到数据库中去。在这里需要注意的是, 用户不能在程序的运行过程中将一个控件或对象的 DataSource 属性设置为 Data 控件或者 Remote Data 控件。如下面的代码将会引起失败:

```
Set Text1.DataSource = Data1
```

但是, 你可以这样做:

```
Set Text1.DataSource = ADODC1
```

在这里, 我们把 DataSource 的属性值选为 Data1, 表示它是通过 Data1 这个已经存在的控件来和数据库中的具体数据相联系的。

(2) DataField 属性

DataField 属性用来设置一个值, 它将控件绑定到当前记录的一个字段中去, 并保证设置的 DataField 属性对每个被绑定的控件都是合法的。如果改变 Data 控件的 RecordSource 属性的设置, 然后再使用 Refresh, 则 Recordset 将标识该新对象。这可能会使被绑定的控件的

DataField 设置失效,并产生一个可以捕获的错误。在这里,我们把 DataField 的属性值选为 Au_ID,它表明了 TextBox1对应的是 Authors 表中的 Au_ID 项。

在 Form1中再加入两个 TextBox 控件和两个 Label 控件,调节它们的大小,改变 Label 控件的 Caption 属性,并且清空 TextBox 控件的 Text 属性。完成操作后如图6.5所示。

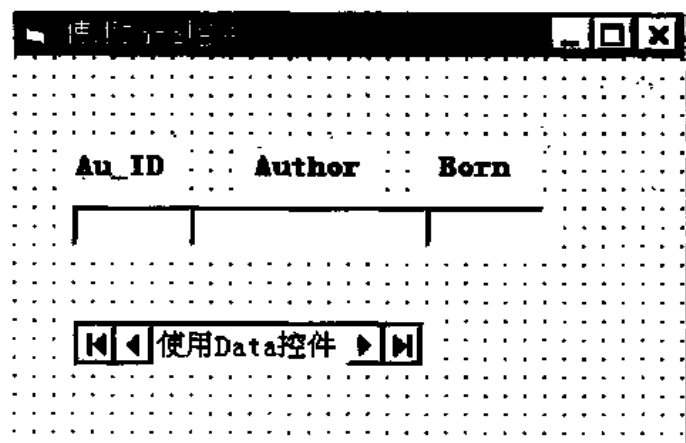


图6.5 再加入 TextBox 和 Label 控件

相应地,使 TextBox2的 DataSource 属性为 Data1,DataField 属性为 Author。使 TextBox3的 DataSource 属性为 Data1,DataField 属性为 Year Born。这样,我们就通过 Data1 把 Biblio 数据库中 Authors 表的内容同三个具体的 TextBox 控件联系起来了。

最后,我们要在 Form1上添加一个退出控件。虽然,用户可以通过使用标准 Windows 窗口中的退出按钮来退出应用程序,但是,为了程序的完整性考虑,我们还是加上了一个退出控件。在工具箱中选择 CommandButton,把它的 Caption 属性改为 Quit。完成后如图6.6所示。

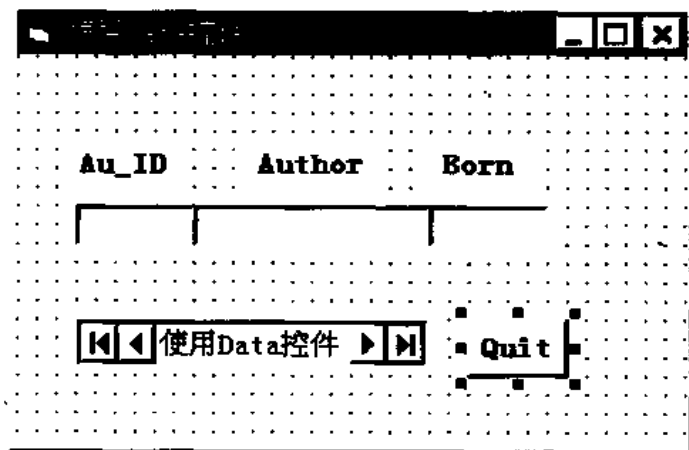


图6.6 加入 Quit 控件

为了在程序执行过程中单击“Quit”控件就能退出应用程序,应该加入如下的子程序段。在加入了代码之后,我们就可以运行应用程序了。

```
Private Sub Command1_Click()  
    End  
End Sub
```

6.2 ADO Data 控件的实例

ADO Data 控件使用 Microsoft ActiveX 数据对象(ADO)来快速建立数据绑定的控件和数据提供者之间的连接。尽管可以在应用程序中直接使用 ActiveX 数据对象,但 ADO Data 控件有作为一个图形控件的优势(具有“向前”和“向后”按钮),以及一个易于使用的界面,使用户可以用最少的代码创建数据库应用程序。数据绑定控件是任何具有“数据源”属性的控件。在 Visual Basic 的“工具箱”中不少控件都可以作为数据绑定的控件,包括复选框、组合框、图像、标签、列表框、图片框及文本框控件等。此外,Visual Basic 还包括了若干种数据绑定的 ActiveX 控件,诸如 DataGrid、DataCombo、Chart 及 DataList 控件等。用户也可以创建自己的数据绑定的 ActiveX 控件,或从其他开发商购买控件。数据提供者可以是任何符合 OLEDB 规范的数据源。使用 Visual Basic 的类模块也可以很方便地创建子集的数据提供者。

在下面的程序示例中,我们将使用两个 ADO Data 控件。并且,我们还将使用一个 DataGrid 控件和一个 DataList 控件。使一个 ADO Data 控件和 DataList 相联系,另一个和 DataGrid 相联系。程序完成后执行时如图 6.7 所示。

DataGrid 控件显示并允许对 Recordset 对象中代表记录和字段的一系列行和列进行数据操作,它是一种数据绑定的 ActiveX 控件。DataGrid 控件的每一个单元格都可以包含文本值,但不能链接或内嵌对象。可以在代码中指定当前单元格,或者用户可以使用鼠标或箭头键在运行时改变它。通过在单元格中键入或编程的方式,单元格可以交互地编辑。单元格能够被单独地选定或按照行来选定。但在本例子中,DataGrid 控件只是用来作为数据显示,而并不对它进行编辑或修改。

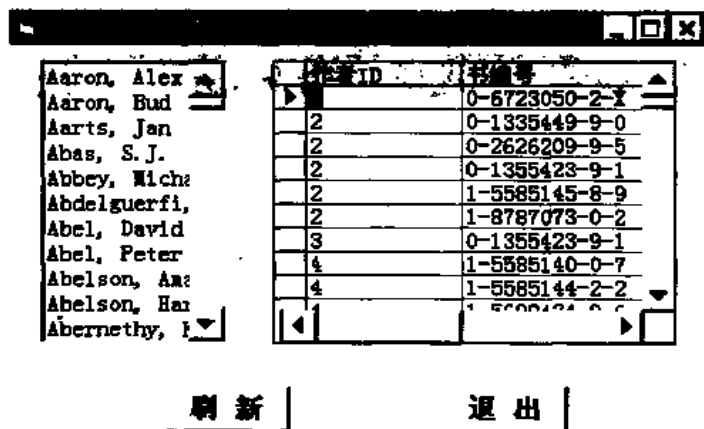


图 6.7 使用 ADO Data 控件

在 DataList 控件中,我们列出的是 Authors 表中 Author 表项。在右边的 DataGrid 控件中,所列出的是 Title Author 表中的表项。当用户单击 DataList 控件中的某一具体项时,DataGrid 控件中会自动地列出该作者所撰写的书目。两者之间的联系是通过 Authors 表和 Title Author 表所共有的表项 Au_ID 进行的。单击“刷新”按钮时,系统会在 DataList 和 DataGrid 控件中分别列出数据库相对应的表中所有的数据。

下面,我们就来逐步地完成该程序。首先新建一个标准的 Exe 工程,并把工程中 Form1 的 Caption 属性设置为使用 ADO Data 控件,这说明该应用程序是一个使用 ADO Data 控件的例子,如图 6.8 所示。

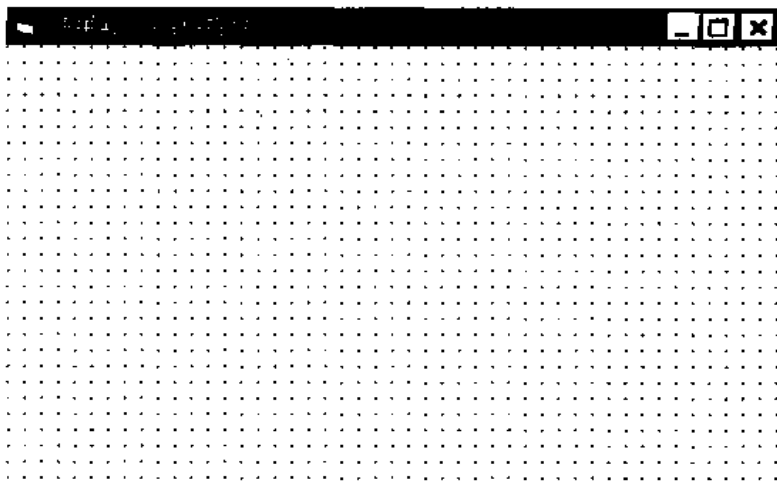


图 6.8 使用 ADO Data 控件的窗口

在窗体中加入两个 ADO Data 控件。如果在工具箱中找不到 ADO Data 控件,可以在工具箱上单击鼠标右键,选择“部件...”菜单项,然后在“部件”窗口的“控件”选项卡中选中 Microsoft ADO Data Control 6.0 (OLEDB)。完成了 ADO Data 控件的选择之后,我们再往 Form1 上加入一个 DataList 控件和一个 DataGrid 控件。类似地,如果在工具箱中不存在这两个控件,可以在“部件”窗口的“控件”选项卡中选中 Microsoft DataList Control 6.0 (OLEDB) 和 Microsoft DataGrid Control 6.0 (OLEDB)。调整控件的大小及位置,完成后如图 6.9 所示。

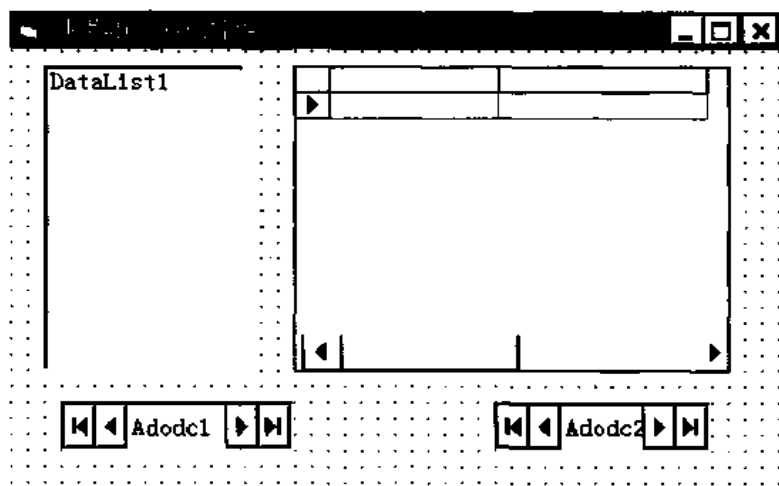


图 6.9 加入控件之后的窗体

接下来,我们要改变和设置这几个控件的一些属性值。

对 Adodc1 来说,我们改变如下几个属性的值:

(1) Name 属性

改变 Name 属性为 Ado_AuID。

(2) ConnectionString 属性

ConnectionString 属性表示 Adodc1 的连接字符串, 可以通过三种方法来设置连接字符串属性。如图 6.10 所示, 在这里我们选择“使用 ODBC 数据资源名称”, 并在下拉框中选择 Biblio 数据库。单击确定按钮, 用户就可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

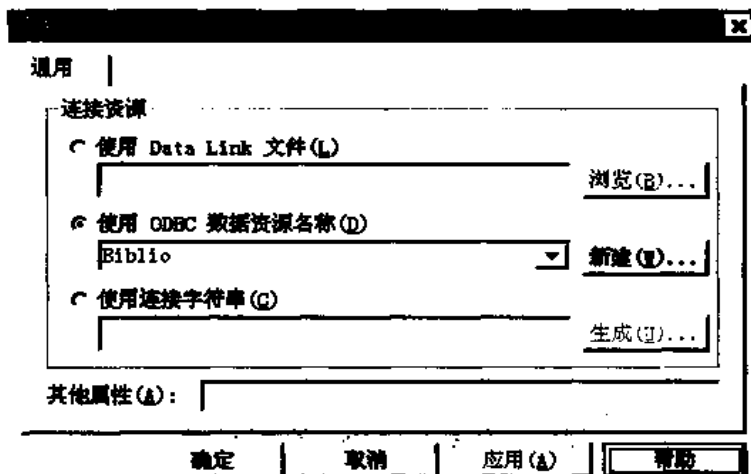


图 6.10 设置 ConnectionString 属性

(3) RecordSource 属性

RecordSource 属性设置 RecordSet 源。在这里, 我们将在属性页的命令文本下键入如下的 SQL 语句:

```
Select * From Authors  
Order By Author
```

属性页如图 6.11 所示。单击确定按钮, 用户就可以在 RecordSource 属性中看到“Select * From Authors Order By Author”的字符串形式。

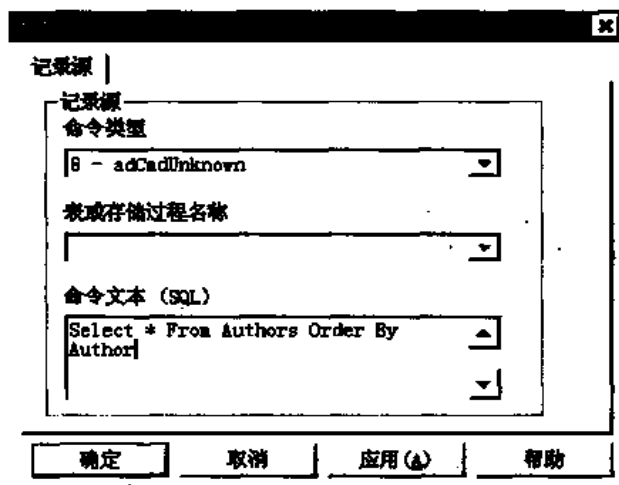


图 6.11 设置 RecordSource 属性

对 Adodc2来说,我们改变如下几个属性的值:

(1) Name 属性

改变 Name 属性为 Ado_Article。

(2) ConnectionString 属性

类似 Adodc1来设置 ConnectionString 属性,用户最后可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

(3) RecordSource 属性

类似于 Adodc1,RecordSource 属性用来设置 RecordSet 源。在这里,我们将在属性页的命令文本下键入如下的 SQL 语句:

```
Select * From "Title Author"  
Order By Au_ID
```

这表明在程序执行的刚开始,通过 Adodc2把 Biblio 数据库 Title Author 表中的所有数据都检索出来,并按照 Au_ID 表项来排序。

对 DataList1来说,我们改变如下几个属性的值:

(1) ListField 属性

ListField 属性用来设置 RecordSet 对象中字段的名称,它用来填充 DataList1控件的列表部分。在这里,我们设置 ListField 属性为 Author,它表明了 DataList1控件的列表部分所填充的是 Title Author 表中的 Author 字段的具体数据。

(2) RowSource 属性

RowSource 属性决定由哪一个数据源为 ListField 属性供应数据。在本例中,我们设置 RowSource 属性为 Ado_AuID,它表明为 ListField 属性提供数据的数据源是名为 Ado_AuID 的 ADO Data 控件,即 Adodc1控件。

(3) BoundColumn 属性

BoundColumn 属性设置一个 RecordSet 对象中的源字段名称,它用于为另一个控件提供数据值。在本例中,BoundColumn 属性被设置为 Authors 表中的第一个字段。即 Au_ID 字段。当单击 DataList1 控件时,BoundText 属性返回与在 DataList1 控件中所显示的作者相关联的 Au_ID 字段的值。这个值将用于对 Title Author 表的查询,该查询为 DataGrid1控件提供数据。

对 DataGrid1来说,我们改变如下几个属性的值:

(1) DataSource 属性

DataSource 属性为该控件指定数据源。在本例中,该属性被设置为名为 Ado_Article 的 ADO Data 控件,这将返回 Title Author 表中的所有记录。

(2) 改变 DataGrid 的属性

在 DataGrid 控件中,单击鼠标右键,在弹出菜单中选择“属性...”菜单项。在属性页中选择“列”选项卡。然后,在“列”右边的下拉框中选择 Column_0(Au_ID);在“标题”的后面键入作者 ID;在“数据字段”右边的下拉框中选择 Au_ID。完成之后如图 6.12 所示。之后,类似地在“列”右边的下拉框中选择 Column_1(ISBN);在“标题”的后面键入书编号;在“数据字段”右边的下拉框中选择 ISBN。

在属性的改变和设置完成后,我们再来看一下应用程序的界面,如图 6.13 所示。

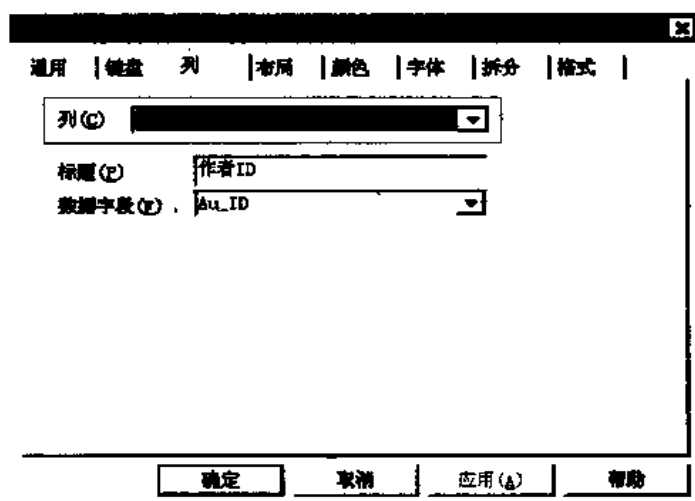


图6.12 DataGrid 控件的属性页

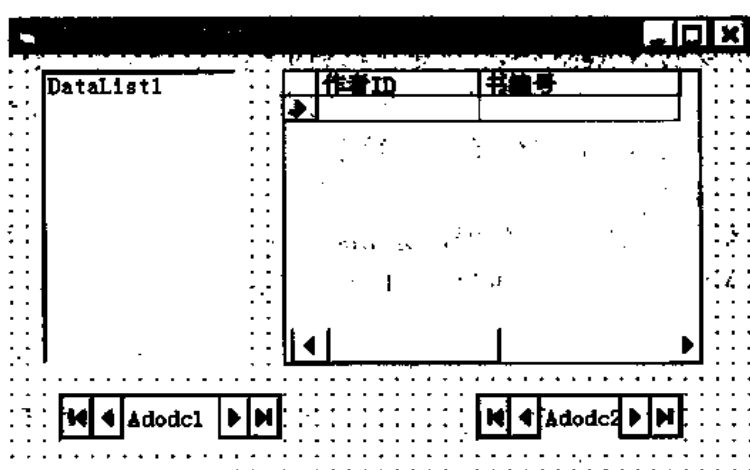


图6.13 属性设置后的应用程序界面

接下来,我们就要给应用程序加入一小段代码。

```
Private Sub DataList1_Click()
    Dim strQuery As String
    strQuery = "Select * FROM " & "" & "Title Author" & "" & " WHERE Au_ID = " & _
        DataList1.BoundText
    With Ado_Article
        .RecordSource = strQuery
        .Refresh
    End With
End Sub
```

本段代码是在用鼠标选择 DataList1 中的一个具体作者时触发。首先,我们定义了一个 String 类型的变量 strQuery。该变量存储了一个 SQL 语句,这在接下来的程序语句 strQuery

= "Select * FROM " & ""Title Author"" & " WHERE Au_ID = " & DataList1.BoundText
中我们就可以看到。最后的一个 With 语句是把 SQL 语句赋给名为 Ado_Article 的 ADO Data 控件的 RecordSource 属性并且刷新。这样,每当用户单击 DataList1 控件中的某一作者时, DataGrid 控件中就会显示出相应作者所编写过的书籍的编号。

最后,我们在窗体上添加两个 CommandButton 控件。调整窗体上各控件的位置和大小并设置两个 CommandButton 控件的 Caption 属性为刷新和退出。在程序的运行过程中单击“刷新”按钮应用程序就会重新从数据库中检索数据。为了在程序运行过程中使两个 ADO Data 控件为不可见,我们把 Adodc1 和 Adodc2 的 Visible 属性均设置为 false。完成以上操作后如图 6.14 所示。

