
学校的理想装备
电子图书·学校专集
校园网上的最佳资源
Visual Interdev 6.0
程序员指南（五）
BOOK
内部资料 非卖品

返回总目录

第四篇　数据库集成

第四篇　数据库集成

第四篇提供的内容包括以下几方面：数据连接、数据库管理、数据环境对象模型、服务器和客户机访问数据库，以及其它的数据访问信息。

第十八章　数据库概念

这一章讨论作为 Visual InterDev 基础的一些概念：“数据访问结构”、“数据环境”和“数据绑定”。

第十九章　查看数据

在 Visual InterDev 中，你可以使用数据绑定设计期间控件来显示你 Web 页面上的数据。这一章阐述“获得记录”和“显示你 Web 页面上的数据”。

第二十章　修改数据

除了显示数据之外，你可以创建允许用户修改数据的页面，使用户可更新已有的记录、添加记录和删除记录。

第二十一章　直接访问数据库

这一章包括调试已存过程和在数据环境中用数据库命令进行工作。

第二十二章　管理数据库项目

数据库项目允许你把控制扩展到你的 SQL 服务器上或 Oracle 数据库上。

第十八章　数据库概念

18.1　数据访问结构

Visual InterDev 允许你非常灵活地在设计 Web 应用中使用数据库部件。你可以使用任何得到 ActiveX Data Object(ADO, Active 数据对象)支持的数据库(你拥有驱动程序),其中包括 Microsoft SQL 服务器、Microsoft Access、Oracle 和其它数据库。

你可以直接同数据库交互工作,或使用视图、已存过程和其它数据库实体来管理数据库。数据库可以实际放到你 Web 服务器的同一个计算机上,也可以放到其它计算机上。你可以使用 Web 服务器完成所有数据库访问,也可以从一个客户计算机上直接访问数据库。

要准备使用具有特色的 Visual InterDev 进行数据访问,首先要理解下述概念:

- 在 Visual InterDev 中的数据库集成
- Web 服务器和数据库
- 数据连接
- 数据库管理

在 Visual InterDev 中的数据库集成

Visual InterDev 把范围广泛的一系列性质都集成起来，帮助你使用数据库部件创建 Web 应用：

- **数据库项目** 你可以把它加到你 Visual InterDev 解决方案中的一类项目，包括建立和管理你的数据库所需要的工具。这些工具作为与 Web 页面分开的部件。
- **数据视窗** 提供数据活动视图的窗口，你的数据库或 Web 项目当前都连接到这个窗口上。不管你是在数据库项目中还是在 Web 项目中，都可以从数据视窗启动管理你数据库的工具。
- **Microsoft Visual Database Tools（可视数据库工具）** 以图形方式管理与查询数据库的一组工具。这个 Database Designer（数据库设计软件）允许你创建和修改表定义、栏定义，以及微软 SQL 服务器和 Oracle 数据库中表之间的关系。使用 Query Designer（查询设计器）你可以创建和运行 SQL 语句。View Designer（视图设计器）允许你创建视图。
- **数据环境** 在你 Web 项目存储信息的地方要求连接并访问数据库中的数据。数据环境中保存着可重复使用的连接串，允许你从 Web 页面上访问数据库。此外，数据环境还保存着数据命令，用于根据数据库对象或 SQL 命令表示记录集。
- **数据绑定控件** 如文本框、按钮等控件，你可以把它们放到页面上，自动绑定在数据库记录中指定的段上。数据绑定控件已经包括建立数据连接、提取数据，以及更新数据库所需要的脚本，因而不需编写脚本就可以把数据库访问机制建立到你的带有标题的 Web 页面上。

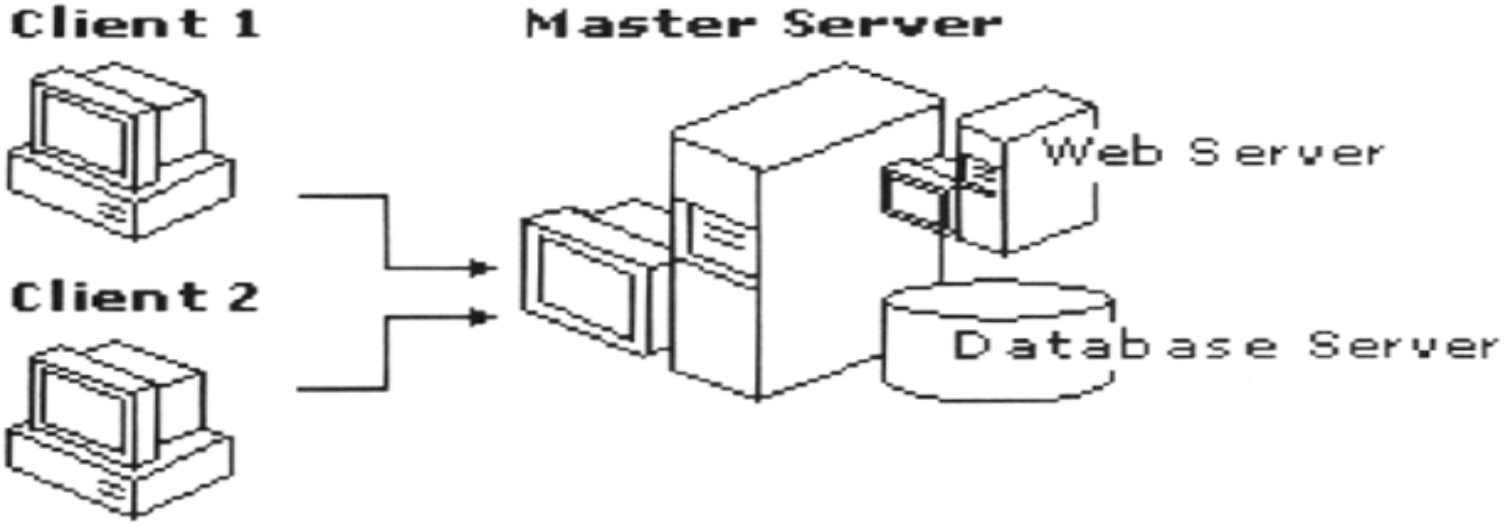
- **数据库对象源控件** 数据库和源控件系统之间的链接，因而你可以存放 SQL 脚本，并在源控件下编译已存过程。这就使得在有多个数据库编程人员的公司中进行开发变得比较容易，也比较安全。

Web 服务器和数据库

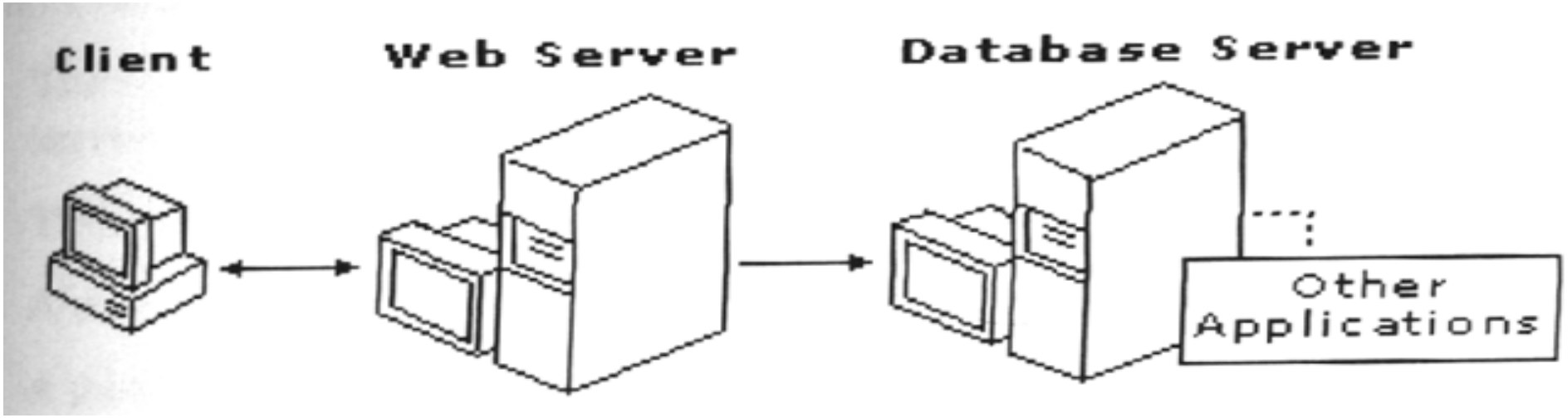
在访问数据库的 Web 应用中，出现两个“相似服务器”（server-like）功能：一个是处理页面请求的 Web 服务器；一个是数据库服务器，或处理数据库访问的等效软件。尽管这两种服务器功能是同一个应用的组成部分，但每个功能是彼此独立的。

你可以用不同方式配置 Web 服务器和数据库服务器，具体用什么方式，取决于你希望人们如何使用这个数据库服务器，你的 Web 应用目标观众是谁，以及在你们业务中这个应用同其它应用是什么关系。下边是可能的配置选项：