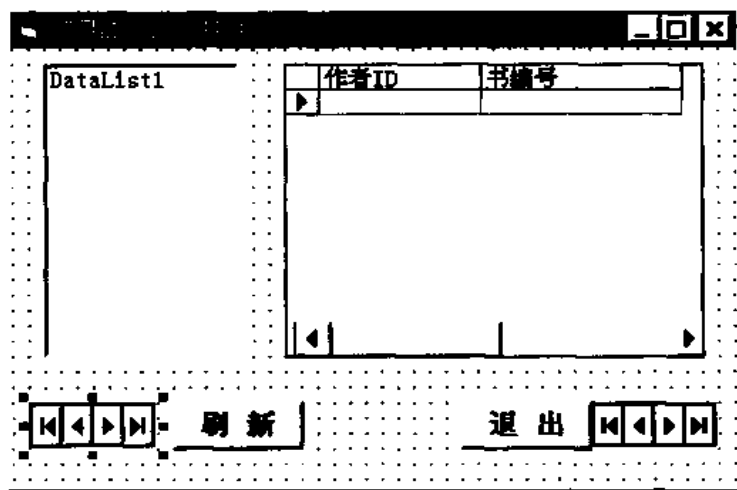


图 6.14 加入 CommandButton 后的界面

我们为刷新按钮加入如下的代码:

```
Private Sub Command1_Click()  
    Dim strQuery As String  
    strQuery = "Select * From" & ""Title Author"" & _  
        "Order By Au_ID"  
    Screen.MousePointer = vbHourglass  
    With Ado_Article  
        .RecordSource = strQuery  
        .Refresh  
    End With  
    Screen.MousePointer = vbArrow  
End Sub
```

这里需要解释的是 Screen.MousePointer = vbHourglass。这条语句使用 Screen 对象的 MousePointer 属性来设置鼠标的形状,其中 vbHourglass 为 Visual Basic 的内部常量,表示鼠标的形状为沙漏状。当刷新完成以后,再利用语句 Screen.MousePointer = vbArrow 把鼠标的形状重新设置为箭头状。其中,vbArrow 也是 Visual Basic 的内部常量。在这里,我们顺便介绍一下代表鼠标形状的 Visual Basic 内部常量,见表 6.1。它们在实际程序的设计过程中是十

分有用的。

我们为退出按钮加入如下的代码：

```
Private Sub Command2_Click()  
    End  
End Sub
```

当用户单击“退出”按钮时，应用程序就会执行 Command2_Click() 事件中的 End 语句从而退出本应用程序。

表6.1 鼠标形状的内部变量

内部常量	常量值	常量描述
vbDefault	0	这是一个缺省值，形状由对象决定
vbArrow	1	箭头
vbCrosshair	2	十字线
vbIbeam	3	I 型
vbIconPointer	4	图标(矩形内的小矩形)
vbSizePointer	5	尺寸线(指向东、南、西和北四方向的箭头)
vbSizeNESW	6	右上-左下尺寸线(指向东北和西南方向的双箭头)
vbSizeNS	7	垂直尺寸线(指向南和北的双箭头)
vbSizeNWSE	8	左上-右下尺寸线(指向东南和西北方向的双箭头)
vbSizeWE	9	水平尺寸线(指向东和西两个方向的双箭头)
vbUpArrow	10	向上的箭头
vbHourglass	11	沙漏(表示等待状态)
vbNoDrop	12	不允许放下
vbArrowHourglass	13	箭头和沙漏
vbArrowQuestion	14	箭头和问号
vbSizeAll	15	四向尺寸线
VbCustom	99	通过 MouseIcon 属性所指定的自定义图标

6.3 Remote Data 控件

RemoteData 控件通过使用 Microsoft 的远程数据对象来实现数据访问。该技术提供了对标准数据库格式的无缝访问，用户不必编写任何代码就可以创建数据识别的应用程序。RemoteData 控件适合于较大的客户机/服务器数据库，包括开放式数据库和连接性(ODBC)数据库，诸如 Microsoft SQL Server 以及 Oracle 等大型数据库。

RemoteData 控件和 Data 控件二者都包括在 Visual Basic 中，以提供向后兼容性。不过，由于 ActiveX 数据对象(ADO)的适应性更广，建议使用 ADO Data 控件来创建新的数据库应用程序。关于 ADO Data 控件的具体使用，我们在上一节中已经给出了一个详细的实例。在这里，读者需要注意的是，RemoteData 控件并不支持 Image 数据类型。

由于 RemoteData 控件同前面两种控件的相似性，在这里我们就不再对 RemoteData 控件

给出一个具体的实例了。RemoteData 控件主要是用来进行远程数据的访问。使用 Remote Data 控件,ADO 可让应用程序直接访问并修改数据源(有时称为双层系统)。例如,如果要连接到包含所需数据的数据源,则该连接即是在双层系统中的直接连接。然而,也需要通过像 Microsoft Internet Information Server (IIS) 这样的媒介间接地访问数据源。这种方法称为三层系统。IIS 采用客户端/服务器系统,该系统可帮助本地或客户端的应用程序通过 Internet 或 Intranet 高效地调用远程或服务器程序。服务器程序访问数据源,并有选择地处理已获得的数据。例如,用户的 Intranet Web 页包含有用 Microsoft(r) Visual Basic(r) Scripting Edition (VBScript) 编写的应用程序,该程序连接 IIS。IIS 相应地连接实际数据源,检索数据,以某种方式处理数据,然后将已处理的信息返回给应用程序。在上面的这个例子中,应用程序从未直接连接数据源,该工作由 IIS 完成。而 IIS 则是利用 ADO 来访问数据的。

RemoteData 控件在远程数据对象(RDO)和数据识别的被绑定的控件之间提供了接口。通过 RemoteData 控件,能够做到以下几点。

- (1) 建立起与基于其本身属性的数据源的连接。
- (2) 创建 rdoResultset。

当使用远程数据对象时,几乎完全使用 rdoResultset 对象来同数据进行交互。rdoResultset 对象由 RemoteData 控件创建,或者由 rdoQuery、rdoTable 或 rdoConnection 对象的 OpenResultset 方法来创建。关于 rdoResultset 的知识其实已经牵涉到 RDO 对象的编程。关于 RDO 对象的编程本书不想做具体的展开,有兴趣的读者可以参考相关的书籍。

- (3) 把当前行的数据传送给相应被绑定的控件。
- (4) 允许对当前行指针进行定位。
- (5) 将对被绑定的控件所做的任何更改反传给数据源。

如果没有 RemoteData 控件,则 Data 控件或其等价物,即窗体上的数据识别(被绑定的)控件就不能自动存取数据。RemoteData 控件和 Data 控件是 DataSource 控件的例子。利用 DataSource 控件在根本不用写任何代码的情况下就能执行大部分远程数据存取操作。被绑定在 DataSource 控件上的数据识别控件,能自动显示来自于当前行某一列或数列的数据,或者在某些情况下,能显示当前行在任一行、一组的某一列或数列的数据。DataSource 控件执行当前行的所有操作。

RemoteData 控件能自动地处理一些意外事件,包括空结果集、增加新行、编辑和更新现有行、转换和显示复杂的数据类型、以及处理某些类型的错误等等。然而,在更完善的应用程序中,必须要考虑到捕获 RemoteData 控件不能处理的错误情况。例如,如果远程服务器在访问数据源时出现错误、权限不够、或者查询不能按照编码正常执行等等,都会产生可捕获的错误。如果错误出现在应用程序过程开始之前,或者作为某些内部错误的结果,则将触发 Error 事件。

利用 RDO 和 RemoteData 控件,应用程序不需使用本地的查询处理程序即可访问 ODBC 数据源。这意味着,在访问远程数据库引擎时,可以获得更好的性能与更大的灵活性。通过使用 RDO,可以实现以下的功能:

- (1) 创建简单的无游标结果集,或更复杂的游标。
- (2) 执行查询并处理任意数量的结果集。
- (3) 执行返回结果集的存储过程,无论存储过程是否带有输出参数和返回值。

(4) 执行包括数据操作或数据定义运算在内的动作查询。

(5) 限制返回或处理的数据行数。

(6) 在不妨碍执行查询的情况下,监视远程数据资源产生的所有信息和错误。

(7) 支持同步、异步或事件驱动的异步处理,因此,即使在执行冗长的查询或者重定位当前行指针时,应用程序也不会被阻塞。

接下来我们就给出 RemoteData 控件与数据相关的一些属性。这些与数据相关的属性可以在设计时设置。

(1) Connect 属性

Connect 属性是一个字符串,该字符串可以包含进行一个连接所需的所有设置值。在该字符串中所传递的参数是与驱动程序相关的。例如,ODBC 驱动程序允许该字符串包含驱动程序、数据库、用户名称以及密码。

(2) UserName 属性

用 UserName 属性来对一个受保护的数据库标识一个用户。该用户必须使用一个合法的、该数据库管理系统能识别的密码。用户名称也可以包含在 Connect 属性值中,此时本属性中的设置就是多余的。

(3) Password 属性

Password 属性与用户名称一起使用,密码使用户可以访问受保护的数据。密码也可以包含在 Connect 属性值中,此时本属性中的设置就是多余的。

(4) SQL 属性

SQL 属性包含了用于检索一个结果集的 SQL 语句。结果集的大小可以决定是使用客户端游标还是使用服务器端的游标。例如,一个小的结果集可以用一个客户端的游标来管理,而如果结果集较大则需要使用服务器端的游标。

(5) RowsetSize 属性

RowsetSize 属性设置在结果集返回的行数,如果光标为键集光标。可以根据计算机的内存资源调整这个数目,以获得性能的改善。

(6) ReadOnly 属性

ReadOnly 属性用来指定数据是否可以写入。如果正在编写的数据是不需要的,将这个属性设为“真”可以获得性能的改善。

(7) CursorDriver 属性

CursorDriver 属性决定驱动程序的位置和类型。这个属性的设置值将影响其它属性的设置。例如,只要结果集比较小,则选择 ODBC 客户端游标可以提高性能。

(8) LockType 属性

LockType 属性决定当其他人试图更改数据时如何锁定该数据。如果不希望其他人更改该数据(当用户正在查阅时),可以将 LockType 设置为“optimistic”——即其他人可以自由地查看和更改该数据。如果将该属性设置为“pessimistic”,那么在用户访问该数据期间,其他人不能访问该数据。

(9) BOFAction、EOFAction 属性

这两个属性决定当该控件位于光标的开始和末尾时的行为。提供的选择包括停留在开始或末尾、移动到第一个或最后一个记录、或添加一个新记录。

(10) ResultSetType 属性

ResultSetType 属性用来决定光标是静态类型还是键集类型。

(11) KeySetSize 属性

如果光标为键集类型,则可以使用 KeySetSize 属性来优化返回的结果集的大小。

(12) LoginTimeout 属性

LoginTimeout 属性设置等待的秒数,超时则返回一个错误。

(13) MaxRows 属性

MaxRows 属性用来指定游标的大小。如何确定这个属性取决于所检索的记录大小,以及用户的计算机上可用内存资源的多少。大的记录(具有很多列以及字符串)与小的记录相比会消耗更多的资源。MaxRows 属性就应该相应地减少。

(14) Options 属性

Options 属性用来指定该控件是否异步地执行查询。当估计一个查询可能要花费好几分钟来执行时,应使用异步操作。

(15) Prompt 属性

在 RDO 基于该 RemoteData 控件的参数打开一个连接时,Connect 属性应当包含足够的用于建立该连接的信息。如果没有提供像数据源名称、用户名称或密码等这样的信息,则 ODBC 驱动程序管理器将显示一个或多个对话框,以便从用户取得这些信息。如果不想让这些对话框出现,可以相应地设置这个 Prompt 属性来取消这一功能。

(16) QueryTimeout 属性

QueryTimeout 属性设置等待一个查询完成的秒数,超时则返回一个错误。

(17) BatchSize 属性

这个属性决定在一个批处理中可以发送多少条语句——如果驱动程序允许使用批处理语句。

在本章要结束之前,我们来对 Data、ADO Data、RemoteData 控件做一个总结。RemoteData 控件在很多的方面类似于 Visual Basic 的 Data 控件和 ADO Data 控件。二者都用于将数据识别的被绑定控件连接到数据源。它们的主要区别是: RemoteData 控件使用 RDO 连接到 ODBC 驱动程序管理器,而 Data 控件和 ADO Data 控件则使用 DAO 或 ADO 连接到 Jet 数据库引擎,后者可以连接到 Jet 数据库,也可以连接到 IISAM 数据源。Data 控件和 ADO Data 控件也可以通过 ODBC Direct 接口间接与 RDO 连接,尽管这不是很常用。尽管这三种控件都使用相同的 ODBC 驱动程序管理器和数据源项,但是,RDO、ODBC Direct、DAO 不能共享数据源连接、数据对象以及其它一些资源。

第七章 使用 DataCombo 和 DataList

DataCombo 控件是一个数据绑定组合框,它自动地由一个附加数据源中的一个字段填充;并且可选择地更新另一个数据源的一个相关表中的一个字段。而 DataList 控件是一个数据绑定列表框,它也是自动地由一个附加数据源中的一个字段填充,并且可选择地更新另一个数据源中一个相关表的一个字段。DataCombo 控件与 DBCombo 控件代码兼容,同样地,DataList 控件与 DBList 控件代码兼容。但是,DataCombo 控件和 DataList 控件被优化来同 ActiveX Data Objects (ADO)一起工作。DataCombo 控件和 DataList 控件均可以在文件 Msdatlst.ocx 中找到。要在应用程序中使用这一控件,必须把 OCX 文件添加到工程中。当发行用户自己的应用程序时,要把文件 Msdatlst.ocx 安装到用户的 Microsoft Windows System 或 System32 目录下。

在本章中我们主要讲述以下两个部分:

- 使用 DataCombo 控件的实例
- 使用 DataList 控件的实例

7.1 使用 DataCombo 控件的实例

DataCombo 控件是以一个下拉框的形式提供给用户的。它可以和一个具体数据库中的表、表中的某些项、一段 SQL 语句相联系,从而在下拉框中显示出具体的数据。下面的一个例子就是使用 DataCombo 控件来操纵数据库中的数据,当然在程序中我们也会用到 ADO Data 控件和 DataGrid 控件。应用程序完成之后的运行状态如图 7.1 所示。

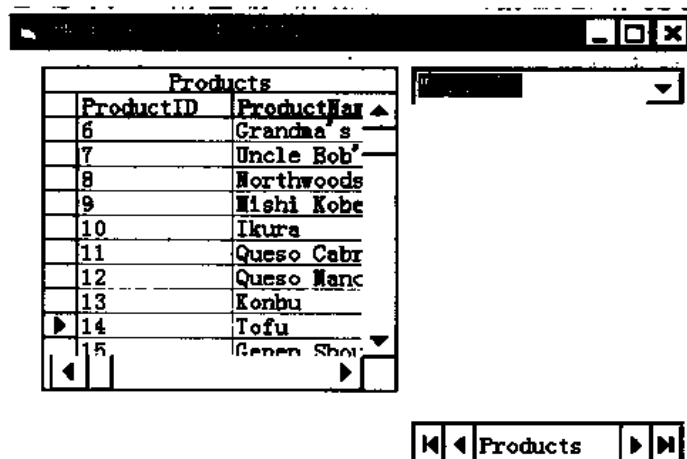


图 7.1 程序完成后的运行界面

首先,在 Visual Basic 的开发环境下新建一个标准的 Exe 工程,把 Form1 的 Caption 属性

改为使用 DataCombo 控件的实例。Form1 的 Caption 属性缺省值为 Form1。完成以后的窗体如图 7.2 所示。

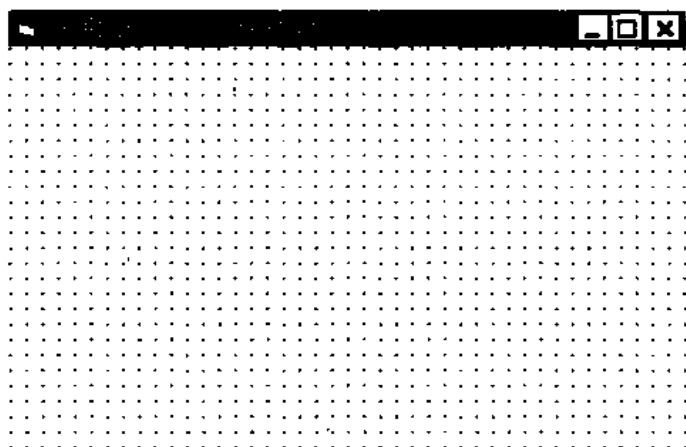


图 7.2 改变 Form1 的 Caption 属性

往 Form1 上加入两个 ADO Data 控件、一个 DataCombo 控件和一个 DataGrid 控件。DataCombo 控件和 DataGrid 控件起到显示具体数据的作用，而两个 ADO Data 控件只是起到在数据库和具体的数据显示中的桥梁作用。调整这四个控件的大小和具体位置，完成之后如图 7.3 所示。

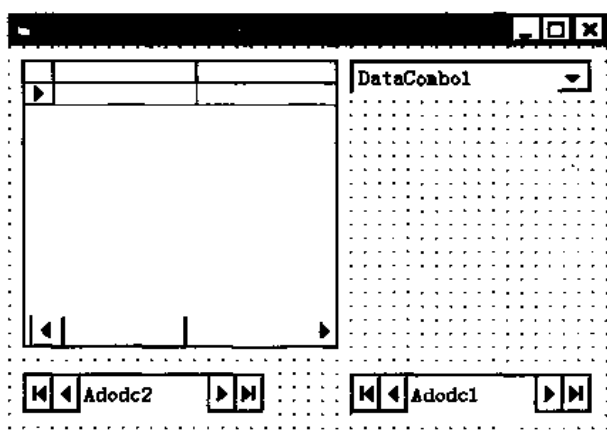


图 7.3 加入控件之后的界面设置

接下来，我们就要改变这四个控件的一些具体的属性值。

对 Adodc1 来说，我们改变如下几个属性的值：

(1) Name 属性

改变 Name 属性为 AdoDataSource。

(2) ConnectionString 属性

ConnectionString 属性表示 Adodc1 的连接字符串，可以通过三种方法来设置连接字符串属性。设置 ConnectionString 如图 6.10 所示，在这里我们选择“使用 ODBC 数据资源名称”，并在下拉框中选择 NorthWind 数据源。如果用户在下拉框中没有见到名为 NorthWind 的数据

源名,请用户先在控制面板的 ODBC 数据源中配置 NorthWind 数据源。单击“确定”按钮,用户就可以在 ConnectionString 属性中看到“DSN = NorthWind”的字符串形式。

(3) RecordSource 属性

RecordSource 属性设置 RecordSet 源。在这里,我们将在属性页的命令文本下键入如下的 SQL 语句:

```
Select *  
From Products
```

属性页如图 6.11 所示,可在其中设置 RecordSource 属性。在命令文本下键入相应的 SQL 语句,单击“确定”按钮,用户就可以在 RecordSource 属性中看到“Select * From Products”的字符串形式。

(4) Caption 属性

Caption 属性表示在 Adodc1 中显示给用户的 Caption 性质。在此,我们把 Caption 属性改为 Products。Adodc1 的 Caption 属性的默认值为 Adodc1。

对 Adodc2 来说,我们改变如下几个属性的值:

(1) Name 属性

改变 Name 属性为 AdoRowSource。Adodc2 的 Name 属性的默认值为 Adodc2。

(2) ConnectionString 属性

类似 Adodc1 来设置 ConnectionString 属性,用户最后可以在 ConnectionString 属性中看到“DSN = NorthWind”的字符串形式。

(3) RecordSource 属性

类似于 Adodc1,RecordSource 属性用来设置 RecordSet 源。在这里,我们将在属性页的命令文本下键入如下的 SQL 语句:

```
Select CompanyName, SupplierID  
From Suppliers;
```

(4) Caption 属性

Adodc2 的 Caption 默认的缺省值为 Adodc2,在这里我们把它改为 Suppliers。

(5) Visible 属性

Visible 属性是用来表示控件在程序的执行过程中是否可见的性质。Adodc2 的默认值为 true,它表示 Adodc2 在应用程序的运行过程中是可见的。在这里,我们要把它改为不可见,即 false。它使得 Adodc2 在程序的执行过程中不可见,当然,它在程序的设计过程中是可见的。

对 DataGrid1 来说,我们改变如下几个属性的值:

(1) Name 属性

改变 Name 属性为 Products。DataGrid1 的 Name 属性的默认值为 DataGrid1。

(2) DataSource 属性

设置 DataSource 属性为 AdoDataSource,这样就把 DataGrid1 控件同 Adodc1 控件联系起来了。

(3) Caption 属性

设置 DataGrid1 的 Caption 属性为 Products。这样在设计和运行时,用户就可以在 DataGrid1 控件的顶端看到 Products 的字样。

对 DataCombo1 来说,我们改变如下几个属性的值:

(1) Name 属性

改变 Name 属性为 Suppliers。DataCombo1 的 Name 属性的默认值为 DataCombo1。

(2) DataSource 属性

DataSource 属性返回或设置一个数据源,通过该数据源,数据使用者被绑定到一个数据库中去。在这里,DataSource 属性被设置为 AdoDataSource。它表示了 DataCombo1 控件是通过名为 AdoDataSource 的 Adodc1 控件绑定到具体的数据库中去的。

(3) DataField 属性

DataField 属性返回或设置数据使用者将被绑定到的字段名。被绑定的控件提供了对数据库中特定数据的访问功能。管理单个字段的被绑定的控件一般显示当前记录中某特定字段的值。被绑定的控件的 DataSource 属性指定一个合法的数据源,DataField 属性则指定一个在数据源所创建的 Recordset 对象中的合法的字段名称。这些属性一起说明哪些数据出现在该被绑定的控件中。在这里,DataSource 属性被设置为 SupplierID。