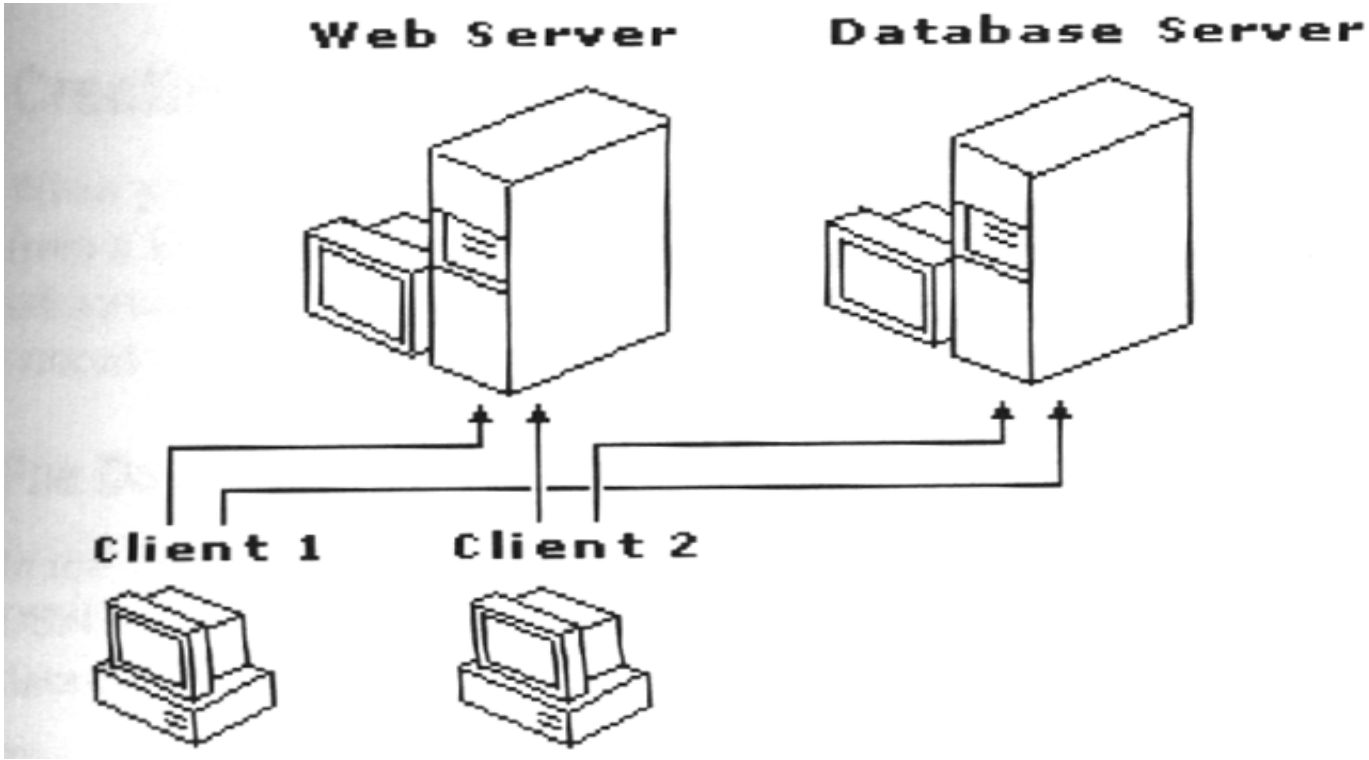
- 数据库服务器可以同 Web 服务器在同一个计算机上运行。



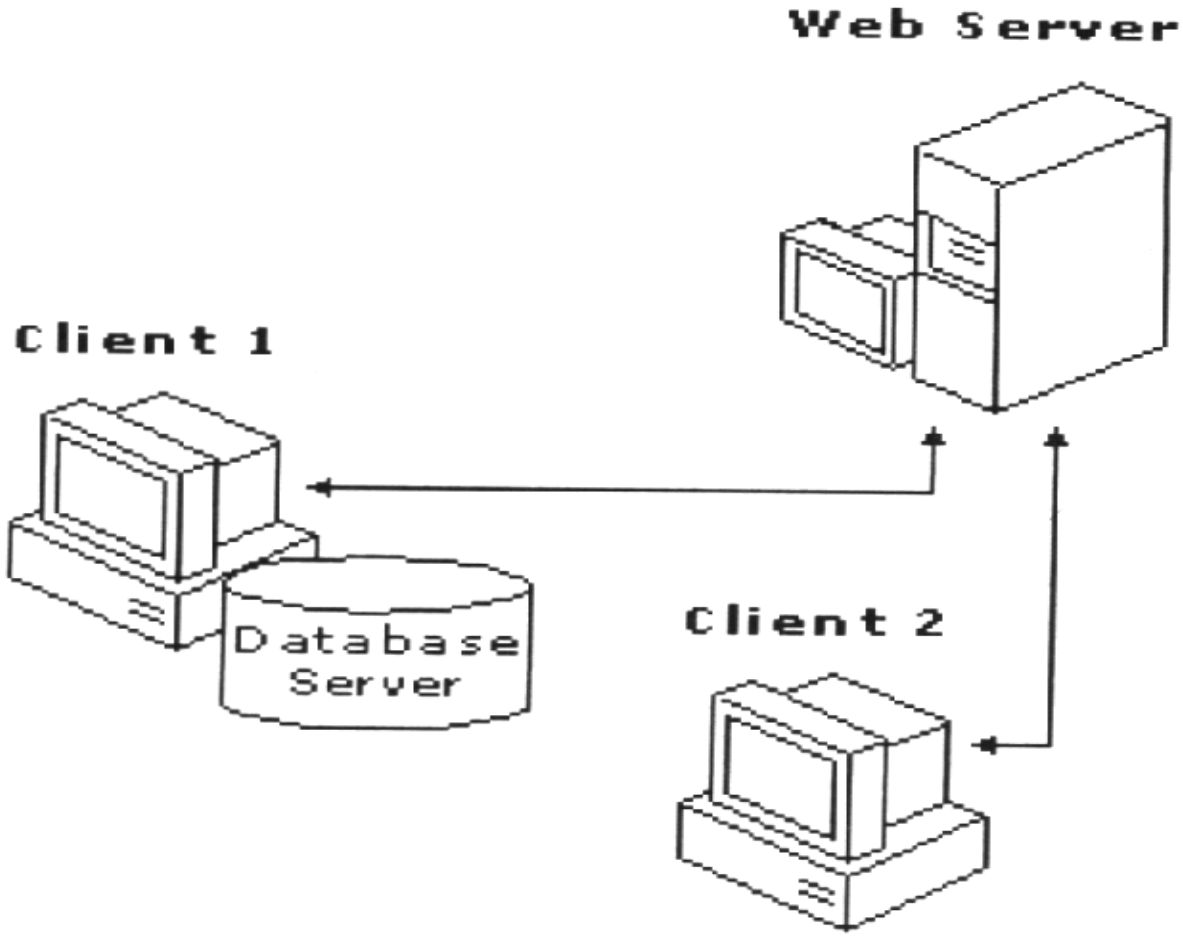
- 数据库服务器可以同 Web 服务器放在不同计算机上。如果你希望每台计算机根据自己的任务进行优化，或者你希望在多个 Web 服务器中分享数据库服务器，或者数据库服务器用于 Web 应用以外的应用，这时，你就应该这么做。



- 数据库服务器和 Web 服务器可以是完全分开的两个进程，彼此不相互通信。如果客户计算机可以直接访问数据库，这种模式就是很实际的，这样可以提高性能。



- 数据库可以在本地（客户）计算机上。在特定情况下，例如测试，你就可以这样做。



使用 Web 服务器作为数据库网关是最普通的策略，因为它使 Web 应用达到最广泛的范围，即服务器干数据库的事，这样，用户有什么类型的浏览器都没关系了。因此服务器方数据库访问，对 Internet 公共站点来说是一种很好的选择，在这里，用户可以使用任何浏览器访问站点。

在另一方面，客户方访问可以提供更丰富的客户经验，尤其是完成任务的

经验，因为浏览器可以独立于 Web 服务器管理数据集。然而，客户方访问要求用 Internet Explorer DHTML 作为浏览器，而且这个数据库要借助于特定数据库驱动器才能访问。因此，客户方数据库访问在内部网站点是最实际的，例如在公司 Web 站点上，在这里，用户使用带有可知性质的标准浏览器访问站点。

关于 Web 服务器和数据库服务器安全的问题，参见第六章“Web 项目概念”中的“安全”一节。

数据连接

要使用数据库，你就要把数据连接加到你的 Visual InterDev 项目上，告诉项目如何访问数据库。对于典型情况，一条数据连接包括诸如下述信息：

- 你正访问的数据库类型（例如，微软 SQL 服务器）和服务器名字（如果合适的话）
- 数据库名字（例如 pubs）
- 用户名字
- 口令

如果需要的话，你可以把许多数据连接加到你的项目上。例如，如果你的应用要求访问两个不同的数据库，你就需要加上两条数据连接。关于如何添加数据连接的细节，参见第三章“数据库基础”中的“连接数据库”一节。

创建连接

当你创建一条数据连接时，Visual InterDev 从你的计算机上 DSN（数据源名字）中读出连接信息。你的计算机可以在文件 DSN（存储在一个 .dsn 文件中）

或机器 DSN，即系统 DSN（存储在你计算机的 Windows 寄存器中）中，提供 DSN 信息。

文件数据源名字

在一个文件 DSN 情况中，Visual InterDev 从该 DSN 读出连接信息，提取连接串。然后把连接节点加到你项目数据环境中。

从 DSN 中提取的连接串存储到你 Web 项目的一个称作 DataEnvironment.asa 的二进制文件中。这种连接称作“无 DSN”连接，因为 Web 项目不再需要 DSN 建立连接。此后，Visual InterDev 便可以简单按照需要从上述二进制文件中读出连接信息。