(4) RowSource 属性

RowSource 属性设置一个指定 ADO Data 控件的值,DataCombo 控件的列表由这个 ADO Data 控件填充。在这里,我们把 RowSource 属性设置为 AdoRowSource。这说明 DataCombo 控件的列表是由名为 AdoRowSource 的 Adodc2 控件来填充的。

(5) ListField 属性

ListField 属性用来设置 RecordSet 对象中字段的名称,它用来填充 DataCombo1 控件的列表部分。在这里,我们设置 ListField 属性为 CompanyName,它表明了 DataCombo1 控件的列表部分所填充的是 Suppliers 表中的 CompanyName 字段的具体数据。

(6) BoundColumn 属性

BoundColumn 属性返回或设置一个 Recordset 对象的源字段的名称,该 Recordset 对象用来为另一个 Recordset 提供数据值。在这里,我们设置 BoundColumn 属性为 SupplierID。

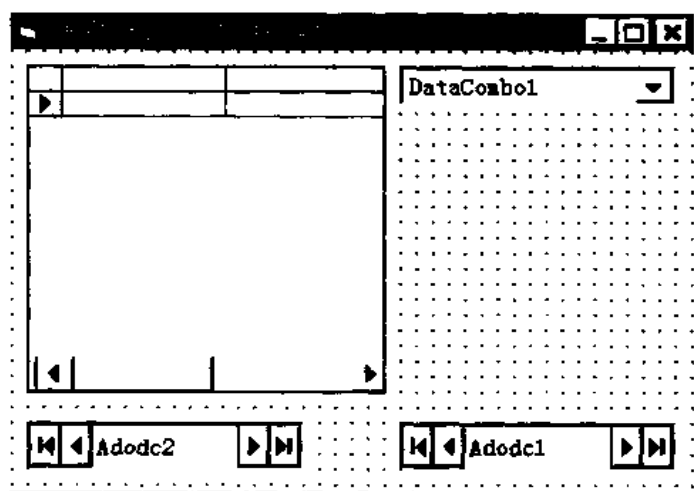


图7.4 完成设计后的界面

这样,我们就完成了 DataCombo 控件实例的设计过程。完成后的界面如图7.4所示。在本

程序中,我们并没有加一行代码。用户可以通过单击可视的 ADO Data 控件上的箭头来浏览记录集。如果这样做,DataCombo1控件将更新和显示每一个产品的供应商的名称。如果要编辑 SupplierID 字段,则单击 DataCombo1控件的箭头来显示一个下拉列表,然后再单击一个不同的供应商,来改变写入到 SupplierID 字段的值。

7.2 使用 DataList 控件的实例

DataList 控件和 DataCombo 控件具有很大程度上的相似性。这些相似性使我们有时在程序设计过程中可以不必过于注重使用 DataList 控件或 DataCombo 控件。DataList 控件和 DataCombo 控件在形式上最大的区别是在设计中是否有下拉框。基于这一点上的考虑,我们有时可以按需要的不同来选择 DataList 或 DataCombo 控件。

DataList 控件和 DataCombo 控件与众不同的特性是具有访问两个不同的表,并且将第一个表的数据链接到第二个表的某个字段的能力。这是通过使用两个数据源来实现的。在一个关系数据库中,对于重复使用的信息并不是在多个地方都保存其全部的信息。大多数这种信息都保存在由多个字段组成的一个记录集中;在这些字段中有一个“标识符”字段来唯一地标识这个记录集。从减少数据冗余的角度上来看,这样做无疑是有利的。

例如,Visual Basic 提供的 Biblio 数据库在一个名为“Publishers”的表中存储了若干个出版公司的名称。这个表包括很多字段,诸如地址、城市、邮政编码及电话号码等。但是为了简单起见,只考虑这个表的两个本质字段 Name 和 PubID 字段。Name 字段存储一个出版商的名称,而 PubID 字段则存储一个相对“不友好的”值,如一个数或代码。但这个不友好的值是很重要的,因为这个值唯一地标识该出版商,并且可以作为一种链接整个记录集的手段。此外,这个值会存储在第二个表中的多个记录集中。第二个表的名称为“Titles”,其每个记录集包含的信息包括标题、出版年份、国际标准书号 ISBN 等。在这些字段中有一个字段的名称就是“PubID”。这个字段的名称与 Publishers 表中的相应字段的名称相同,因为这个字段存储了将该标题和一个特定的出版商链接在一起的值。

这种可行方案提出了一个小问题:给定一个允许用户插入新标题的数据库应用程序,用户必须用某种方法输入标识出版商的整数。如果用户能记住每个出版商的唯一标识符,那么也还是可行的,不过如果一方面用户能看到出版商的名称,另一方面存入应用程序的又是数据库中相应的值,则会显得更加方便。而 DataList 和 DataCombo 控件就可以轻松地解决这个问题。

在以下的介绍中,笔者不给出具体的操作过程。相信读者完全可以按照上一节给出的例子来自行地完成本实例。本应用程序使用了两个数据源,共用到了三个字段,并且无任何的编码。

DataList 和 DataCombo 控件使用两个数据源来实现这个应用程序。在只显示出版商的名称(来自 Publishers 表)的同时,DataList 或 DataCombo 控件只将 PubID 字段的值写入到 Titles 表。通过“属性”窗口,将 RowSource 设置为提供要写入的数据的数据源(即 Publishers 表)。然后将 DataSource 属性设置为要写入数据的数据源(即 Titles 表)。最后,设置 DataField、ListField、BoundColumn 属性。图7.5则演示了如何将两个数据源(以两个 Data 控件的形式)以及三个字段指定给一个 DataCombo 控件。关于 DataSource、DataField、RowSource、BoundColumn、ListField 等属性的具体所代表的含义读者可以参看上一小节中使用 DataCombo 控件的实例。

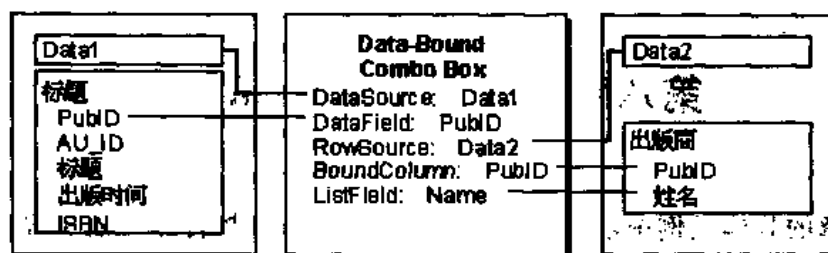


图7.5 数据源之间的关系

简要而言, ListField 属性决定该控件所显示的是哪一个字段。在本例中就是出版商的名称。另一方面, BoundColumn 属性则决定 Publishers 表中由哪一个字段向 Title 表供应实际所需的值。注意 Publishers 表中的 PubID 字段不能也不应该被编辑。相反, 在 PubID 字段中的值将写入到由 DataField 属性所指定的字段。在本例中, 这个属性就是 Titles 表中的 PubID 字段。

需要注意的是: DataList 和 DataCombo 控件也完全可以与单个数据控件一起使用。要实现这一点, 可以将 DataSource 和 RowSource 属性设置为同一个数据控件, 并且将 DataField 和 BoundColumn 属性设置为该数据控件的记录集中的同一个字段。在这种情形下, 将使用 ListField 的值来填充该列表, 且这些值来自于被更新的同一个记录集。如果指定了一个 ListField 属性, 但没有设置 BoundColumn 属性, 则 BoundColumn 将自动被设置为 ListField 字段。

由此, 我们完成了本章的学习。希望读者在学习之后, 能够多想一想、试一试, 注意不同控件之间的比较。

第八章 使用 DataGrid 控件

在前面几章的具体实例中,我们已经使用过 DataGrid 控件来控制 and 显示数据库中的数据。DataGrid 控件是一种类似于电子数据表格的绑定控件,使用它可以用来显示一系列行和列来表示 Recordset 对象的记录和字段。可以使用 DataGrid 来创建一个允许最终用户阅读和写入到绝大多数数据库的应用程序。DataGrid 控件可以在设计时快速进行配置,只需少量代码或无需代码。当在设计时设置了 DataGrid 控件的 DataSource 属性后,就会用数据源的记录集来自动填充该控件,以及自动设置该控件的列标头。然后您就可以编辑该网格的列;删除、重新安排、添加列标头或者调整任意一列的宽度。

在本章中我们主要讲述以下几个部分:

- 更改数据源的实例
- 关于列的实例
- DataGrid 的视图
- 使用书签的实例
- 与类模块一起使用 DataGrid 控件

8.1 更改数据源的实例

8.1.1 更改数据源的 RecordSource

在 Visual Basic 中更改所显示的数据的最通常的方法是改变与显示数据的控件相连控件的 DataSource 属性。例如,如果 DataGrid 控件使用一个 ADO Data 控件作为其 DataSource,则重写 RecordSource、刷新该 ADO Data 控件,都将改变所显示的数据。在接下来的例子中,我们将利用第六章中使用 Data 控件的实例作为基础来给读者讲述如何更改用户应用程序数据源的 RecordSource。

打开使用 Data 控件的工程,在窗体上做如下的修改:

(1) 删除窗体上的 Data 控件。

(2) 添加一个 ADO Data 控件到窗体,并设置 ADO Data 控件的如下属性:

- 在 ConnectionString 属性中选择使用 ODBC 数据资源名称项下的 Biblio 数据库,表示所连接的数据源为 Biblio。
- 在 RecordSource 属性的命令文本项下键入 Select * From Authors 的 SQL 语句。

(3) 添加一个 CommandButton 控件到窗体,并设置 CommandButton 控件的如下属性:

- 改变 Caption 属性为 Fresh。
- 设置 Font 属性为宋体,字体样式为 Bold,大小为小五号。

(4) 删除三个 TextBox 控件和三个 Label 控件,添加一个 DataGrid 控件。并把 DataGrid 控件的 DataSource 属性设置为 Adodc1。

(5) 最后,改变窗体的 Caption 属性为更改数据源的 RecordSource。调整各控件的大小及位置,完成以后的界面如图8.1所示。

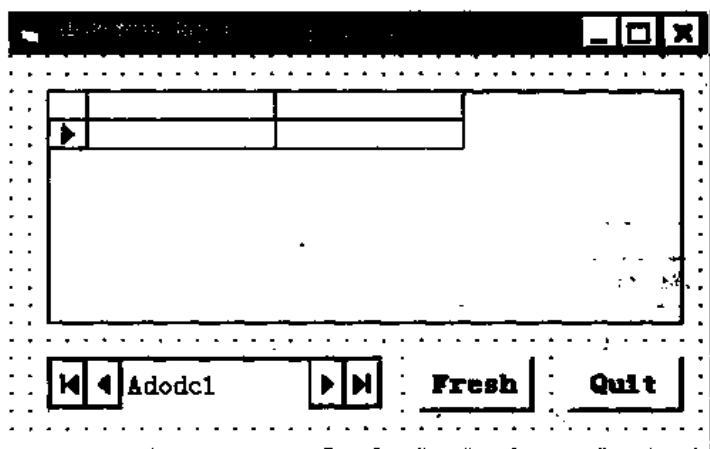


图8.1 更改数据源的 RecordSource

运行该应用,用户就可以利用 Adodc1的左右箭头按钮来操纵 DataGrid 控件中数据的显示。下面,我们要在 Fresh 按钮的 Click 事件中加入一段代码,使得在应用程序的运行过程中单击 Fresh 按钮就会在 DataGrid 控件中选出 Au_ID 为6的数据行。程序代码如下:

```
Private Sub Command2_Click()  
    Dim strQuery As String  
    strQuery = "SELECT * FROM Authors WHERE Au_ID = 6"  
    Adodc1.RecordSource = strQuery  
    Adodc1.Refresh  
End Sub
```

接下来,我们要在窗体上再添加一个名为 Restore 的 CommandButton 控件,使得用户单击 Restore 按钮后程序会恢复刚运行时的状态。在 Restore 控件的 Click 事件中加入如下的代码:

```
Private Sub Command3_Click()  
    Dim strQuery As String  
    strQuery = "SELECT * FROM Authors"  
    Adodc1.RecordSource = strQuery  
    Screen.MousePointer = vbHourglass  
    Adodc1.Refresh  
    Screen.MousePointer = vbArrow  
End Sub
```

程序全部完成之后的设计界面如图8.2所示。

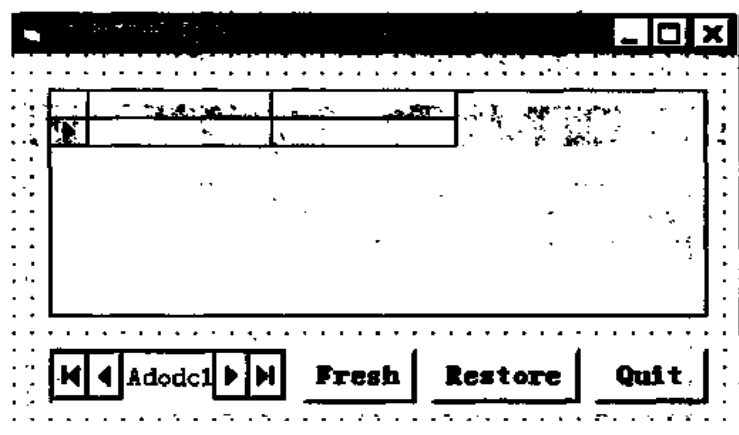


图8.2 加入 Restore 按钮后的界面

8.1.2 更改数据源

在应用程序运行时,可以将 DataSource 属性重新设置为一个不同的数据源。例如,用户可能具有若干个 ADO Data 控件,每个控件连接着不同的数据库,或者设置为不同的 RecordSource 属性。可以简单地将 DataSource 从一个 ADO Data 控件重新设置为另一个 ADO Data 控件。在接下来的程序实例中,我们将使用多个的 ADO Data 控件来更改数据源。

首先,我们在 Form1窗体上放置一个 DataGrid 控件、一个 ComboBox 控件、一个 CommandButton 控件和三个 ADO Data 控件,如图8.3所示。

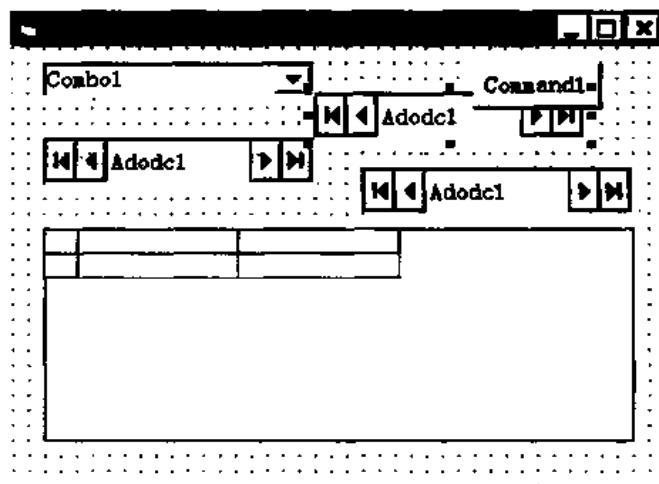


图8.3 在窗体上放置控件

细心的读者或许已经发现,在 Form1上放置了三个名为 Adodc1 的 ADO Data 控件。在这里,我们使用的是控件数组的方法。读者可以把控件数组理解为一个类型为某一种控件的数组,数组中的每个元素为类型相同的一个控件。

要创建一个控件数组的方法很简单,读者只需要在放置了第一控件之后用简单的 Copy 和 Paste 命令就能够实现。如果读者这样做了,便会看到一个对话框。在设计 Visual Basic 应用

程序期间,任何时候多次使用相同的控件名(Name 特征的值),Visual Basic 都会问是否真的想创建控件数组。单击 Yes 按钮(或按 Enter),很快便可以看到,在应用程序运行期间,将能够增加相同类型的更多的控件。在控件数组中,每个新控件都称作控件数组的元素。

现在,由于多个控件具有相同的名称,Visual Basic 需要一种方法来区别它们。这要用一种新特性,即 Index 特性。在 Visual Basic 创建一个控件数组时,第一个控件为0的 Index 特性,第二个控件为1的 Index 特性,依此类推。与 Visual Basic 对象的任何特性一样,可以在开始时使用 Properties 窗口来改变 Index 特性。事实上,如果在设计时赋值任何数字到控件的 Index 特性,Visual Basic 自动创建一个控件数组。这样可以在设计时不必使用多个控件,而创建一个控件数组。在理论上,一个控件数组中至多有255种元素,但是,这样的情况是非常少见的,因为这会浪费过多的 Windows 资源。

如果要在控件数组中使用一个事件过程,那么在此事件过程中要使用一个新的参数 Index As Integer,如图8.4所示。Index 参数是控件数组平稳作用的关键。Visual Basic 调用此事件过程时,会将 Index 参数传送给过程。使用这种方式,事件过程可以使用索引来确定要做什么。

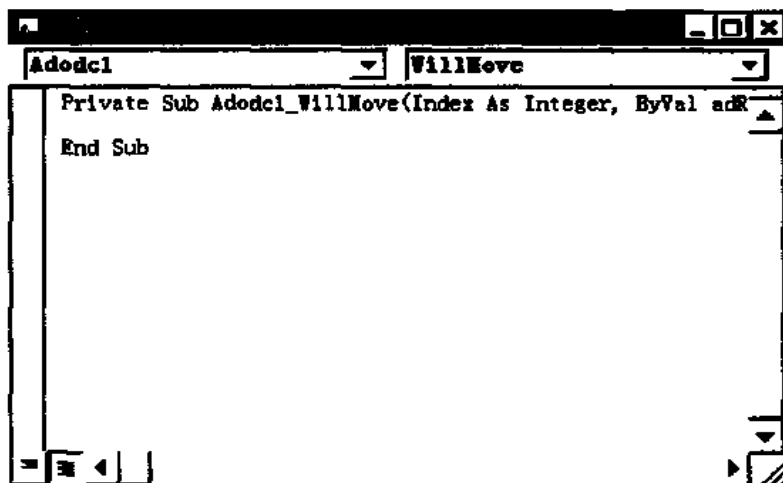


图8.4 具有 Index 参数的事件过程模板

接下来,我们改变和设置如下几个控件的属性。

对 Combo1控件来说,我们改变如下几个属性的值:

(1) Text 属性

Combo1控件的默认属性为 Combo1,在这里我们把它改成 Authors。

(2) List 属性

List 属性用来设置控件的列表部分的项目。列表是一个字符串数组,数组的每一项都是一列表项目。打开 List 属性的下拉框,在下拉框中分行地输入 Authors、Publishers 和 Titles。这样在程序运行时,用户可以在 Combo1的下拉框中选取 Authors、Publishers 或 Titles。

(3) Item Data 属性

Item Data 属性用来返回或设置 ComboBox 控件中每个项目具体的编号。在这里,我们打开 Item Data 属性的下拉框,在下拉框中分行地输入0、1和2。

对 Adodc1(0)控件来说,我们改变如下几个属性的值:

(1) Captions 属性

Adodc1(0)控件的默认属性为 Adodc1,在这里我们把它改成 Authors。

(2) ConnectionString 属性

用户在 ConnectionString 属性页的“使用 ODBC 数据资源名称”项中选择 Biblio 数据库。用户最后可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

(3) RecordSource 属性

在这里,我们将在 RecordSource 属性页的命令文本下键入如下的 SQL 语句:

```
Select *  
From Authors
```

(4) Visible 属性

在这里,我们设置 Adodc1(0)的 Visible 属性为 True。

对 Adodc1(1)控件来说,我们改变如下几个属性的值:

(1) Captions 属性

Adodc1(1)控件的默认属性为 Adodc1,在这里我们把它改成 Publishers。

(2) ConnectionString 属性

用户在 ConnectionString 属性页的“使用 ODBC 数据资源名称”项中选择 Biblio 数据库。用户最后可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

(3) RecordSource 属性

在这里,我们将在 RecordSource 属性页的命令文本下键入如下的 SQL 语句:

```
Select *  
From Publishers
```

(4) 在这里,我们设置 Adodc1(1)的 Visible 属性为 False。

对 Adodc1(2)控件来说,我们改变如下几个属性的值:

(1) Captions 属性

Adodc1(2)控件的默认属性为 Adodc1,在这里我们把它改成 Titles。

(2) ConnectionString 属性

用户在 ConnectionString 属性页的“使用 ODBC 数据资源名称”项中选择 Biblio 数据库。用户最后可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

(3) RecordSource 属性

在这里,我们将在 RecordSource 属性页的命令文本下键入如下的 SQL 语句:

```
Select *  
From Titles
```

(4) 在这里,我们设置 Adodc1(2)的 Visible 属性为 False。

对 DataGrid1控件来说,我们改变它的 DataSource 属性为 Adodc1(0)。这使得在程序运行的一开始,DataGrid1和 Adodc1(0)控件相联系。