机器数据源名字

在使用机器或称作系统 DSN 情况中，不使用连接串，而是在每个开发机器上或 Web 服务器中都要重新创建 DSN。

ODBC 驱动程序

在运行的时候，服务器必须有相应的 ODBC 驱动程序以产生到数据源的连接。按照默认配置，在你安装 Visual InterDev 的时候，ODBC 驱动程序便安装到你的开发计算机和你的服务器上。然而，如果你在其它服务器上使用你的应用时，必须保证这些服务器也有正确的 ODBC 驱动程序。

数据连接的安全

一般说来，在你创建一个应用的时候，你会希望拥有各种可能特权，以便

你可以按照需要操纵数据库和数据。然而，在用户访问数据时，你希望他们只拥有运行应用所需要的最小特权。

你的最大风险是在数据库上创建用户开工文件(profiles)。例如，你可能创建一个管理员级用户开工文件，为你自己在开发时使用。你还可以创建一种用户开工文件，这种开工文件具有与你应用的用户相应的特权。如果数据库服务器不与 Web 服务器在同一个计算机上，你还必须保证在操作系统级别上已经定义了正确的用户开工文件。要了解更多内容，参见第六章“Web 项目概念”。

接着，当你创建一条数据连接的时候，你可以规定设计期间验证和运行期间验证。设计期间验证是在开发应用的时候你使用的用户名和口令验证。运行期间验证是应用正在运行的情况下连接数据库时，Visual InterDev 使用的用户名和口令进行的验证。

在规定设计期间验证的情况下，你要选择你的验证达到的安全程度。对于最可靠的安全，你可以选择在你每次连接数据库的时候都提示键入口令。如果你对安全不关心，你可以选择不要提示口令。在这种情况下，Visual InterDev 把你的口令进行加密，存储到项目中。

当你规定运行期间验证情况下，你不必这样选择：你不能向用户提示口令，因为在 Web 服务器上还需要出现提示。因此你必须在用户名字上包含口令。这个口令是加密的，并且存储在项目中，以便在应用运行的时候，用户每次连接数据库，都可以把口令传递给数据库。

数据库管理

除了帮助你创建链接数据库的 Web 页面，Visual InterDev 允许你管理数据

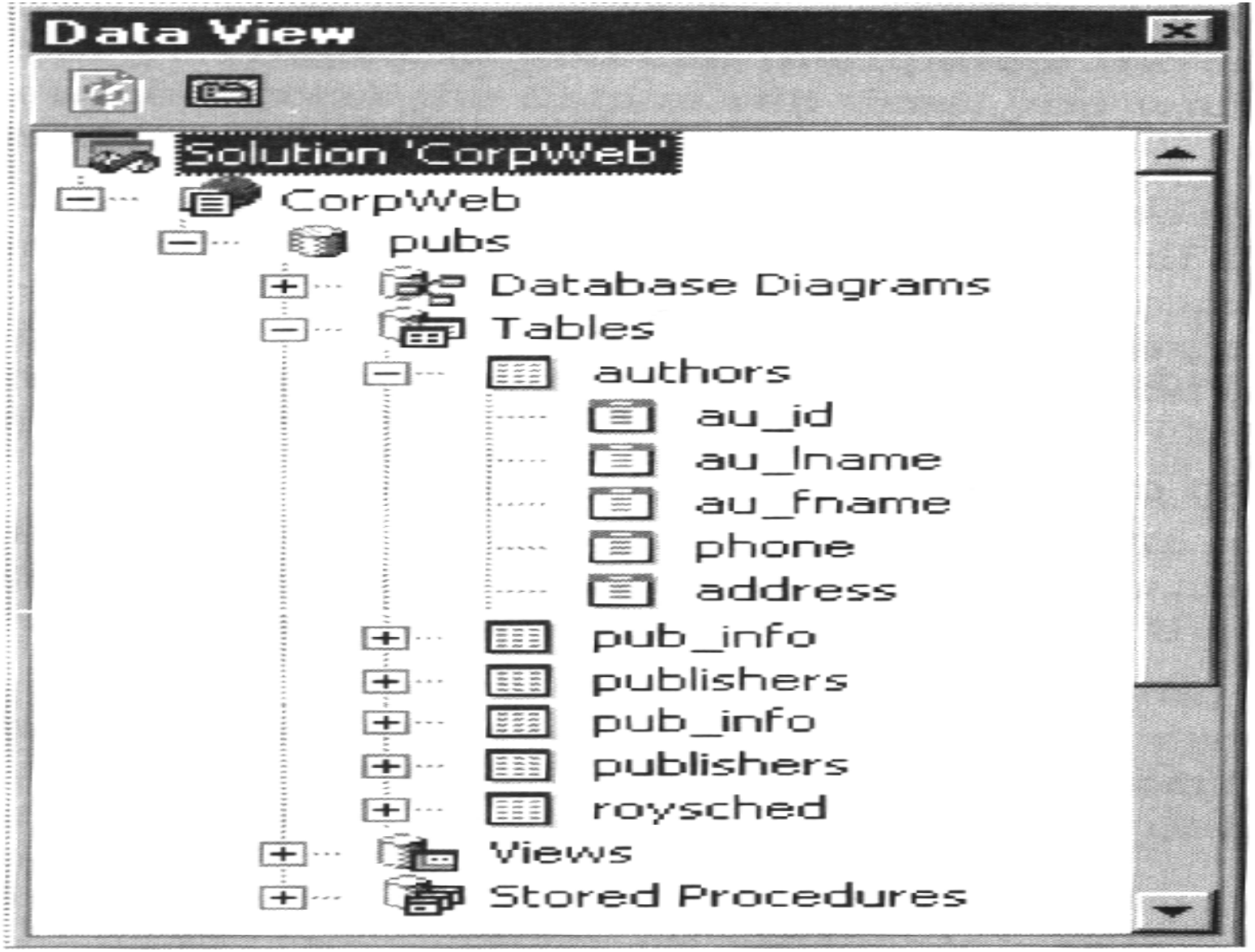
库，而在设计的时候是直接连接到数据库上的。根据你的数据库特性和你访问的特权，你可以使用 Visual InterDev 工具添加、移动或修改下列内容：

- 数据库
- 表或栏
- 视图和同义词
- 表之间的关系
- 索引
- 约束与触发器
- 已存过程、功能和程序包
- 查询返回的数据组，或者查询对数据库的修改：添加、更新、复制和删除记录

注意：允许你对数据库进行结构编辑的特点，只有在 Visual Studio 企业版中才是可行的。

管理数据库同把数据库功能加到Web应用上是不同的任务。因此，要在 Visual InterDev 中管理数据库，你就要创建一个数据库项目。为了帮助你完成管理不同数据库的任务，在 Visual InterDev 中的数据库项目提供下列工具：

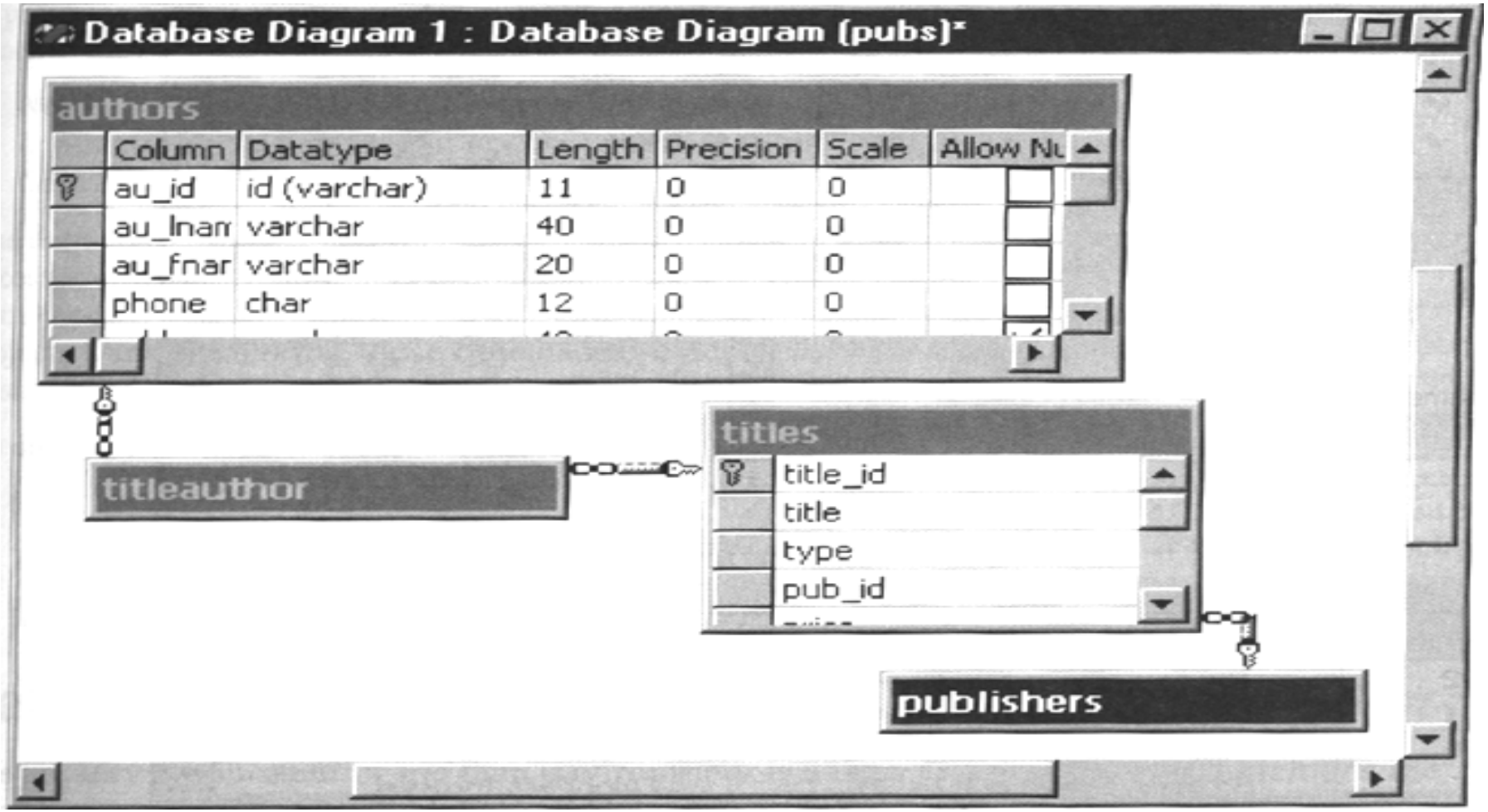
- **数据视窗** 显示你可以在当前使用的所有数据库项目的窗口。从这个数据视窗，你可以编辑项目，例如表格、视图、已存过程和触发器。