对于 Command1控件来说,我们改变它的 Caption 属性为离开。它表示单击此按钮就终止应用程序的执行。

最后,更改 Form1的 Caption 属性为更改数据源实例,它表示本程序是一个给读者演示的

更改数据源的例子。

调整各控件的大小及其位置,并使得三个 ADO Data 控件在窗体上的同一个位置。完成以后如图8.5所示。

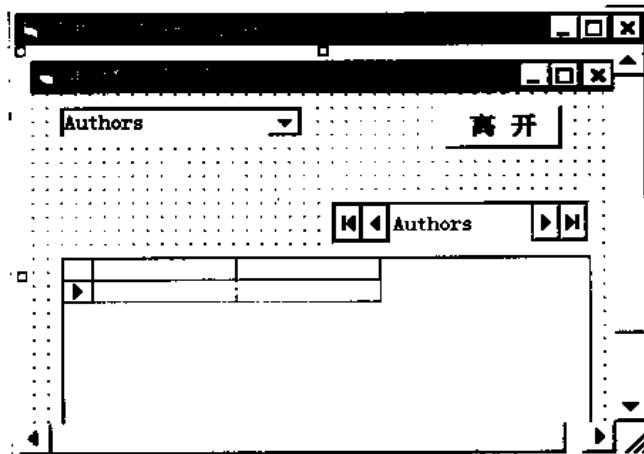


图8.5 完成属性设置后的窗体

现在,我们给具体的控件加上代码。在这里,有两个控件需要加代码。一个为 Combo1 控件,在它的 Click 事件中加入一段代码,具体代码如下。由于这个代码段比较简单,我们在这里就不再解释了,相信读者完全有能力来自己读懂它。

```
Private Sub Combo1_Click()  
    Select Case Combo1.Text  
        Case "Authors"  
            Screen.MousePointer = vbHourglass  
            Set DataGrid1.DataSource = Adodc1(0)  
            Adodc1(0).Visible = True  
            Adodc1(1).Visible = False  
            Adodc1(2).Visible = False  
            Screen.MousePointer = vbArrow  
        Case "Publishers"  
            Screen.MousePointer = vbHourglass  
            Set DataGrid1.DataSource = Adodc1(1)  
            Adodc1(0).Visible = False  
            Adodc1(1).Visible = True  
            Adodc1(2).Visible = False  
            Screen.MousePointer = vbArrow  
        Case "Titles"  
            Screen.MousePointer = vbHourglass  
            Set DataGrid1.DataSource = Adodc1(2)  
            Adodc1(0).Visible = False  
            Adodc1(1).Visible = False  
            Adodc1(2).Visible = True  
            Screen.MousePointer = vbArrow  
    End Select  
End Sub
```

```
End Select  
End Sub
```

在 Caption 属性为离开的 Command1 控件的 Click 事件中,我们加入如下代码。它表示在用户单击离开按钮时退出应用程序。

```
Private Sub Command1_Click()  
End  
End Sub
```

8.2 关于列的实例

8.2.1 在 DataGrid 中添加列

应用程序在运行过程中,DataGrid 控件中列的数目是可以变化的。用户可以给相应的 DataGrid 控件添加或删除一列或者若干列。在数据库程序的实际应用中,添加列的功能是会经常使用到的。读者可以想象,在实际数据库的操作中,我们经常会对数据库中某一个具体的表进行添加或者删除其中的一列。在接下来的例子中,我们就要演示如何往 DataGrid 控件中添加一列。

首先,我们在新建标准的 Exe 工程的 Form1 上放置一个 DataGrid、一个 ADO Data 控件、一个 ComboBox 和两个 CommandButton 控件,如图8.6所示。

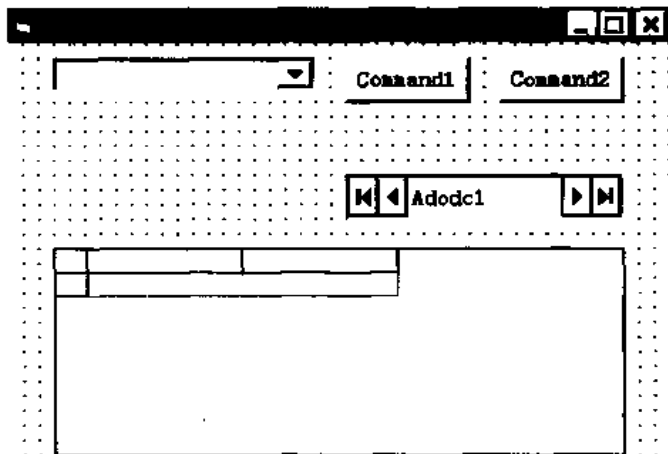


图8.6 在窗体上放置控件

接下来,我们改变和设置如下几个控件的属性。

对 Combo1 控件来说,我们改变如下几个属性的值:

(1) Text 属性

Combo1 控件的默认属性为 Combo1,在这里我们把它改成空。

(2) List 属性

打开 List 属性的下拉框,在下拉框中分行地输入 Name、Company Name、Address 和 City。这样在程序运行时,用户可以在 Combo1 的下拉框中选取 Name、Company Name、Ad-

dress 或 City。

(3) Item Data 属性

Item Data 属性用来返回或设置 ComboBox 控件中每个项目具体的编号。在这里,我们打开 Item Data 属性的下拉框,在下拉框中分行地输入0、1、2和3。

对 Adodc1控件来说,我们改变如下几个属性的值:

(1) Captions 属性

Adodc1控件的默认属性为 Adodc1,在这里我们把它改成 Publishers。它表示 Adodc1控件所对应的是 Publishers 表。

(2) ConnectionString 属性

用户在 ConnectionString 属性页的“使用 ODBC 数据资源名称”项中选择 Biblio 数据库。用户最后可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

(3) RecordSource 属性

在这里,我们将在 RecordSource 属性页的命令文本下键入如下的 SQL 语句:

```
Select PubID,State,Zip,Telephone,Fax  
From Publishers
```

对 DataGrid1控件来说,我们改变它的 DataSource 属性为 Adodc1。这使得在程序运行时,DataGrid1和 Adodc1控件相联系。

对于 Command1控件来说,我们改变它的 Caption 属性为添加,它表示单击此按钮就会往 DataGrid1控件中添加一列。对于 Command1控件来说,我们改变它的 Caption 属性为离开,它表示单击此按钮就终止应用程序的执行。

最后,更改 Form1的 Caption 属性为添加列的实例,它表示本程序是一个给读者演示如何往 DataGrid 控件中添加一列的例子。在完成属性的设置以后,我们就要往控件中添加程序代码,这些代码无疑是程序运行的关键。

首先,在 Form 的 Load 事件中加入一段程序代码,如下所示。Form_Load()事件是在程序运行的一开始触发的。在 Form_Load()事件的代码段中,我们要检查 Combo1控件的 Text 属性是否为空。如果为空,表示在 Combo1控件的文本框中无任何的显示,此时添加按钮无法同具体需要添加的列相联系,添加按钮应为不可用。否则,添加按钮为可用。

```
Private Sub Form_Load()  
    If Combo1.Text = "" Then  
        Command1.Enabled = False  
    Else  
        Command1.Enabled = True  
    End If  
End Sub
```

接着我们在 Combo1控件的 Click 事件中加入一段程序代码,如下所示。用户要在 Combo1的下拉框中来选择需要添加的列。当用户进行选择时,很明显会触发 Combo1控件的 Click 事件。如果用户选择了一个列名,那么 Combo1控件的 Text 属性就不再为空,添加按钮为可用。反之,如果用户在触发 Combo1控件的 Click 事件时并没有选择一个具体的列名,那么 Combo1控件的文本框仍然为空,即 Text 属性为空。根据 Form_Load()代码段的解释,当 Combo1控

件的 Text 属性为空时,添加按钮为不可用。

```
Private Sub Combo1_Click()  
    If Combo1.Text = "" Then  
        Command1.Enabled = False  
    Else  
        Command1.Enabled = True  
    End If  
End Sub
```

然后我们在 Command1控件(添加按钮)的 Click 事件中加入一段程序代码,如下所示:

```
Private Sub Command1_Click()  
    Dim c As Column  
    Dim i As Integer  
    i = Combo1.ListIndex  
    Set c = DataGrid1.Columns.Add(DataGrid1.Columns.Count)  
    With c  
        .Visible = True  
        .Width = 1500  
        .Caption = Combo1.Text  
        .Alignment = dbgRight  
    End With  
    Combo1.Text = ""  
    Combo1.RemoveItem i  
    Command1.Enabled = False  
End Sub
```

在上面的程序代码中,Dim c As Column 是定义一个 Column 对象的变量。在语句 i = Combo1.ListIndex 中,ListIndex 返回 Combo1控件中当前选择项目的索引值。Set c = DataGrid1.Columns.Add(DataGrid1.Columns.Count)是在 DataGrid1控件中添加一列,并把这一列赋给 Column 对象的变量 c。在接下来的 With 语句段中,设置新加的列的属性。最后,应该把已添加的列名从 Combo1的下拉框中删除,并恢复 Combo1控件的 Text 属性和 Enabled 属性。

在 Command2(离开按钮)中的代码如下所示:

```
Private Sub Command2_Click()  
    End  
End Sub
```

8.2.2 在 DataGrid 中删除列

同在 DataGrid 控件中添加一列对应,有时我们需要从 DataGrid 控件中删除一个原来存在的列。下面我们就来演示一个这样的实例,本实例是在8.2.1节中实例的基础上进行的。

首先,我们改变8.2.1节中实例的界面,改变后的界面如图8.7所示。在完成界面的修改后,我们把 Form1的 Caption 属性改为删除列的实例;把名为 Command1的 CommandButton 控件的 Caption 属性改为删除,它表示从 DataGrid1控件中删除一列;把名为 Adodc1的 ADO Data

控件的 RecordSource 属性改为 Select * From Publishers。这样,我们就完成了本实例中属性的修改。

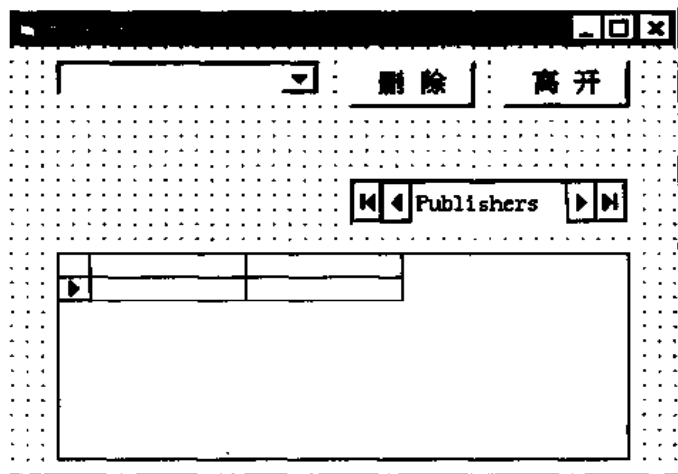


图8.7 删除列的实例的界面

在完成上述的修改之后,我们接下来进行程序代码的改变。在本例子中,我们只需改变 Command1 控件中的程序代码,如下所示:

```
Private Sub Command1_Click()  
    Dim c As Column  
    Dim i As Integer  
    i = Combo1.ListIndex  
    DataGrid1.Columns.Remove(Combo1.Text)  
    Combo1.Text = ""  
    Combo1.RemoveItem i  
    Command1.Enabled = False  
End Sub
```

在上述程序代码中,我们要注意的语句是 DataGrid1.Columns.Remove(Combo1.Text)。此语句调用 DataGrid1 控件中的 Columns 对象的 Remove 方法来删除一列。在 Remove 方法的参数调用中,原先是应该传入一个列的序号值。不过,我们可以用列名来代替列的序号。这一点请读者千万要注意。读者可以考虑一下,如果 Remove 方法的参数调用中只能传入一个列的序号值,那么上述的程序代码应该如何进行修改。

8.3 DataGrid 视图

一个“拆分”的网格使最终用户对相同的数据可以拥有多个视图。例如,假设有一个由10个字段组成的大表。在这种情况下,在控件中察看的记录集将有10列宽,除非窗体非常宽,否则用户将无法同时看见所有列的内容。更进一步,假设用户只对第一列和最后一列感兴趣。为了能同时看到在两端的列(不重新安排列的顺序),可以对网格进行拆分。

在接下来的例子中,我们就来演示一个拆分的实例。假设程序在运行时 DataGrid 控件中

显示的是 Biblio 数据库中 Titles 表的全部内容,当我们单击窗体上的拆分按钮时,DataGrid 控件分为两个部分,其中第一个部分显示 Titles 表中的 Title 列内容,第二个部分则显示 Titles 表中的 Comments 内容。程序的界面如图8.8所示。

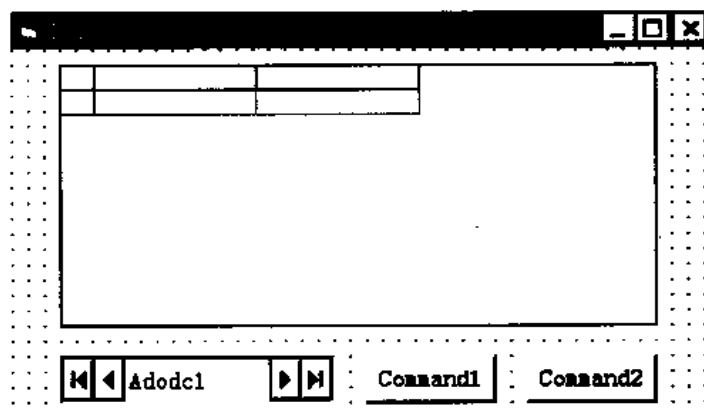


图8.8 DataGrid 视图的界面

在这里,ADO Data 控件的 RecordSource 属性为 Select * From Titles。其它控件的一些属性相信读者能够用上面的说明和图8.8来自行地设置。下面我们来看程序代码。显而易见,在退出按钮的 Click 事件中我们所加的是 End 语句。本实例的关键代码是在显示为拆分按钮的 CommandButton 控件中,代码段如下。

```
Private Sub Command1_Click()  
    DataGrid1.Splits.Add 1  
    With DataGrid1  
        .Splits(0).ScrollGroup = 1  
        .Splits(1).ScrollGroup = 1  
    End With  
    For i = 0 To DataGrid1.Splits(0).Columns.Count - 1  
        If DataGrid1.Splits(0).Columns(i).DataField <> "Title" Then  
            DataGrid1.Splits(0).Columns(i).Visible = False  
        End If  
    Next i  
    For i = 0 To DataGrid1.Splits(1).Columns.Count - 1  
        If DataGrid1.Splits(1).Columns(i).DataField <> "Comments" Then  
            DataGrid1.Splits(1).Columns(i).Visible = False  
        End If  
    Next i  
End Sub
```

首先,我们使用 DataGrid1.Splits.Add 1语句来为 DataGrid1控件添加一个拆分。其中,Splits 对象表示 DataGrid1控件的拆分,Visual Basic 默认规定,当创建 DataGrid 对象时,它将包含一个 Splits 对象。Add 是一个用来进行添加拆分的方法。在添加了一个拆分之后,为了使得拆分同步,我们设置两个拆分的 ScrollGroup 属性为同一个值。接下来的两个循环语句是为了在 Splits(0)中显示 Title 列,在 Splits(1)中显示 Comments 列。

8.4 使用书签的实例

书签在 Visual Basic 中能够起到标识记录的作用。读者可以想象,对于取消了指针类型的 Visual Basic 语言来说,书签的引入具有多么重大的意义。在 Visual Basic 中,提供了两种表示书签的属性 Bookmarks 和 SelBookmarks。Bookmarks 和 SelBookmarks 提供了标记记录的一种手段,当编写应用程序中的特定功能(诸如允许最终用户手工地选择多个不相邻的记录,进行所选记录的大批更新)时,这就很有必要。在这些情况中,需要标记哪些记录已被选择,因此可以使用 SelBookmarks 集合及其属性。在这里读者还需要注意的是,有两个函数,分别是 CellText 和 CellValue 方法,需要有书签的标记才能正确执行。

8.4.1 标记用户的选择

SelBookmarks 集合包含所有选定的记录的书签。当最终用户手工选择记录时(即在单击时按住 Ctrl 键),每一个选定的记录的书签都会加入到该集合。使用标准的循环,用户可以知道已经选定了什么,也可以保存书签(因为可能需要恢复某个值),以及执行相应的操作。在下面的程序段中,我们声明一个数组,并把书签保存在该数组中。当用户需要使用书签时,就可以从该数组中取得相应的值。

```
Dim i as Integer
Dim intCount As Integer
intCount = DataGrid1.SelBookmarks.Count - 1
ReDim arrSelBK(intCount)
For i = 0 To intCount
    ArrSelBK(i) = DataGrid1.SelBookmarks(i)
Next i
```

8.4.2 通过添加到 SelBookmarks 集合来选择记录

通过将记录添加到这个 SelBookmarks 集合,是可以在程序中选定记录的。在接下来的程序实例中,我们就要演示如何通过程序代码来选定记录。首先,新建一个标准的 Exe 工程,在 Form1 窗体中添加一个 DataGrid 控件、一个 ADO Data 控件和两个 CommandButton 控件。调整各个控件的大小及位置,完成后如图 8.9 所示。

接下来,我们改变和设置如下几个控件的属性。

对 Adodc1 控件来说,我们改变如下几个属性的值:

(1) Captions 属性

Adodc1 控件的默认属性为 Adodc1,在这里我们把它改为 Titles。它表示 Adodc1 控件所对应的是 Titles 表。

(2) ConnectionString 属性

用户在 ConnectionString 属性页的“使用 ODBC 数据资源名称”项中选择 Biblio 数据库。用户最后可以在 ConnectionString 属性中看到“DSN = Biblio”的字符串形式。

(3) RecordSource 属性

在这里,我们将在 RecordSource 属性页的命令文本下键入如下的 SQL 语句:

```
Select *
From Titles
```

对 DataGrid1 控件来说,我们改变它的 DataSource 属性为 Adodc1。这使得在程序运行时,DataGrid1 和 Adodc1 控件相联系。

对于 Command1 控件来说,我们改变它的 Caption 属性为书签。对于 Command1 控件来说,我们改变它的 Caption 属性为离开,它表示单击此按钮就终止应用程序的执行。

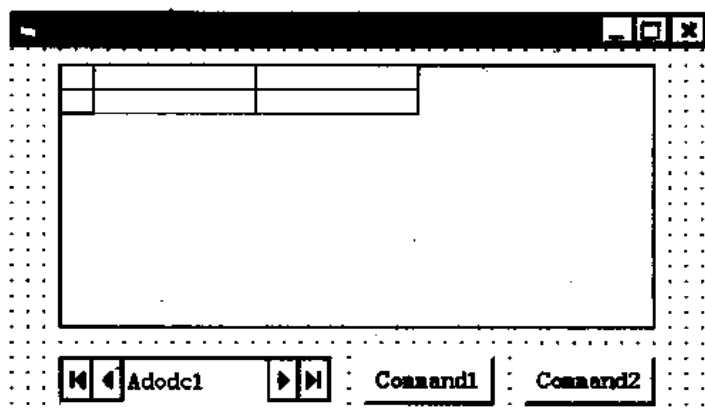


图8.9 在窗体上放置各控件

最后,更改 Form1 的 Caption 属性为通过添加到 SelBookmarks 集合来选择记录。完成后的界面如图8.10所示。在完成属性的设置以后,我们就要往控件中添加程序代码了。

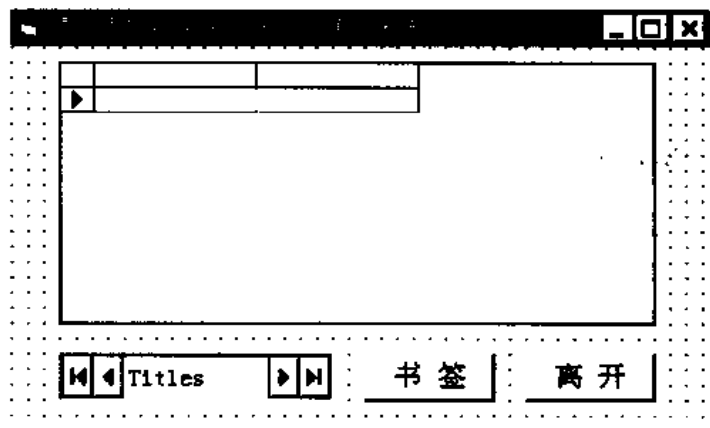


图8.10 完成属性设置后的界面

在名为 Command1 的 CommandButton 控件中,我们添加如下的代码段:

```
Private Sub Command1_Click( )
    Dim rs As Recordset
    Set rs = Adodc1.Recordset
    While Not rs.EOF
        DataGrid1.SelBookmarks.Add rs.Bookmark
    
```

```
rs.MoveNext
Wend
End Sub
```

上述的代码段中,我们通过一个 RecordSet 对象的变量 rs 来往 DataGrid1 控件中添加书签。用户可以运行一下该应用程序,从中观察程序在执行时具体书签的添加过程。在名为 Command2 的 CommandButton 控件中,我们添加如下的代码:

```
Private Sub Command2_Click()
    End
End Sub
```

8.5 类模块一起使用 DataGrid 控件

8.5.1 使用类模块创建数据源

如果想要访问以自定义格式或以 ODBC 驱动程序不直接支持的格式存放的数据,可以创建一个类来封装该数据。然后可以编写该类的自定义函数来检索这些数据。这样该类就变成了一种数据源,可以被任何数据使用者(如 DataGrid 控件)使用。