显示可用数据库对象的数据视窗

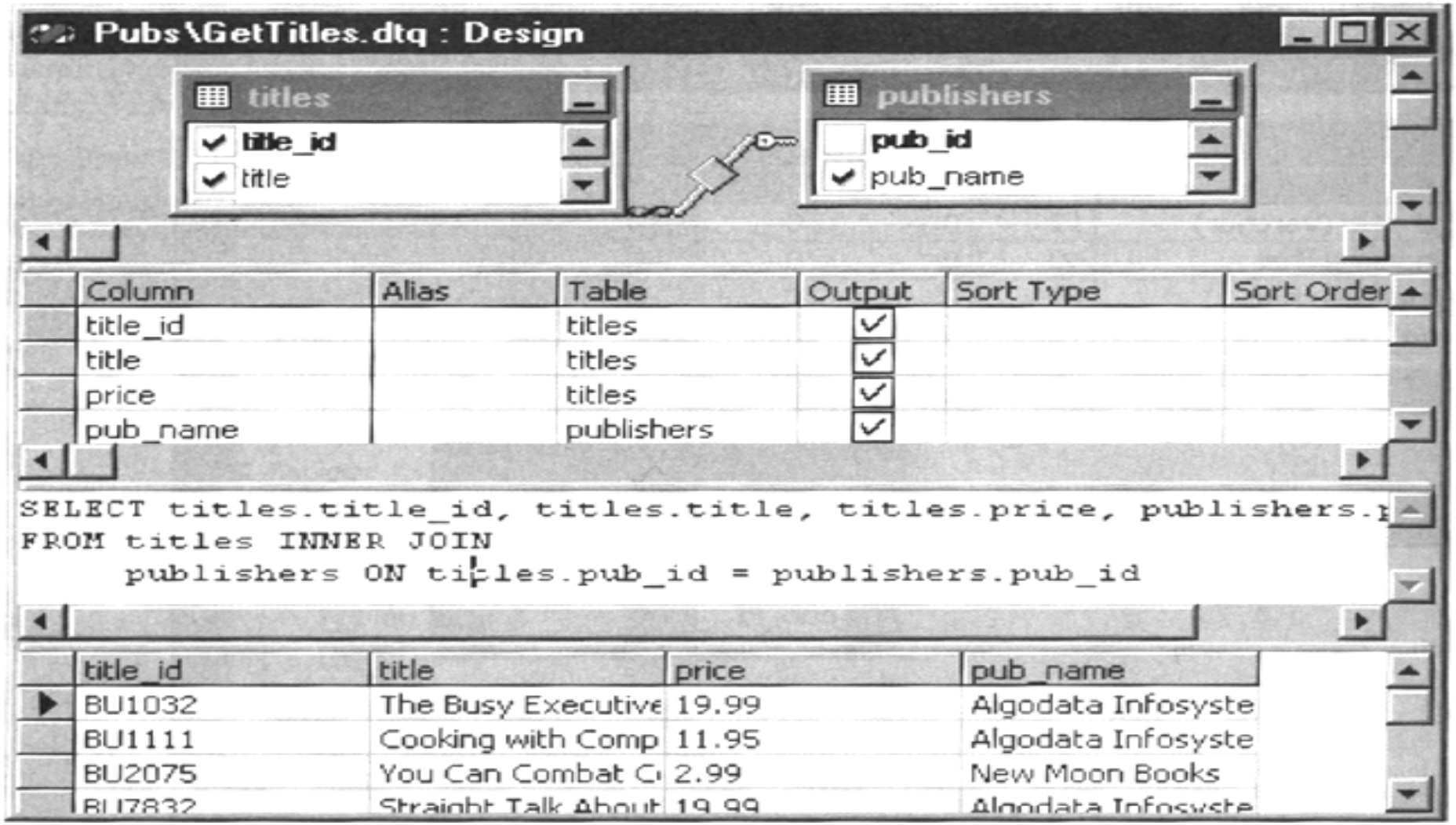
- Database Designer（数据库设计器）该工具显示 Microsoft SQL Server