首先,我们新建一个标准的 Exe 工程。为了确保 ADO 对象的可使用性,我们在“工程”菜单中,单击“引用”。在“引用”对话框中,单击“Microsoft ActiveX Data Objects 2.0 Library”。在确保了 ADO 对象的可用性之后,给窗体添加一个 DataGrid 控件。改变 Form1 的 Caption 属性为与类模块一起使用 DataGrid 控件。

由于本实例是通过使用类模块来和 DataGrid 控件相联系,所以在完成了上面的操作之后,需要在“工程”在菜单中,单击“添加类模块”菜单项来给工程添加一个数据类模块。添加了类模块之后,我们需要改变类模块的 Name 属性为 NamesData;改变类模块的 DataSourceBehavior 属性为 vbDataSource。DataSourceBehavior 是用来设置 Class 是否作为数据源的属性。完成后如图 8.11 所示。

随后,我们在该类模块的 Declarations 部分,创建一个 ADODB Recordset 变量,如下所示:

```
Option Explicit
Private WithEvents rsNames As ADODB.RecordSet
```

在这里的变量声明中,我们用到了 WithEvents 关键字。WithEvents 关键字用在 Private 语句中是一个可选的关键字,它说明紧跟它之后的变量是用来响应由 ActiveX 对象所触发的事件的对象变量。这种使用方法只有在类模块中才是合法的。使用 WithEvents,可以用来定义任意个所需的单变量,但不能使用 WithEvents 来创建数组。并且,需要注意的是 New 关键字和 WithEvents 不能一起使用。

在类模块的 Initialize 事件中,我们加入如下的代码:

```
Private Sub Class_Initialize()
    DataMembers.Add "ID"
```

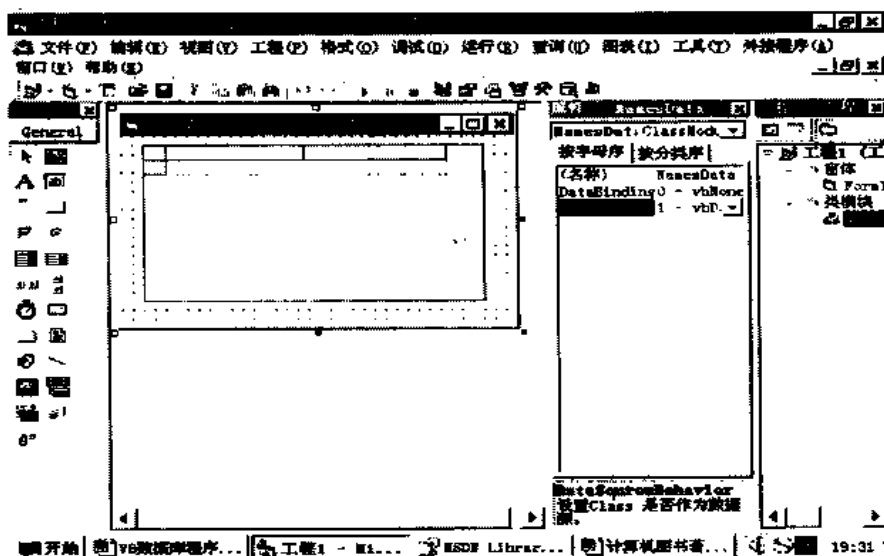


图8.11 与类模块一起使用 DataGrid 控件

```

DataMembers.Add "Names"
Set rsNames = New ADODB.Recordset
With rsNames
    .Fields.Append "ID", adInteger
    .Fields.Append "Name", adBSTR, 255
    .CursorType = adOpenStatic
    .LockType = adLockOptimistic
    .Open
End With
Dim i As Integer
For i = 1 To 10
    rsNames.AddNew
    rsNames!ID = i
    rsNames!Name = "Name " & i
    rsNames.Update
Next i
rsNames.MoveFirst
End Sub

```

在上述的程序代码中,我们首先使用 DataMembers 的 Add 方法将新的数据成员的名称添加到 DataMember 集合中,它使得其它对象可以看见这些可用的 DataMembers。然后我们创建一个具有两个字段的 RecordSet 对象变量,并打开该 RecordSet 对象变量。在 With 语句中,我们给 RecordSet 对象变量赋列名。第一个列是具有一个整数的数据类型,在 .Fields.Append "ID", adInteger 语句中,adInteger 表示第一列的是数据类型为整型。第二个列是一个最大可达 256 个字符的字符串,在 .Fields.Append "Name", adBSTR, 255 语句中,adBSTR 表示的是名为 "Name" 的列是一个可达 256 个字符的字符串类型。接下来使用的 CursorType 属性用来指定打开 Recordset 对象时应该使用的游标类型。Recordset 关闭时 CursorType 属性为

读/写,而 Recordset 打开时该属性为只读。在表8.1中,我们给出了 CursorType 属性的说明。

表8.1 CursorType 属性的说明

常 量	说 明
AdOpenForwardOnly	仅向前游标,默认值。除了只能在记录中向前滚动外,与静态游标相同。当只需要在记录集中单向移动时,使用它可提高性能
AdOpenKeyset	键集游标。尽管从用户的记录集中不能访问其他用户删除的记录,但除无法查看其他用户添加的记录外,键集游标与动态游标相似。仍然可以看见其他用户更改的数据
AdOpenDynamic	动态游标。可以看见其他用户所作的添加、更改和删除。允许在记录集中进行所有类型的移动,但不包括提供者不支持的书签操作
AdOpenStatic	静态游标。可以用来查找数据或生成报告的记录集合的静态副本。另外,对其他用户所作的添加、更改或删除不可见

在上面的程序段中,CursorType 被设置为 OpenStatic。它表示一个可更新的对一组记录的快照。LockType 属性则指示编辑过程中对记录使用的锁定类型。关于 LockType 属性的说明见表8.2。在这里我们设置 CursorType 属性为 LockOptimistic,它表示允许对该 RecordSet 对象进行更新。

表8.2 LockType 属性的说明

常 量	说 明
AdLockReadOnly	默认值,只读。无法更改数据
AdLockPessimistic	保守式记录锁定(逐条)。提供者执行必要的操作确保成功编辑记录,通常采用编辑时立即锁定数据源的记录的方式
AdLockOptimistic	开放式记录锁定(逐条)。提供者使用开放式锁定,只有在调用 Update 方法时锁定记录
AdLockBatchOptimistic	开放式批更新。用于与立即更新模式相反的批更新模式

在设置完 RecordSet 对象变量的属性后,我们使用 Open 方法来打开 RecordSet 对象变量。打开名为 rsName 的 RecordSet 对象变量后,我们使用一个 For...Next 循环来往变量中添加记录。rsNames.AddNew 语句用来添加一条新的记录,rsNames!ID = i 语句是用来给 rsNames 变量的 ID 域赋值,rsNames!Name = "Name" & i 语句用来给 rsNames 变量的 Name 域赋值。在添加完记录后使用 rsNames.Update 语句来更新数据库。最后一句语句 rsNames.MoveFirst 是用来移到记录集的开始。

在该类的 GetDataMember 事件中,我们添加下述代码:

```
Private Sub Class_GetDataMember(ByVal DataMember As String,Data As Object)
    Set Data = rsNames
End Sub
```

上面的代码表示只要发生该事件,即当该类对象被绑定到一个数据使用者,如 DataGrid 控件时,代码将返回该 RecordSet 对象。事实上,GetDataMember 事件是当此事件的类对象被绑定到一个具体的数据使用者时触发的。

接着,我们到名为 Form1 的窗体中创建一个数据类的对象变量。代码如下。在这里,我们声明了一个 NamesData 类对象的变量 datNames。

Option Explicit

Private datNames As NamesData

在 Form 对象的 Load 事件的代码中,将 DataGrid 控件的 DataSource 设置为该类对象。代码如下:

```
Private Sub Form_Load()
    ' 创建一个新的 NamesData 对象
    Set datNames = New NamesData
    ' 将这个 DataGrid 绑定到新的数据源 datNames
    Set DataGrid1.DataSource = datNames
End Sub
```

8.5.2 结果集事件的编程

在完成了8.4.1节的程序实例后,我们现在要往上面的这个实例中添加一些代码,从而实现结果集的编程。所谓对结果集的编程,事实上也就是对上述 Recordset 对象的事件进行编程。

在该类模块中,单击“对象”框(位于左上角),然后单击“rsNames”。在“过程/事件”框(在右上角)中,下拉列表将显示这个 Recordset 对象的所有事件,如图8.12所示。接下来我们就要来具体地解释一下这些与类有关联的事件。

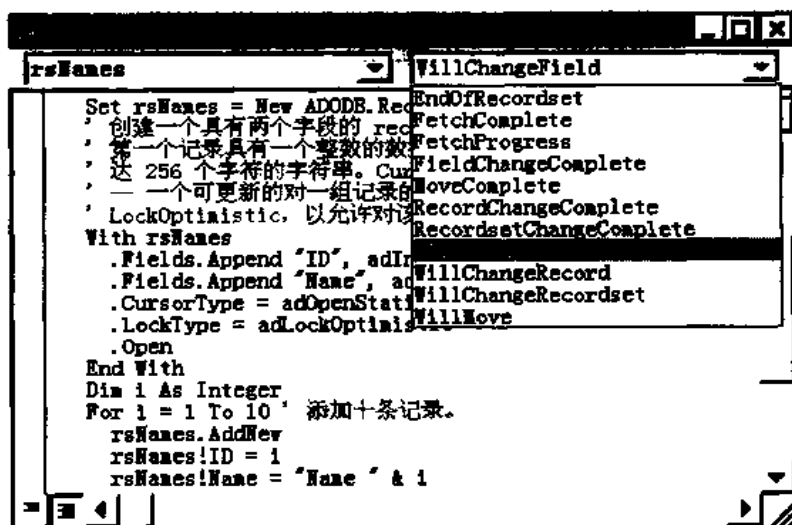


图8.12 显示类模块中的对象

(1) EndOfRecordset 事件

如果应用程序试图移动到超过 Recordset 结尾的行时,系统将调用 EndOfRecordset 方法。

(2) FetchComplete 事件

FetchComplete 事件在长异步操作中所有记录已经恢复(获取)到记录集之后调用。

(3) FetchProgress 事件

FetchProgress 事件在长时间的异步操作期间定期调用,以便报告当前已经恢复(获取)到

Recordset 中的行的数目。

(4) WillChangeField 事件

WillChangeField 事件在挂起操作对 Recordset 中一个或多个 Field 对象的值进行更改前调用。

(5) FieldChangeComplete 事件

FieldChangeComplete 事件是在一个或多个 Field 对象的值已经更改后调用。

(6) MoveComplete 事件

MoveComplete 事件是在 Recordset 的当前位置更改后调用。

(7) RecordChangeComplete 事件

RecordChangeComplete 事件是在一个或多个记录更改之后调用。

(8) RecordsetChangeComplete 事件

RecordsetChangeComplete 事件是在 Recordset 更改后调用。

(9) WillChangeRecord 事件

WillChangeRecord 事件是在 Recordset 中的一个或多个记录(行)更改之前调用。

(10) WillChangeRecordset 事件

WillChangeRecordset 事件是在挂起操作更改 Recordset 之前调用。

(11) WillMove 事件

WillMove 事件是在挂起操作更改 Recordset 中的当前位置前调用。

由于 Visual Basic 是完全支持面向对象的编程的,对象是被封装的。它同时包含其代码和数据,这比传统的编写代码方法更容易维护。Visual Basic 对象具有属性、方法和事件。属性是描述对象的数据,方法告诉对象应做的事情,事件是对象所产生的事情,事件发生时可以编写代码进行处理。在 Visual Basic 中,对象是由类创建的,因此对象被说成是类的一个实例。类定义对象的界面,不管对象是否是公共的和在什么环境下建立的都是这样。类的描述保存在类型库中,而且通过对象浏览器可以查看。图8.13显示了 NamesData 对象的成员。在这里我们假设读者已经具有了面向对象编程的概念。如果读者以前没有接触过面向对象的编程,那么就请先参考有关的书籍。

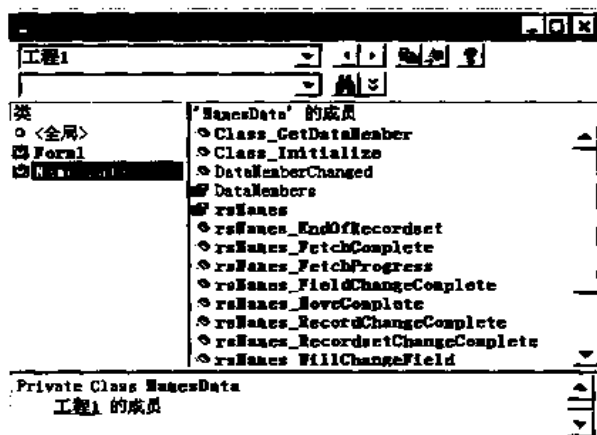


图8.13 NamesData 对象浏览器

在 Visual Basic 中,类属性定义最简单的方法,是向类的模块添加公共变量。例如,只要在名为 Account 的类中声明两个公共变量,可以非常简单地创建 Account 类,如下所示:

```
Public Balance As Double
```

```
Public Name As String
```

给类创建私有数据也很简单,只需声明一个 Private 变量,它就能被该类模块中的编码所访问。如下所示:

```
Private Balance As Double
```

```
Public Name As String
```

在 Visual Basic 中还有第三种方法来创建属性,那就是通过属性过程来实现。如果只能通过声明公共变量来创建属性,那么数据隐藏就没有多大用处。Visual Basic 提供了表8.3所列的三种属性过程。

表8.3 三种属性过程

过 程	目 的
Property Get	返回属性的值
Property Let	设置属性的值
Property Set	设置对象属性的值(即,该属性含有对象引用)

从上表可见,每一个属性过程都有自己特殊的功能。典型的属性由一对属性过程组成:Property Get 检索属性的值,Property Let 或 Property Set 给属性赋值。接下来,我们就要使用属性过程给 NamesData 类添加一个属性。下面的代码演示了如何先给类添加属性。当从其它对象调用该对象时,这一属性将返回该类的 RecordCount。

```
Public Property Get RecordCount( ) As Long  
    RecordCount = rsNames.RecordCount  
End Sub
```

第九章 使用 Data Report 实例

数据报表(Data Report)在数据库应用程序设计中无所不在。在一个实际的数据库应用中,使用数据报表可以实现诸如查询、统计、汇总等功能。例如,我们设计一个单位系统的数据库应用程序,那么每个月底本单位各个职工的工资、奖金和福利待遇等需要以报表的形式上报给上级部门,在这时我们就需要使用 Data Report 控件来显示和打印报表了。事实上,一个较为完整的数据库应用程序都离不开 Data Report 控件的使用。在本章中,我们就要来讲述如何在应用程序中使用 Data Report 控件。

在本章中我们主要讲述以下两个部分:

- 创建数据报表
- 扩展数据报表

9.1 创建数据报表

9.1.1 数据报表设计器

数据报表设计器是 Visual Basic 6.0版本中新添加的功能。使用数据报表设计器可以使用户在完全可视化的环境下进行数据报表的设计工作。在 Visual Basic 的先前版本中,对数据库的设计工作(包括对数据报表的设计)是在各自不同的集成开发环境中进行的。例如,在 Visual Basic 4.0版本中,建造数据库的唯一方法是使用 Data Manager。Data Manager 实际上是一个可独立于 Visual Basic 运行的独立应用程序,它完全是用 Visual Basic 4.0编写的,应用程序的名称为 DATAMGR.EXE。

使用过 Data Manager 的读者可能知道,使用它来创建一个数据报表并没有想象中的那么方便。Visual Basic 添加了数据报表设计器,它使得在完全可视化的环境下进行数据报表的设计工作成为可能。用户要在应用程序中添加一个 Data Report,可以单击“工程”菜单下的“添加 Data Report”菜单选项来完成,如图9.1所示。

在 Visual Basic 中的数据报表设计器有如下几个特性。

(1) 对字段的拖放功能

对字段的拖放功能是指把字段从 Microsoft 数据环境设计器拖到数据报表设计器。当进行这一操作时,Visual Basic 会自动地在数据报表上创建一个文本框控件,并设置被拖放字段的 DataMember 和 DataField 属性。也可以把一个 Command 对象从数据环境设计器拖到数据报表设计器,在这种情况下,对于每一个 Command 对象包含的字段,将在数据报表上创建一个文本框控件;每一文本框的 DataMember 和 DataField 属性将被设置为合适的值。

(2) Toolbox 控件

数据报表设计器以它自己的一套控件为特色。当数据报表设计器被添加到工程时,控件被自动创建在一个名为 DataReport 的新“工具箱”选项卡上。多数的控件在功能上与 Visual Basic 内部控件相同,并且包括 Label、Shape、Image、TextBox 和 Line 控件。第六个控件,即

Function 控件,自动地生成如下四种信息中的一种:Sum、Average、Minimum 或 Maximum。它们分别对应于求和、求平均、最小值或最大值。

(3) 打印预览

打印预览是通过使用 Show 方法来预览报表的,然后生成数据报表并显示在它自己的窗口内。如果要在打印预览方式中显示报表,首先必须要在计算机上安装一台打印机。

(4) 打印报表

用户可以通过调用 PrintReport 方法,以编程方式打印一个报表。当数据报表处于预览方式,用户也可以通过单击工具栏上的打印机图标打印报表。如果要打印报表,首先必须要在计算机上安装一台打印机。

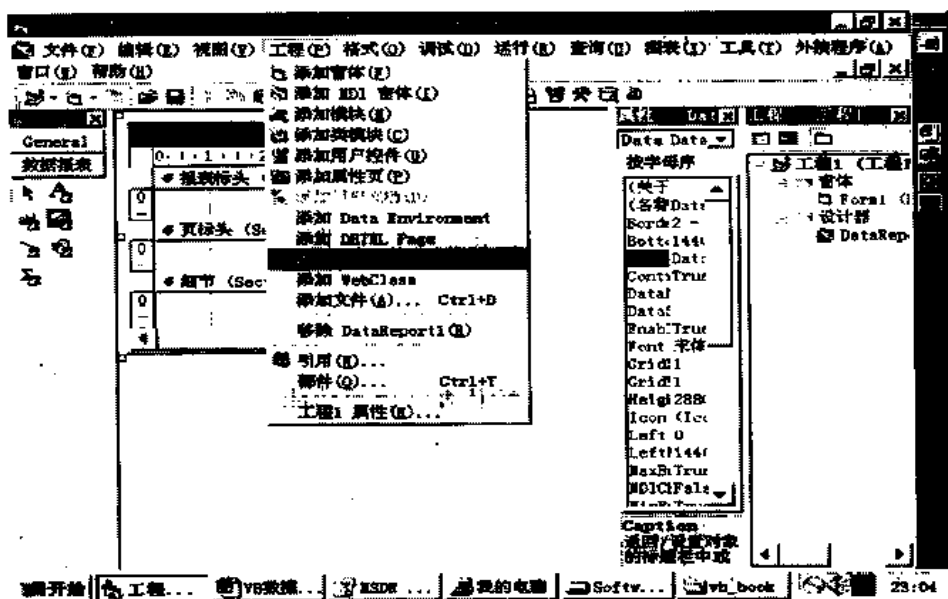


图9.1 数据报表设计器

(5) 文件导出

用户可以通过使用 ExportReport 方法来导出数据报表信息,导出格式包括了 HTML 和文本。

(6) 导出模板

用户完全可以创建一个文件模板集合,以使它同 ExportReport 方法一起使用。这对于以多种格式(每种都是报表类型剪裁)导出报表是很有用的。

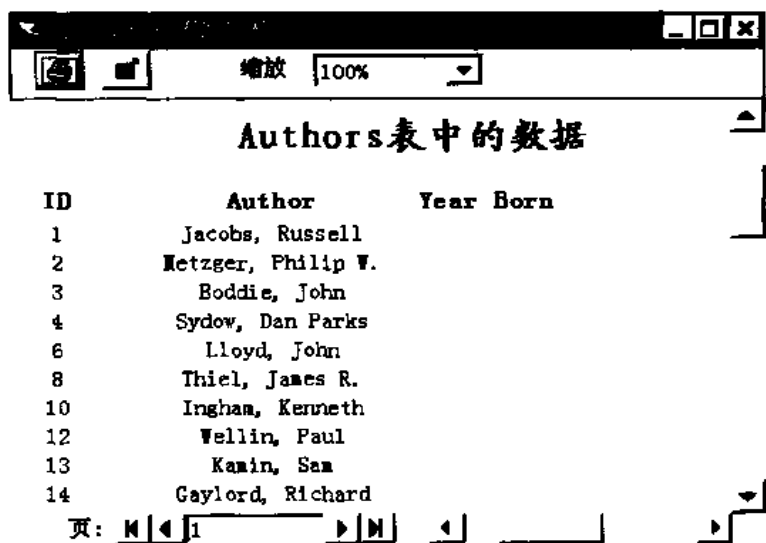
(7) 异步操作

DataReport 对象的 PrintReport 和 ExportReport 方法是异步操作。使用 Processing-Timeort 事件可以监视这些操作的状态,并取消任何花费时间过长的操作。

9.1.2 一个简单数据报表的实例

在本小节中,我们要创建一个简单的数据报表的实例。通过这个例子,读者应该学会如何使用数据报表设计器来生成一个具体的数据报表。在这里,我们使用 Biblio 数据库中的 Authors 表来作为数据报表的数据源。数据报表完成后的运行界面如图9.2所示。接下来,我们就

要讲述设计程序的具体步骤。



ID	Author	Year Born
1	Jacobs, Russell	
2	Metzger, Philip W.	
3	Boddie, John	
4	Sydow, Dan Parks	
6	Lloyd, John	
8	Thiel, James R.	
10	Ingham, Kenneth	
12	Wellin, Paul	
13	Kamin, Sam	
14	Gaylord, Richard	

图9.2 显示 Authors 表中的数据

(1) 数据环境设计器的设计

首先,我们在“工程”菜单上,单击“添加 Data Enviroment”菜单选项来向工程中添加一个设计器。如果设计器没有在“工程”菜单上列出,则单击“部件”菜单选项。单击“设计器”选项卡。并单击“数据环境”把设计器添加到菜单上。完成后的数据环境设计器如图9.3所示。

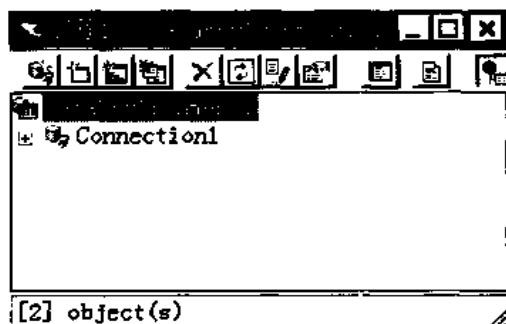


图9.3 数据环境

在添加了一个 DataEnvironment 后,用户就会看到在 DataEnvironment1 下有一个名为 Connection1 的连接。选中 Connection1 并单击鼠标右键,在弹出菜单中选择“属性...”菜单项,就会弹出一个设置数据链接属性的窗口,如图9.4所示。在数据链接属性窗口的“提供者”选项卡中,我们选择 Microsoft Jet 4.0 OLE DB Provider 项,这表示数据源的提供者 of Microsoft Jet 4.0。单击“连接”选项卡,我们在“选择或输入数据库名称”下面的下拉框中选择具体路径下的 Biblio 数据库。在选择完数据库之后,我们可以单击图9.4窗口中的“测试连接”按钮来测试同数据库的连接是否成功。如果测试的结果为不成功,就请读者重新输入或选择数据链接属性窗口中的具体项。

接下来,我们选中 Connection1 并单击鼠标右键,在弹出菜单中选择“添加命令”菜单项。这

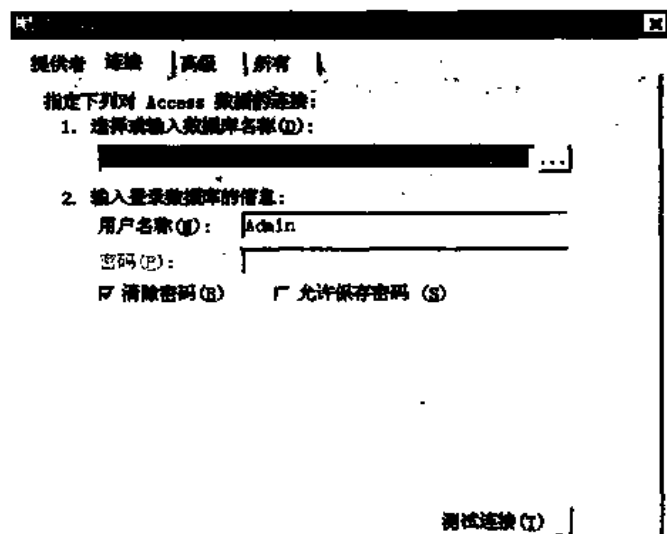


图9.4 数据链接属性

样,我们就会在 Connection1 下面见到一个名为 Command1 的命令。选中 Command1 并单击鼠标右键,在弹出菜单中选择“属性...”菜单项,出现如图9.5所示的窗口。

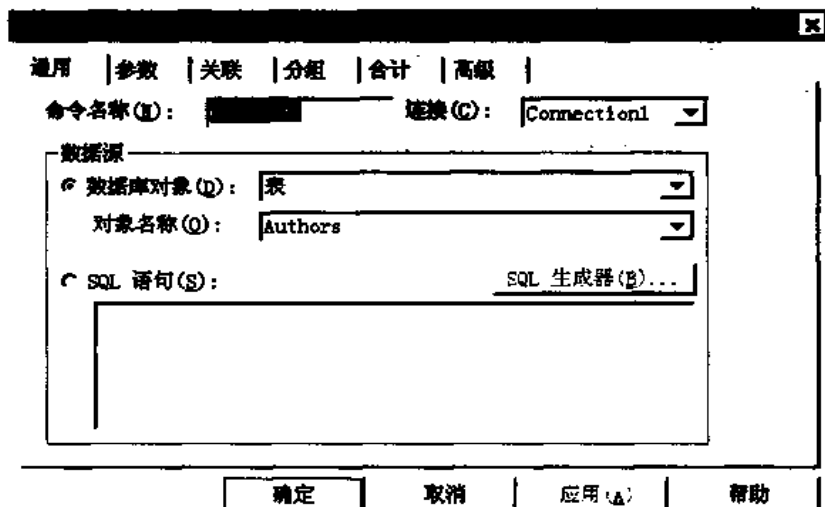


图9.5 命令的属性

在“数据库对象”右边的下拉框中选择“表”,在“对象名称”右边的下拉框中选择“Authors”。“数据库对象”右边的下拉框中有表、存储过程、视图和同义词四种选择,分别代表所对应的数据源为表、存储过程、视图和同义词。“对象名称”右边的下拉框中则列出了 Biblio 数据库中的各个具体的表。完成之后,我们展开 Command1,可以看到在 Command1 下面列出了 Authors 表中的各个具体表项(分别为 Au_ID, Author 和 Year Born),如图9.6所示。至此为止,我们已经完成了在数据环境设计器中的设计过程。下面,读者将跟随我们一起来到数据视图图中进行下一步的设计。

(2) 在数据视图图中的设计

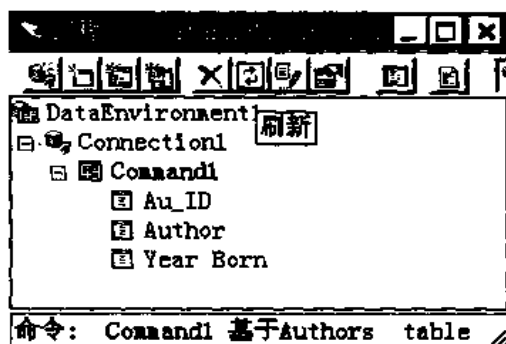


图9.6 展开后的 Command1

要打开“数据视图”窗口,请单击“视图”菜单中的“数据视图”菜单项,或者单击“标准型”工具栏上的 Data View 按钮。

“数据视图”窗口在一个或者多个数据库连接上显示一个视图,提供对一个特定连接的数据库的整体结构的访问。“数据视图”窗口提供使用 Microsoft 可视化数据库工具(查询设计器和数据库设计器)的方法,来可视地操作数据库的结构。在数据视图中包括数据链接(DataLink)和数据环境连接两个部分,关于数据环境的连接我们在上面已经做了详细的介绍,接下来我们要讲述的是数据链接部分。事实上,“数据视图”窗口监视所有添加到您工程中的数据环境,并且显示从这些数据环境的连接。

我们选中数据链接并单击鼠标右键,在弹出菜单中选择“添加数据链接”菜单项,或者可以直接单击“数据视图”窗口中的“添加新的数据链接”来创建一个新的数据链接。在创建新的数据链接时,会弹出如图9.4所示的窗口。按上面的设置数据链接属性的方法来完成新数据链接的属性设置。在完成之后,读者就会看到在数据链接的下面已经添加了一个名为 DataLink1 的数据链接。双击 DataLink1,就可以看到这个新的数据链接下面包括表、视图和存储过程三个部分,它们分别对应着新数据链接所对应的数据源下的表、视图和存储过程,如图9.7所示。在本例子中,数据源为 Biblio 数据库,它只包括了表和视图,并不存在着存储过程,读者如果试图展开存储过程就会看到在存储过程的下一级无任何对象。

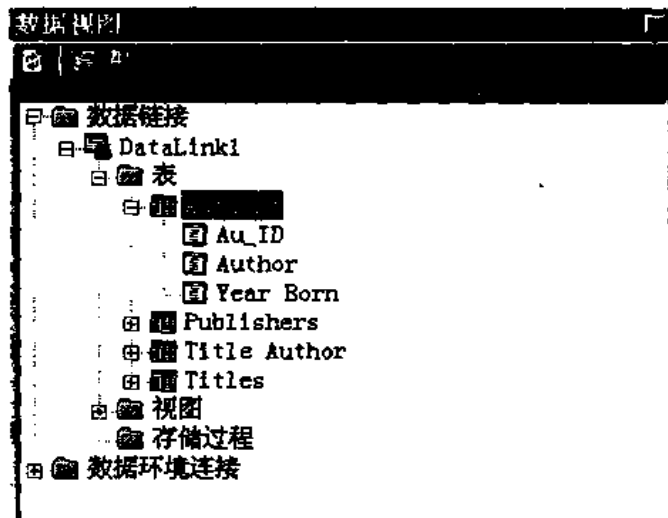


图9.7 数据视图窗口

下面,我们来展开 DataLink1下的表部分。双击 DataLink1下面的表部分或者单击表部分左边的加号就可以展开表。展开以后,读者所看到的是 Biblio 数据库中所包含的全部的表, Authors、Publishers、Title Author 和 Titles。在这里,我们所要用到的只是 Authors 表,其它的三个表只是作相应的显示而并不真正地起作用。选中表部分并单击鼠标右键,在弹出菜单中选择“显示系统对象”菜单项,我们就可以在表部分的下面看到一些系统表,这些系统表包括 MSysACEs、MSysObjects、MSysQueries、MSysRelationships。如果要取消显示系统表,可以在弹出菜单中再选择“显示系统对象”菜单项。

按展开表的方法来展开 Authors 表,可以看到 Authors 表的下面包括三个表项,它们分别是 Au_ID、Author 和 Year Born。从形式上来看,表项已经到达了数据视图的最后一级,读者已无法再在数据视图窗口中来展开表项了。但是,如果读者双击某一具体的表项,就会弹出该表项的具体属性窗口。这个属性窗口包括了该表项的一些基本的属性,例如 Datatype、Length、Name 和 ADODatatype 等。

值得注意的是,读者只可以对这些表项的具体属性做读操作,而不能改变某一个具体属性的值(写操作)。现在,我们在 Authors 表的 Au_ID 表项上进行双击,从而打开 Au_ID 表项的属性窗口,如图9.8所示。

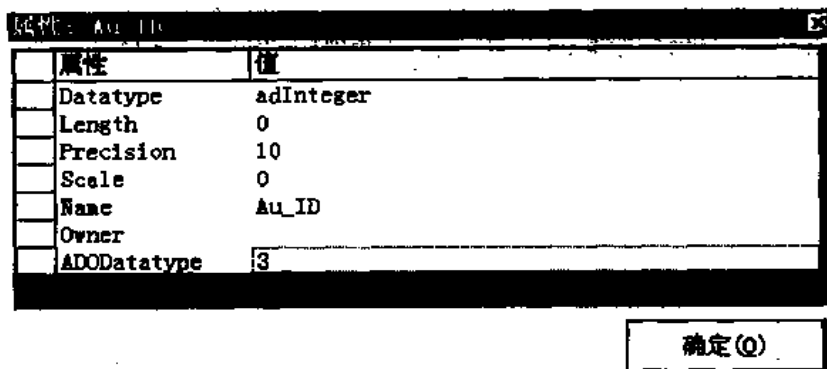


图9.8 表项的属性窗口

(3) 添加和设计 Data Report 控件

在完成数据环境设计器的设计和数据视图的设计后,我们就要进入本实例的核心部分,即添加和设计一个 Data Report 控件。Data Report 控件是用来显示和操作具体报表的控件,我们可以在图9.1中为应用程序添加一个 Data Report 控件。

Data Report 控件分为5个部分,它们分别是报表标头、页标头、细节、页注脚和报表注脚,如图9.9所示。现在,我们来分别讨论这5个部分。

- 报表标头: 可以在报表标头中加入一个 RptLabel 控件来显示整个报表的头部。
- 页表头: 可以在页标头中加入 RptLabel 控件来显示报表数据项的文字说明。
- 细节: 应该在细节带中加入 RptTextBox 控件报表的具体数据项。
- 页注脚: 可以在页注脚中加入 RptLabel 控件来显示每一页的注脚。
- 报表注脚: 可以在报表注脚中加入 RptFunction 控件来对报表的具体数据项做一个统计工作。

在本实例中,我们要改变 Data Report 控件的一些具体的属性值,从而使 Data Report 控

件能够真正地按我们所设想的来显示数据库中对应表的数据。

- Caption 属性：我们把 Data Report 控件的 Caption 属性改为“显示 Authors 表的报表”。
- DataSource 属性：我们把 Data Report 控件的 DataSource 属性设置为 DataEnvironment1。
- DataMember 属性：我们把 Data Report 控件的 DataMember 属性设置为 Command1。

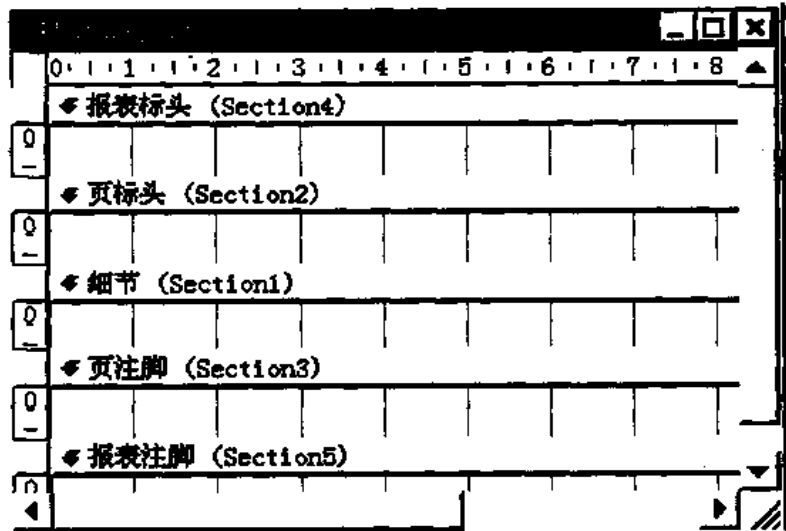


图9.9 Data Report 控件

在添加了一个 Data Report 控件之后,下面我们就要按照本应用的具体要求来设计这个控件。首先,我们在“报表标头”带上放置一个 RptLabel 控件,改变和设置 RptLabel 控件的如下属性值:

- Alignment 属性: Alignment 属性指出控件中文本的对齐或调整方式。Alignment 属性中共有三种文本对齐的方式,它们分别是 rptJustifyLeft;左对齐;rptJustifyRight;右对齐;rptJustifyCenter;文本居中。在这里,我们选择 rptJustifyCenter,即使 RptLabel 控件中的文本按居中的方式来显示。
- Caption 属性: 改变 RptLabel 控件中的 Caption 属性为“Authors 表中的数据”。
- Font 属性: Font 属性设置文本的显示方式和大小,在这里我们在“字体”中选择“楷体_GB2312”;在字体“样式”中选择“Bold”;在“大小”中选择“小三”。

接着,我们在页标头带上放置一个 RptLabel 控件,改变和设置 RptLabel 控件的如下属性:

- Alignment 属性: 在这里,我们设置 Alignment 属性为 rptJustifyCenter。
- Caption 属性: 在这里,我们把 Caption 属性改为“ID”。
- Font 属性: 在这里,我们在“字体”中选择“宋体”;在字体“样式”中选择“Bold”;在“大小”中选择“五号”。

类似地,我们在页标头带上再放置两个 RptLabel 控件,它们的 Caption 属性分别为 Author 和 Year Born。其它的属性与第一个 RptLabel 控件相同。

接下来,我们在细节带中放置一个 RptTextBox 控件。改变 RptTextBox 控件的如下属性

值:

- Alignment 属性: 我们设置 Alignment 属性为 rptJustifyCenter。
- DataMember 属性: 在这里,我们设置 RptTextBox 控件的 DataMember 属性为 Command1。
- DataField 属性: 在这里,我们设置 RptTextBox 控件的 DataField 属性为 Au_ID。它表示这个 RptTextBox 控件显示的是 Authors 表中的 Au_ID 表项数据。

类似地,我们在细节带上再放置两个 RptTextBox 控件,它们的 DataField 属性分别为 Author 和 Year Born。其它的属性与第一个 RptTextBox 控件相同。完成之后的 Data Report 控件如图9.10所示。

Authors表中的数据													
ID	Author	Year	Born										
Au_ID	Author [Command1]	Year	Born										

图9.10 完成后的 Data Report 控件

最后,读者需要注意的是,如果现在运行本实例,并不会显示数据报表,这是因为在 Visual Basic 中有一个启动对象的概念。启动对象就是当 Visual Basic 的应用程序运行时,哪一个对象作为启动而最先运行。这里,我们要在“工程资源管理器”中把启动对象改为 Data Report1。那么,当应用程序一开始运行时,我们就能见到数据报表了。

9.2 扩展数据报表

9.2.1 向数据报表添加 Function 控件

数据报表设计器以它自己的一组控件为特色。在这些控件中,Function 控件在 Visual Basic 内部的控件中没有相对应的控件,需要更详细的解释。Function 控件使用一个内置的函数,在报表生成时显示运行时计算的数据。向数据报表设计器添加一个 Function 控件的步骤如下:

- (1) 在数据报表设计器的一个适当的注脚部分绘制一个 Function 控件。
- (2) 设置 DataMember 和 DataField 属性为适当的值。

在这一小节中,我们要在9.1.2节的实例基础上添加 Function 控件。打开9.1.2节的实例,在 Data Report 控件的报表注脚带中添加一个 RptFunction 控件。改变和设置 RptFunction 控

件的如下属性:

- Alignment 属性: 设置 RptFunction 控件的 Alignment 属性为 rptJustifyCenter。
- DataMember 属性: 在这里,我们设置 RptFunction 控件的 DataMember 属性为 Command1。
- DataField 属性: 在这里,我们设置 RptFunction 控件的 DataField 属性为 Year Born。
- FunctionType 属性: FunctionType 属性设置用于计算时此函数控件显示的值的函数,它共有8种选择,具体的含义见表9.1。在这里,我们设置 FunctionType 属性为 Min,它表示计算 Year Born 表项的最小值。

表9.1 Function 控件的具体函数

函 数	对函数的描述
Sum	合计一个字段的值
Min	显示一个字段的的最小值
Max	显示一个字段的的最大值
Average	显示一个字段的平均值
Standard Deviation	显示一系列数字的标准偏差
Standard Error	显示一系列数字的标准错误
Value Count	显示包含非空值的字段数
Row Count	显示一个报表部分中的行数

在完成了属性设置后,我们就可以在应用程序运行界面的最后一行中见到 Year Born 表项的最小值。完成后的界面如图9.11所示。

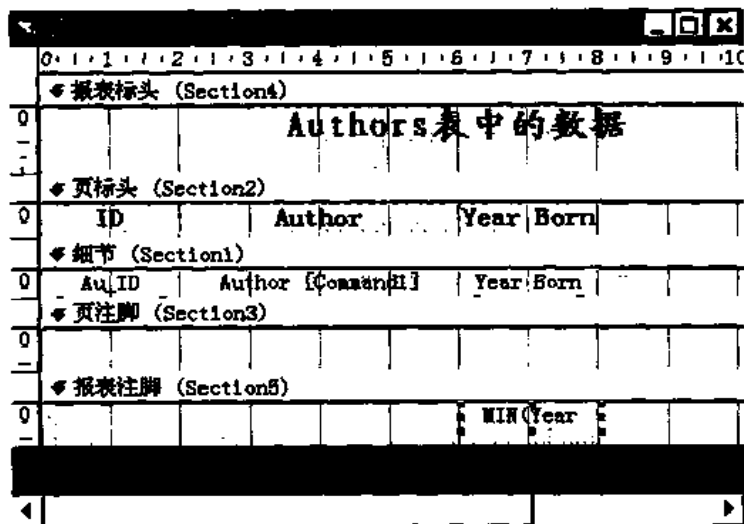


图9.11 添加了 Function 控件后的界面

最后,我们在 Data Report 控件中再加入两个 RptFunction 控件,如图9.12所示。两个 RptFunction 控件的 FunctionType 属性均设置为 RCNT (Row Count),一个控件绑定 Authors 表的 Au_ID 表项,另一个控件绑定 Authors 表的 Author 表项。由于 RCNT 表示的是绑定对象(在这里被绑定的对象为 Author 表中的表项)的行数,所以当应用程序运行时这两项

值应该相同。运行后的程序如图9.13所示。

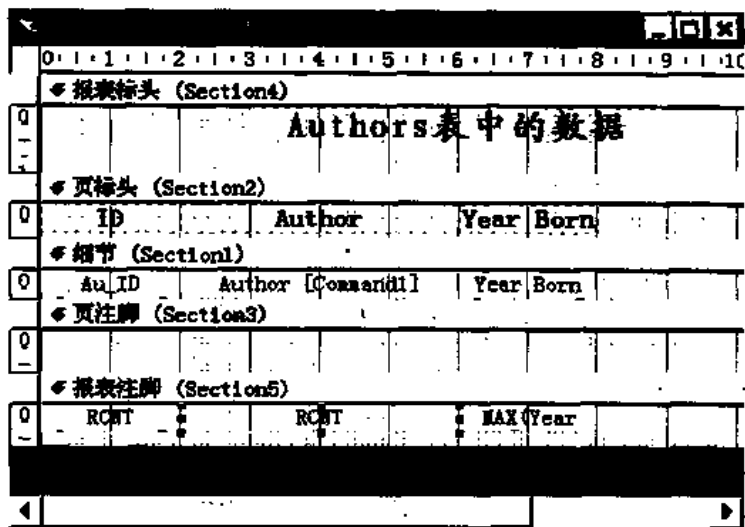


图9.12 添加了 RCNT 后的界面

下面,我们来看图9.13,即程序运行后的状况。我们把报表的页面翻到最后一页,可以看到在 ID 项和 Author 项的最后有两个相同的值。这表示从 Au_ID 表项中计算的行数是从 Au-thor 表项中计算的行数是相同的,事实上它们也应该相同。读者可以自己想象一下,如果这两项不相同是代表什么意思。

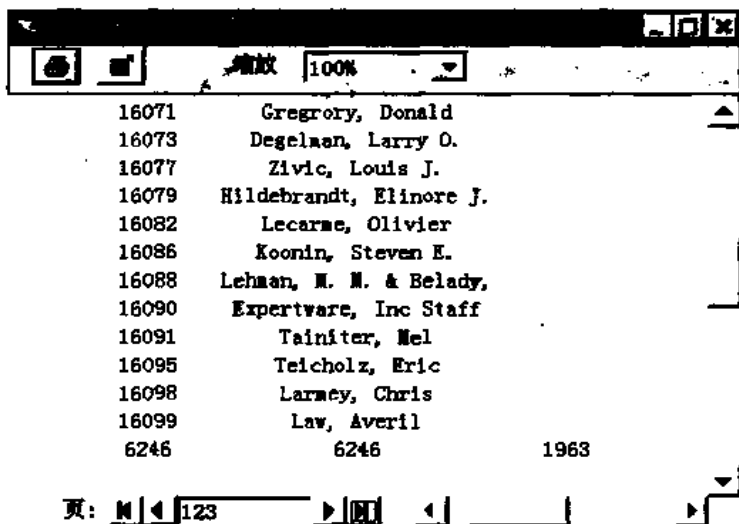
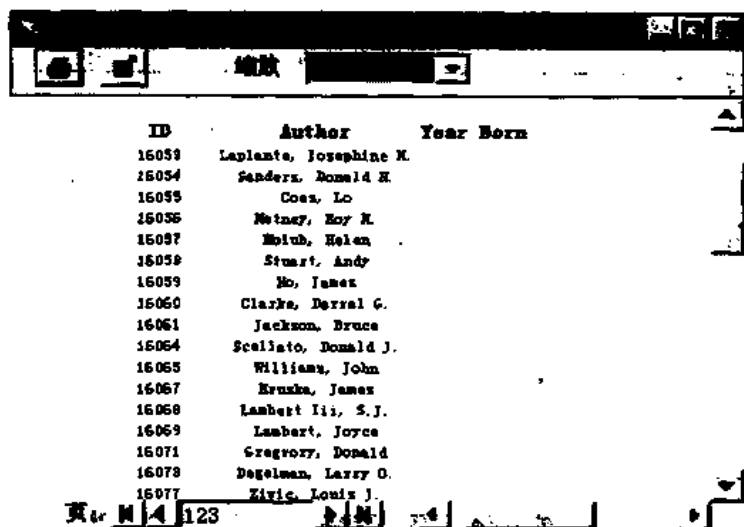


图9.13 添加 RCNT 后的运行界面

现在,我们来介绍一下运行后的界面。在运行界面的左上角有两个按钮,左边的一个为打印按钮,用户可以单击此按钮来打印报表。右边的一个为导出按钮,用户可以单击此按钮来把报表中的内容导出并存储为一个 Html 的文件。在按钮右边有一个缩放的功能,它表示报表显示的具体大小。在这里,我们可以看到缩放的值为100%,它表示报表是按照正常的大小来显示给用户的。我们可以选取缩放功能中的其它值来按不同的大小把报表中的内容显示给用户,图

9.14就是以正常大小的75%来显示报表的。当以不同的大小来显示报表时,一页中能够显示的行数是相同的,这一点要请读者注意。细心的读者可能已经注意到了在右下角的控件非常熟悉,其实在左下角“页”右边的控件为一个 ADO Data 控件,它同在 Visual Basic 开发环境中的 ADO Data 控件一样有向左、向右、置首和置尾功能。



ID	Author	Year Born
16033	Laplante, Josephine M.	
16034	Sanders, Donald H.	
16035	Coss, Lo	
16036	Metzger, Roy M.	
16037	Molub, Helen	
16038	Stuart, Andy	
16039	Mo, James	
16060	Clarke, Derral G.	
16061	Jackson, Bruce	
16064	Scelinto, Donald J.	
16065	Williams, John	
16067	Kruska, James	
16068	Lambert III, S.J.	
16069	Lambert, Joyce	
16071	Gregory, Donald	
16078	Degelman, Larry O.	
16077	Zivic, Louis J.	

图9.14 按75%来显示报表

9.2.2 往数据报表中添加分组

上面的一些例子告诉了用户如何通过数据报表来显示数据,但是这种显示是用一种非预处理的方式。有些时候,为了直观起见,用户希望数据是以成组的形式在数据报表中显示的。成组的信息可以让用户从另一个角度看待数据。和数据环境设计器串联使用,数据报表设计器提供根据表中任何字段对数据分组的能力。

在数据环境中的 Command 对象提供的分组字段,不同于已经由创建分组标头和注脚获得的分组字段。不用将数据库的表作为分组基础,数据环境设计器的 Grouping 功能允许选择表中的一个特定的字段作为分组字段,而不必创建一个新的 Command 对象。这对于用户来说无疑是一件简便的事。

下面,我们要往数据报表中添加一个分组,具体的步骤如下:

(1) 首先,在数据环境设计器中创建一个分组字段。这个创建要在数据环境设计器中的 Command 对象的属性窗口的“分组”选项卡中进行。在“分组”选项卡中,用户可以根据具体的需要来创建一个或者多个分组。

(2) 向数据报表设计器添加一个分组标头/注脚,对应于新的分组命令。这时,用户只需在数据报表的弹出菜单中选择“插入分组标头/注脚”,这样我们就会在数据报表中见到一个分组标头带和一个分组注脚带。

(3) 重新将数据报表的 DataMember 属性设置为在数据环境中创建的新的 Grouping Command 对象。

(4) 把分组字段从数据环境拖到数据报表。

在了解了往数据报表中添加分组的步骤后,我们就根据这些步骤和上面一个实例中的方

9.2.3 向数据报表中添加页标题

有时候数据报表中会有很多页,按照一般报表的形式我们需要在这些页当中加入页标头和页注脚。可以通过向页标头或页注脚部分添加适当的 Label 控件来将标头和注脚数据添加到数据报表的每一页。数据报表设计器以下面的标头和注脚部分为特色。

(1) 报表标头

报表标头只出现一次,就在报表的顶部。可以用来介绍材料,如报表的目的等。

(2) 报表注脚

报表注脚像报表标头一样,只出现一次,就在报表的最底部。报表注脚可以用来总结材料,例如结论,或报表的突出部分,或使用 Function 控件的总计等。

(3) 页标头

页标头在每一页的顶部出现。使用页标头可以用作报表的页号、日期和时间,或报表的标题等。

(4) 页注脚

页注脚在每一页的底部出现。使用页注脚也可以用作报表的页号、日期和时间,或附带信息等。

数据报表设计器包含几个预先配置的控件,可以用来快速地向报表的任何部分添加日期、时间、页号或报表标题。这些控件包含显示如表9.2所示的信息代码的 Label 控件。

表9.2 控件的函数和代码

函 数	代 码
Current Page Number	%p
Total Number of Pages	%P
Current Date (Short Format)	%d
Current Date (Long Format)	%D
Current Time (Short Format)	%t
Current Time (Long Format)	%T
Report Title	%i

注意,在上面的表中,代码都是以%来开头的。如果用户要显示百分号,那么就要使用%%来表示。在一个数据报表的设计过程中,若要向页标头添加标题和页号,请按照下列步骤执行:

(1) 单击数据报表设计器的标题栏选中它。

(2) 把焦点移到属性窗口上,单击“标题”项并键入文本。这些键入的文本是将显示在页标头中的标题。

(3) 右键单击数据报表设计器,并单击“显示页标头/页脚”。

(4) 右键单击页标头部分,并单击“插入”。

(5) 从菜单中单击“报表标题”。

(6) 在页标头部分的中间重新定位新的控件,使它伸展到足够宽,以显示整个报表标题。

(7) 重复步骤4。

(8) 在菜单中单击“当前页码”。

(9) 将新控件定位在设计器的右角。

(10) 设置新控件的 Alignment 属性为1—Justify Right,它表示了具体的对齐方式。

当然,在上面的10个步骤中并不是每个步骤都是必需和固定不变的,用户可以根据具体要求的不同来更改其中的某些步骤。接下来,我们要根据上述的步骤来具体地看一个实例。这个实例是在9.2.2节实例的基础上进行的,完成后的数据报表界面如图9.17所示。

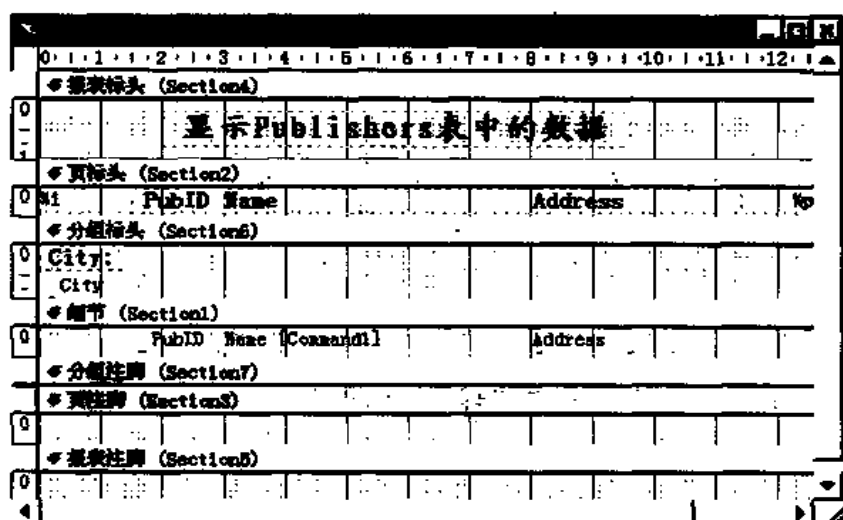


图9.17 添加了页标头的数据报表

在这里,对应于步骤2,在 Caption 属性中输入的文本为“添加页标题”。由于页脚注带的设计方式与页标头带相似,在这里我们就不再重复了,有兴趣的读者可以往数据报表中添加。报表运行后如图9.18所示。

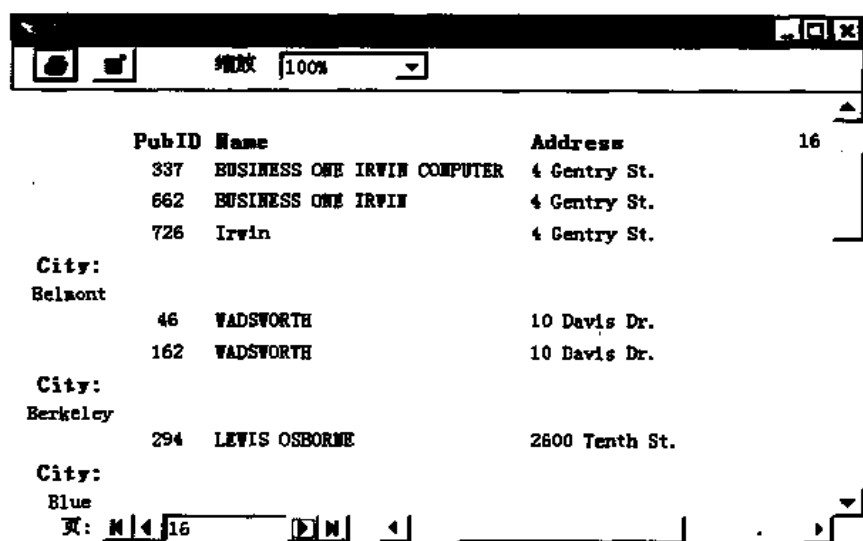


图9.18 运行后的显示界面

读者可以看到,在报表的右上角有一个数字,该数字代表的含义是当前页的页码。

9.2.4 Data Report 事件

同其它标准的 Visual Basic 控件一样,Data Report 控件也有自身的事件。在这一小节中我们将简单地介绍一下常用的 Data Report 事件,读者可以根据需要在实际的应用程序开发过程中有选择地使用这些事件。

(1) Initialize 事件

Initialize 事件在查询完成之后发生。

(2) Resize 事件

Resize 事件在数据报表第一次显示或当数据报表的窗口状态发生改变时发生。

(3) Activate 事件

Activate 事件在当设计器变为活动窗口时发生。

(4) ProcessingTimeout 事件

ProcessingTimeout 事件大约每秒钟发生一次,直到所有处理都已经结束。使用该事件可以决定处理是否占用太长时间,并取消处理。在查询完成之前该事件将不发生。

(5) Deactivate 事件

Deactivate 事件是当设计器不再是活动窗口时发生。使用该事件决定用户是否单击了另一个窗体或设计器。

(6) QueryClose 事件

QueryClose 事件在设计器终止之前发生。设置 Cancel 参数为 True 以取消终止。Close-Mode 参数返回引起终止的动作类型。

(7) Terminate 事件

Terminate 事件是当所有对设计器的引用都被设置为0时发生。

至于这些事件的具体实例,由于受篇幅所限,在这里就不再一一给出了。

第十章 关于 ADO 对象的实例

关于 ADO 对象的一些知识我们在前面的章节中已经有了说明。我们知道,ADO 对象包括 Command 对象、Connection 对象、Fields 对象、RecordSet 对象和 Error 对象等。在经过了前面几章节的学习后,相信读者对 ADO 对象的编程已经有了一定的了解。在本章中,我们会较为深入地来讲述关于 ADO 对象编程的一些实例。图10.1给出了 ADO 各对象之间的关系。

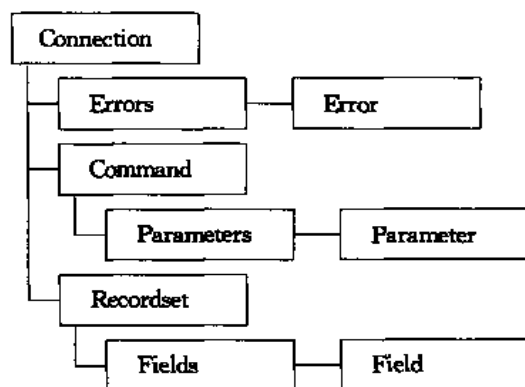


图10.1 ADO 对象的关系

在本章中我们主要讲述以下实例：

- DataControl 对象实例
- DataSpace 对象和 CreateObject 方法实例
- DataFactory 对象实例

10.1 DataControl 对象实例

为了方便起见,在本章中提供的所有实例我们不再像前面的章节一样给出具体的操作步骤,只是提供一些基本的代码,读者可以根据需要自己动手完成有关界面和其它的一些代码书写。

10.1.1 DataControl 对象实例1

以下代码显示如何在设计时设置 RDS.DataControl 参数,以及如何使用称为 ADCDemo 的 ODBC 数据源将其绑定到数据标识的控件当中。如果用户按照指导操作,ADCDemo 将作为 SQL Server ODBC 数据源安装到服务器上。请将其剪切并粘贴到标准 HTML 文档的 <Body></Body> 标记之间,然后将其命名为 ADCap1.asp。ASP 脚本将标识服务器。

```
<Center><H2>RDS API Code Examples</H2>
<HR><BR>
<H3>Remote Data Service</H3>
```

```

<Object CLASSID="clsid:AC05DC80-7DF1-11d0-839E-00A024A94B3A"
  CODEBASE="http://<% = Request.ServerVariables("SERVER_NAME") %>/MSADC/
  Samples/Sheridan.cab"
  ID=GRID1
  datasrc = #ADC
  HEIGHT=125
  WIDTH=495>
  <PARAM NAME="AllowAddNew" VALUE="TRUE">
  <PARAM NAME="AllowDelete" VALUE="TRUE">
  <PARAM NAME="AllowUpdate" VALUE="TRUE">
</OBJECT>
<!-- 设计时设置的具有参数的 Remote Data Service -->
<OBJECT classid="clsid:BD96C556-65A3-11D0-983A-00C04FC29E33"
ID=ADC>
  <PARAM NAME="SQL" VALUE="Select * from Employee for browse"> <PARAM
  NAME="SERVER" VALUE="http://<% = Request.ServerVariables
  ("SERVER_NAME") %>">
  <PARAM NAME="CONNECT" VALUE="dsn=ADCDemo;UID=ADCDemo;
  PWD=ADCDemo;">
</OBJECT><BR><HR></Center>

```

10.1.2 DataControl 对象实例2

以下范例显示如何在运行时设置 RDS.DataControl 的必要参数。要测试该范例,请将其剪切并粘贴到标准 HTML 文档的<Body></Body>标记之间,然后将其命名为 ADCapi2.asp。ASP 脚本将标识服务器。

```

<Center><H2>RDS API Code Examples </H2>
<HR><BR>
<H3>Remote Data Service Run Time</H3>
<Object CLASSID="clsid:AC05DC80-7DF1-11d0-839E-00A024A94B3A"
CODEBASE="http://<% = Request.ServerVariables("SERVER_NAME") %>
/MSADC/Samples/Sheridan.cab"
ID=GRID1
  datasrc = #ADC
  HEIGHT=125
  WIDTH=495>
  <PARAM NAME="AllowAddNew" VALUE="TRUE">
  <PARAM NAME="AllowDelete" VALUE="TRUE">
  <PARAM NAME="AllowUpdate" VALUE="TRUE"> <PARAM NAME="Caption" VAL-
  UE="Remote Data Service Run Time">
</OBJECT>
<!-- 设计时设置的不带参数的 RDS.DataControl -->
<OBJECT classid="clsid:BD96C556-65A3-11D0-983A-00C04FC29E33"
  ID=ADC>
</OBJECT>
<HR>

```



```

<Input Size=70 Name="txtServer" Value="http://<% =Request.ServerVariables
("SERVER_NAME") %>"><BR>
<Input Size=70 Name="txtConnect" Value="dsn=ADCDemo,
UID=ADCDemo,PWD=ADCDemo;">
<BR>
<Input Size=70 Name="txtSQL" Value="Select * from Employee">
<HR>
<INPUT TYPE=BUTTON NAME="Run" VALUE="Run"><BR>
<H4>Fill Grid with these values or change them to see data from another ODBC data source on your
server</H4>
</Center>
<Script Language="VBScript">
<!--
' 在运行时设置 RDS.DataControl 的参数
Sub Run_OnClick
    ADC.Server = txtServer.Value
    ADC.SQL = txtSQL.Value
    ADC.Connect = txtConnect.Value
    ADC.Refresh
End Sub
-->
</Script>

```

10.2 DataSpace 对象和 CreateObject 方法实例

10.2.1 DataSpace 对象和 CreateObject 方法实例1

以下范例显示如何使用具有默认业务对象 RDSServer.DataFactory 的 RDS.DataSpace 的 CreateObject 方法。要测试该范例,请将其剪切并粘贴到标准 HTML 文档的 <Body> </Body> 标记之间,然后将其命名为 ADCapi8.asp。ASP 脚本将标识服务器。

```

<Center><H2>RDS API Code Examples </H2>
<HR><H3>Using Query Method of RDSServer.DataFactory</H3>
<!-- RDS.DataSpace ID ADS1-->
<OBJECT ID="ADS1" WIDTH=1 HEIGHT=1
CLASSID="CLSID:BD96C556-65A3-11D0-983A-00C04FC29E36">
</OBJECT>
<!-- 运行时设置的具有参数的 RDS.DataControl -->
<OBJECT classid="clsid:BD96C556-65A3-11D0-983A-00C04FC29E33"
ID=ADC>
</OBJECT>
<Object CLASSID="clsid:AC05DC80-7DF1-11d0-839E-00A024A94B3A"
CODEBASE="http://<% =Request.ServerVariables("SERVER_NAME") %>/MSADC/Samples/
Sheridan.cab" ID=GRID1
    datasrc=#ADC

```

```

HEIGHT=125
WIDTH=495>
<PARAM NAME="AllowAddNew" VALUE="TRUE">
<PARAM NAME="AllowDelete" VALUE="TRUE">
<PARAM NAME="AllowUpdate" VALUE="TRUE">
<PARAM NAME="Caption" VALUE="RDSServer. DataFactory Run Time">
</OBJECT>
<HR>
<INPUT TYPE=BUTTON NAME="Run" VALUE="Run"><BR>
<H4>Click Run. The CreateObject Method of the RDS. DataSpace Object Creates
an instance of the RDSServer. DataFactory. The Query Method of the
RDSServer. DataFactory is used to bring back a Recordset. </H4></Center>
<Script Language="VBScript">
<!--
Dim ADFDim strServer
Dim strConnect
Dim strSQLstrServer = "http://<%=Request.ServerVariables("SERVER_NAME")%>"
strConnect = "dsn=ADCDemo;UID=ADCDemo;PWD=ADCDemo;"strSQL =
"Select * from Employee"Sub Run_OnClick()
Dim objADORS      ' 创建 Recordset 对象
Set ADF = ADS1.CreateObject("RDSServer. DataFactory", strServer) ' 获取 Recordset
Set objADORS = ADF.Query(strConnect, strSQL)
' 使用 RDS. DataControl 将 Recordset 绑定到网格敏感控件
ADC.SourceRecordset = objADORS
End Sub-->
</Script>

```

10.2.2 DataSpace 对象和 CreateObject 方法实例2

以下范例显示如何使用 CreateObject 方法创建自定义业务对象 VbBusObj. VbBusObjCls 的实例。该范例还使用 Active Server Pages 脚本标识 Web 服务器名。要查看完整的范例,请通过范例应用程序的选择器,在“客户端层”列中选择“VBScript in Internet Explorer”,并在“中间层”列中选择“自定义 Visual Basic 业务对象”。

```

Sub Window_OnLoad( )
    strServer = "http://<%=Request.ServerVariables("SERVER_NAME")%>"
    Set BO = ADS1.CreateObject("VbBusObj. VbBusObjCls", strServer)
    txtConnect.Value = "dsn=pubs;uid=sa;pwd="
    txtGetRecordset.Value = "Select * From authors for Browse"
End Sub

```

10.3 DataFactory 对象实例

该范例使用 RDS. DataSpace 对象的 CreateObject 方法创建 RDSServer. DataFactory 对象。要测试该范例,请将该代码剪切并粘贴到标准 HTML 文档的<Body></Body>标记之

间,并命名为 ADCapi7.asp。ASP 脚本将标识您的服务器。

```

<Center><H2>RDS API Code Examples</H2>
<HR><H3>Using Query Method of RDSServer. DataFactory</H3>
<!-- RDS. DataSpace ID ADS1-->
<OBJECT ID="ADS1" WIDTH=1 HEIGHT=1CLASSID="
CLSID:BD96C556-65A3-11D0-983A-00C04FC29E36">
</OBJECT>
<!-- 运行时设置的具有参数的 RDS. DataControl -->
<OBJECT classid="clsid:BD96C556-65A3-11D0-983A-00C04FC29E33"
ID=ADC>
</OBJECT>
<Object classid ="clsid:AC05DC80-7DF1-11d0-839E-00A024A94B3A"
CODEBASE="http://<%=Request.ServerVariables _
("SERVER_NAME")%>/MSADC/Samples/Sheridan.cab"
ID=GRID1
    datasrc= # ADC
    HEIGHT=125
    WIDTH=495>
<PARAM NAME="AllowAddNew" VALUE="TRUE">
<PARAM NAME="AllowDelete" VALUE="TRUE">
<PARAM NAME="AllowUpdate" VALUE="TRUE">
<PARAM NAME="Caption" VALUE=" RDSServer. DataFactory Run Time">
</OBJECT>
<HR>
<INPUT TYPE=BUTTON NAME="Run" VALUE="Run"><BR>
<H4>Click Run. The CreateObject Method of theRDS. DataSpace
Object Creates an instance of theRDSServer. DataFactory. The Query
Method of the RDSServer. DataFactory is usedto bring back a Recordset. </H4>
</Center>
<Script Language="VBScript">
<!--
Dim ADF
Dim strServer
Dim strConnect
Dim strSQLstrServer = "http://<%=Request.ServerVariables _("SERVER_NAME")%>"
strConnect = "dsn=ADCDemo;UID=ADCDemo;PWD=ADCDemo;"strSQL =
"Select * from Employee"
Sub Run_OnClick( )
' 创建 RDSServer. DataFactory 对象
    Dim objADOrs
' 获得记录集
    Set ADF = ADS1.CreateObject("RDSServer. DataFactory", strServer)
    Set objADOrs = ADF.Query(strConnect, strSQL)
' 设置 RDS. DataControl 运行时参数
    ADC.Server = strServer
    ADC.SQL = strSQL

```

```
    ADC.Connect = strConnect
    ADC.Refresh
End Sub
-->
</Script>
```

第十一章 关于 ADO 属性的实例

在这一章中,我们要讲述关于 ADO 属性的具体实例。从理论上来说,ADO 属性分为静态属性和动态属性两类,这两类属性都为 ADO 对象的内部属性。虽然读者可以根据自己的需要来给 ADO 对象添加用户属性,但是本章中所讲述的均为 ADO 对象的内部属性。在本章中,我们假设已有一个记录雇员信息的雇员表。

在本章中我们主要讲述以下实例:

- AbsolutePage、PageCount 和 PageSize 属性实例
- AbsolutePosition 和 CursorLocation 属性实例
- ActiveConnection 等属性实例
- ActualSize 和 DefinedSize 属性实例
- Attributes 和 Name 属性实例
- BOF、EOF 和 Bookmark 属性实例
- CacheSize 属性实例
- Connect 属性实例
- ConnectionString 等属性实例
- Count 属性实例
- CursorType 等属性实例
- Description 等属性实例
- ExecuteOptions 和 FetchOptions 属性实例
- Filter 和 RecordCount 属性实例
- FilterColumn 等属性实例
- IsolationLevel 和 Mode 属性实例
- MarshalOptions 属性实例
- MaxRecords 属性实例
- NumericScale 和 Precision 属性实例
- OriginalValue 和 UnderlyingValue 属性实例
- Prepared 属性实例
- Provider 和 DefaultDatabase 属性实例
- Recordset 和 SourceRecordset 属性实例
- ReadyState 属性实例
- Server 属性实例
- Source 属性实例
- SQL 属性实例
- State 属性实例
- Status 属性实例
- Type 属性实例

- Value 属性实例
- Version 属性实例

11.1 AbsolutePage、PageCount 和 PageSize 属性实例

AbsolutePage 属性用来指定当前记录所在的页。PageCount 属性用来指示 Recordset 对象包含的数据页数。使用 PageCount 属性可确定 Recordset 对象中数据的页数。“页”是大小等于 PageSize 属性设置的记录组。即使最后页不完整,由于记录数比 PageSize 值少,该页也会作为 PageCount 值中的附加页进行计数。如果 Recordset 对象不支持该属性,该值为 -1 以表明 PageCount 无法确定。PageSize 属性用来指示 Recordset 中一页所包含的记录数。使用 PageSize 属性可确定组成逻辑数据页的记录数。建立页的大小允许使用 AbsolutePage 属性移动到特定页的第一个记录。在希望允许用户对数据进行分页时,该属性在 Web 服务器方案中非常有用,可用来在某一时刻查看一定数量的记录。

本实例使用 AbsolutePage、PageCount 和 PageSize 属性,以每次五个记录的方式显示雇员表中的姓名和受雇日期。

```
Public Sub AbsolutePageX()
    Dim rstEmployees As ADODB.Recordset
    Dim strCnn As String
    Dim strMessage As String
    Dim intPage As Integer
    Dim intPageCount As Integer
    Dim intRecord As Integer
    ' 使用客户端游标为雇员表打开一个记录集。
    strCnn = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    Set rstEmployees = New ADODB.Recordset
    ' 使用客户端游标激活 AbsolutePosition 属性。
    rstEmployees.CursorLocation = adUseClient
    rstEmployees.Open "employee", strCnn, , , adCmdTable
    ' 显示姓名和受雇日期,每次五个记录。
    rstEmployees.PageSize = 5
    intPageCount = rstEmployees.PageCount
    For intPage = 1 To intPageCount
        rstEmployees.AbsolutePage = intPage
        strMessage = ""
        For intRecord = 1 To rstEmployees.PageSize
            strMessage = strMessage & _
                rstEmployees!fname & " " & _
                rstEmployees!lname & " " & _
                rstEmployees!hire-date & vbCrLf
        Next intRecord
        rstEmployees.MoveNext
    If rstEmployees.EOF Then Exit For
    Next intPage
End Sub
```

```

        MsgBox strMessage
    Next intPage
    rstEmployees.Close
End Sub

```

11.2 AbsolutePosition 和 CursorLocation 属性实例

AbsolutePosition 属性用来指定 Recordset 对象当前记录的序号位置。使用 AbsolutePosition 属性可根据其在 Recordset 中的序号位置移动到记录,或确定当前记录的序号位置。提供者必须支持该属性的相应功能才能使用该属性。CursorLocation 属性用来设置或返回游标服务的位置,该属性允许在可用于提供者的各种游标库中进行选择。通常,可以选择使用客户端游标库或位于服务器上的某个游标库。

本实例说明 AbsolutePosition 属性如何对枚举所有 Recordset 记录的循环进程进行跟踪。它通过将游标设置为客户端游标,使用 CursorLocation 属性激活 AbsolutePosition 属性。

```

Public Sub AbsolutePositionX( )
    Dim rstEmployees As ADODB.Recordset
    Dim strCnn As String
    Dim strMessage As String
    ' 使用客户端游标为雇员表打开一个记录集。
    strCnn = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    Set rstEmployees = New ADODB.Recordset
    ' 使用客户端游标激活 AbsolutePosition 属性。
    rstEmployees.CursorLocation = adUseClient
    rstEmployees.Open "employee", strCnn, , , adCmdTable
    ' 枚举记录集。
    Do While Not rstEmployees.EOF
        ' 显示当前记录信息。
        strMessage = "Employee: " & rstEmployees!Name & vbCr & _
            "(record " & rstEmployees.AbsolutePosition & _
            " of " & rstEmployees.RecordCount & ")"
        If MsgBox(strMessage, vbOKCancel) = vbCancel Then
            Exit Do
        End If
        rstEmployees.MoveNext
    Loop
    rstEmployees.Close
End Sub

```

11.3 ActiveConnection 等属性实例

本实例使用 ActiveConnection、CommandText、CommandTimeout、CommandType、Size 和 Direction 属性执行存储过程。

```

Public Sub ActiveConnectionX( )
    Dim cnn1 As ADODB.Connection
    Dim cmdByRoyalty As ADODB.Command
    Dim prmByRoyalty As ADODB.Parameter
    Dim rstByRoyalty As ADODB.Recordset
    Dim rstAuthors As ADODB.Recordset
    Dim intRoyalty As Integer
    Dim strAuthorID As String
    Dim strCnn As String
    ' 定义存储过程的命令对象。
    Set cnn1 = New ADODB.Connection
    strCnn = "Provider=sqloledb; & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=; "
    cnn1.Open strCnn
    Set cmdByRoyalty = New ADODB.Command
    Set cmdByRoyalty.ActiveConnection = cnn1
    cmdByRoyalty.CommandText = "byroyalty"
    cmdByRoyalty.CommandType = adCmdStoredProc
    cmdByRoyalty.CommandTimeout = 15
    ' 定义存储过程的输入参数。
    intRoyalty = Trim(InputBox("Enter royalty;"))
    Set prmByRoyalty = New ADODB.Parameter
    prmByRoyalty.Type = adInteger
    prmByRoyalty.Size = 3
    prmByRoyalty.Direction = adParamInput
    prmByRoyalty.Value = intRoyalty
    cmdByRoyalty.Parameters.Append prmByRoyalty
    ' 通过执行该命令创建记录集。
    Set rstByRoyalty = cmdByRoyalty.Execute( )
    ' 打开作者表以便显示作者姓名。
    Set rstAuthors = New ADODB.Recordset
    rstAuthors.Open "authors",strCnn,,,adCmdTable
    ' 打印记录集中的当前数据,从作者表中添加作者姓名。
    Debug.Print "Authors with " & intRoyalty & " percent royalty"
    Do While Not rstByRoyalty.EOF
        strAuthorID = rstByRoyalty!au_id
        Debug.Print,rstByRoyalty!au_id & ", ";
        rstAuthors.Filter = "au_id = " & strAuthorID & ""
        Debug.Print rstAuthors!au_fname & " " & _
            rstAuthors!au_lname
        rstByRoyalty.MoveNext
    Loop
    rstByRoyalty.Close
    rstAuthors.Close
    cnn1.Close
End Sub

```


11.4 ActualSize 和 DefinedSize 属性实例

ActualSize 属性指示字段的值的实际长度。使用 ActualSize 属性可返回 Field 对象值的实际长度。对于所有字段, ActualSize 属性为只读。如果 ADO 无法确定 Field 对象值的实际长度, ActualSize 属性将返回 adUnknown。DefinedSize 属性用来指示 Field 对象所定义的大小。DefinedSize 属性与 ActualSize 属性不同。例如, 假设有一个 Field 对象, 其声明类型为 adVarChar, DefinedSize 属性值为 50, 并包含单个字符。该对象返回的 ActualSize 属性值为单个字符的字节长度。在下面的实例中我们使用 ActualSize 和 DefinedSize 属性显示字段的定义大小和实际大小。

```
Public Sub ActualSizeX( )
    Dim rstStores As ADODB.Recordset
    Dim strCnn As String
    ' 打开存储表的记录集。
    strCnn = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password="
    Set rstStores = New ADODB.Recordset
    rstStores.Open "stores", strCnn, , adCmdTable
    ' 在显示 stor_name 字段内容、字段定义大小和实际大小的记录集中进行循环。
    rstStores.MoveFirst
    Do Until rstStores.EOF
        MsgBox "Store name: " & rstStores(stor_name) & _
            vbCr & "Defined size: " & _
            rstStores(stor_name).DefinedSize & vbCr & "Actual size: " & _
            rstStores(stor_name).ActualSize & vbCr
        rstStores.MoveNext
    Loop
    rstStores.Close
End Sub
```

11.5 Attributes 和 Name 属性实例

Attributes 属性用来指示对象的一项或多项特性。使用 Attributes 属性可设置或返回 Connection 对象、Paramete 对象、Field 对象或 Property 对象的特性。设置多个属性时, 可以将相应的常量相加。如果将属性值设置为包括不兼容常量的总和, 那么将产生错误。Name 属性用来指示对象的名称。使用 Name 属性可将名称赋给 Command、Field、Parameter 或 Property 对象, 或检索它们的名称。

本实例显示 Connection、Field 和 Property 对象的 Attributes 属性值。它使用 Name 属性显示每个 Field 和 Property 对象的名称。

```
Public Sub AttributesX
    Dim cnn1 As ADODB.Connection
```

```

Dim rstEmployees As ADODB.Recordset
Dim fldLoop As ADODB.Field
Dim proLoop As ADODB.Property
Dim strCnn As String
' 打开连接和记录集。
strCnn = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
Set cnn1 = New ADODB.Connection
cnn1.Open strCnn
Set rstEmployees = New ADODB.Recordset
rstEmployees.Open "employee",cnn1,,adCmdTable
' 显示连接的属性。
Debug.Print "Connection attributes = " & cnn1.Attributes
' 显示雇员表字段的属性。
Debug.Print "Field attributes:" For Each fldLoop In rstEmployees.Fields
    Debug.Print "    " & fldLoop.Name & " = " & _
        fldLoop.Attributes
Next fldLoop
' 显示是 NULLABLE 的雇员表的字段。
Debug.Print "NULLABLE Fields:"
For Each fldLoop In rstEmployees.Fields
    If CBool(fldLoop.Attributes And adFldIsNullable) Then
        Debug.Print "    " & fldLoop.Name
    End If
Next fldLoop
' 显示雇员表属性的属性。
Debug.Print "Property attributes:"
For Each proLoop In rstEmployees.Properties
    Debug.Print "    " & proLoop.Name & " = " & _
        proLoop.Attributes
Next proLoop
rstEmployees.Close
cnn1.Close
End Sub

```

11.6 BOF、EOF 和 Bookmark 属性实例

BOF 指示当前记录位置位于 Recordset 对象的第一个记录之前。EOF 指示当前记录位置位于 Recordset 对象的最后一个记录之后。Bookmark 属性返回唯一标识 Recordset 对象中当前记录的书签,或者将 Recordset 对象的当前记录设置为由有效书签所标识的记录。使用 Bookmark 属性可保存当前记录的位置并随时返回到该记录。书签只能在支持书签功能的 Recordset 对象中使用。

11.6.1 BOF、EOF 和 Bookmark 属性实例1

在该实例中我们使用了 BOF 和 EOF 属性,在用户试图移过 Recordset 的第一个和最后

一个记录时显示一条信息。它通过 Bookmark 属性使用户对 Recordset 中的记录进行标记,稍后再返回给它。

```
Public Sub BOFX()
    Dim rstPublishers As ADODB.Recordset
    Dim strCnn As String
    Dim strMessage As String
    Dim intCommand As Integer
    Dim varBookmark As Variant
    ' 使用来自出版商表的数据打开记录集。
    strCnn = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    Set rstPublishers = New ADODB.Recordset
    rstPublishers.CursorType = adOpenStatic
    ' 使用客户端游标启用 AbsolutePosition 属性。
    rstPublishers.CursorLocation = adUseClient
    rstPublishers.Open "SELECT pub_id, pub_name FROM publishers " & _
        "ORDER BY pub_name", strCnn, , , adCmdText
    rstPublishers.MoveFirst
    Do While True
        ' 显示关于当前记录的信息并让用户输入。
        strMessage = "Publisher, " & rstPublishers!pub_name & _
            vbCr & "(record " & rstPublishers.AbsolutePosition & _
            " of " & rstPublishers.RecordCount & ")" & vbCr & vbCr & _
            "Enter command;" & vbCr & _
            "[1 - next / 2 - previous /" & vbCr & _
            "3 - set bookmark / 4 - go to bookmark]"
        intCommand = Val(InputBox(strMessage))
        Select Case intCommand
            ' 向前或向后移动,捕获 BOF 或 EOF。
            Case 1
                rstPublishers.MoveNext
                If rstPublishers.EOF Then
                    MsgBox "Moving past the last record." & _
                        vbCr & "Try again."
                    rstPublishers.MoveLast
                End If
            Case 2
                rstPublishers.MovePrevious
                If rstPublishers.BOF Then
                    MsgBox "Moving past the first record." & _
                        vbCr & "Try again."
                    rstPublishers.MoveFirst
                End If
                ' 保存当前记录的书签。
            Case 3
                varBookmark = rstPublishers.Bookmark
                ' 转到由存储的书签所指示的记录。
            Case 4
```

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors",adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors",adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors",adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。


```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```

        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub

```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```

Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub

```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

```
        If IsEmpty(varBookmark) Then
            MsgBox "No Bookmark set!"
        Else
            rstPublishers.Bookmark = varBookmark
        End If
    Case Else
        Exit Do
    End Select
Loop
rstPublishers.Close
End Sub
```

11.6.2 BOF、EOF 和 Bookmark 属性实例2

在本实例中我们使用 Bookmark 和 Filter 属性创建记录集的限定视图,将只允许访问书签数组所引用的记录。

```
Public Sub BOFX2( )
    Dim rs As New ADODB.Recordset
    Dim bmk(10)
    rs.CursorLocation = adUseClient
    rs.ActiveConnection = "Provider=sqloledb;" & _
        "Data Source=srv;Initial Catalog=pubs;User Id=sa;Password=;"
    rs.Open "select * from authors", ,adOpenStatic,adLockBatchOptimistic
    Debug.Print "Number of records before filtering: ",rs.RecordCount
    ii = 0
    While rs.EOF <> True And ii < 11
        bmk(ii) = rs.Bookmark
        ii = ii + 1
        rs.Move 2
    Wend
    rs.Filter = bmk
    Debug.Print "Number of records after filtering: ",rs.RecordCount
    rs.MoveFirst
    While rs.EOF <> True
        Debug.Print rs.AbsolutePosition, rs("au_lname")
        rs.MoveNext
    Wend
End Sub
```

11.7 CacheSize 属性实例

CacheSize 属性用来指示缓存在本地内存中的 Recordset 对象的记录数。使用 CacheSize 属性可控制提供者缓存在本地内存中所保存的记录数目,并可控制一次恢复到本地内存的记录数。例如,如果 CacheSize 为10,首次打开 Recordset 对象后,提供者将前面10个记录调入本地内存。

责任编辑

杨锡林

封面制作

南方立德 (Leader) 信息技术中心

V

Visual Basic 6.0数据库

恭喜您选中本丛书,
它具有如下特点:

精选最新的软件

注重原理和概念

突出方法和技巧

加强实例的训练

Visual Basic 6.0数据库



ISBN 7-301-01764-2



9 787301 017647 >

ISBN 7-301-01764-2/TP·120

定 价: 28.00 元