或 Oracle 数据库作为数据报表，也可编辑以增加/修改报表和栏的定义，定义关系，创建索引，及增加限制条件。



数据库报表表示可视数据库

- **Query Designer（查询设计器）** 允许你直观创建 SQL 语句，查询和修改数据库的设计软件。



用于创建 SQL 语句的查询设计器

- **View Designer（视图设计器）** 查询设计器的一种版本，它允许你直观创建一个可以定义视图的 SQL 语句。
- **已存观察编辑器** 创建已存过程的窗口，其中包括一条到达用于创建

SQL 语句的“查询设计器”的链接。

- **触发器编辑器** 用于创建触发器的窗口。
- **脚本编辑器** 用于创建 SQL 脚本的窗口，而 SQL 脚本是独立于任何具体数据库的 SQL 语句。你还可以把 SQL 脚本放到源控件之下。

18.2 数据环境

数据环境是在你的 Visual InterDev Web 项目中存储信息的环境，这里的信息是在服务器脚本中用于连接和操纵数据库中数据的所需要的信息。它提供一种标准界面用于创建可重复使用的数据关联对象，并把这些对象放到 Web 页面上。

注意： 数据环境可在服务器上使用。如果你正在设计客户可访问数据（使用 Internet Explorer 4.0）的 Web 应用，数据环境在设计期间就可以使用，但是在运行期间不能使用。

数据环境还提供给你在脚本中可引用的对象，允许你访问与管理数据库对象，如表格、视图、已存过程和编程用的 SQL 命令。它还提供一种容易使用的围绕 ActiveX Data Object（ADO，ActiveX 数据对象）的换行器(wrapper)，使得这些对象在 Visual InterDev 中更容易访问更容易使用。

要了解数据环境，你必须了解下述概念：

- 数据环境内容
- 数据环境中拖放方案(scenario)
- 数据环境对象模型

数据环境内容

数据环境中主要部件是数据连接，这个数据连接包括使用指定用户名连接到一个数据库上所需要的信息。例如，你的数据环境可能包括一条连接，这条连接使用用户名字 Admin(设计期间)和 Guest(运行期间)链接到 Microsoft SQL 服务器上 Pubs 数据库中你的应用上。如果你的应用需要访问多个数据库，你可以向数据环境加上多条数据连接。

在每条数据连接中，你可以加上一个或多个数据命令（命令对象），这些命令定义工作中使用的一组数据。命令对象可以引用一个数据库对象，例如表格、查询、视图、同义对象、已存过程或 SQL 语句。例如，你可能创建一个引用作者表格的命令对象，以便在 Web 页面上显示这个表格的内容。你还可以定义辅助命令对象，引用你可以调用的查询和已存过程，以便显示和更新其它表格中的数据。

显示数据环境和数据命令的一个 Web 项目

在你应用中的任何页面都可以访问命令对象，因而命令对象变成可重复使

用的对象。如果更改这个基础数据库，你可以对命令对象进行单项更改，这样引用命令对象的所有页面都可继续正常工作。

每个命令对象都是一个结点，它包含与该命令对象有关的辅助信息。例如，命令对象引用一个表格，表格中包含一个列表栏。一个涉及到已存过程的命令对象包含一个栏列表，列表中的这些栏是由该过程返回的；命令对象也可以包含过程参见列表。关于添加命令对象的更多内容，参见第十九章“查看数据”中的“获得记录”一节。

Visual InterDev 在你第一次定义一条数据连接的时候为项目创建一个数据环境。你一旦把连接建起来，Visual InterDev 就创建 DataEnvironmet 文件夹，而且把它作为一个结点置于 Global.asa 文件之下。你加上的这个数据连接在 DataEnvironmet 节点上显示出来。

在你建立辅助数据连接的时候，这些辅助连接都加到 DataEnvironmet 节点上。

注意：在 Visual InterDev 项目中你只有一个 DataEnvironmet 节点。添加数据连接的更多内容，参见第三章“数据库基础”中的“连接数据库”一节。

数据环境中的拖放方案

数据环境的一个重要特性就是你可以拖放对象，简化向你的应用添加数据库访问机制的过程。要创建新命令，你可以把数据库对象从数据视窗拖放到数据环境中。此外，你还可以通过把命令和数据库段从数据环境拖放到你页面的办法在页面上创建数据绑定控件。

下表汇集了如何在数据环境中使用拖放功能。

拖动对象	从	拖到	作用
数据库对象	数据视窗	数据库环境中的连接	为你拖动的数据库对象创建命令对象。例如，拖到一个表格去创建一个命令对象，而它的数据库对象类型就是表格
命令对象或字段对象	数据环境	Web 页面	创建数据绑定控件
数据库对象	数据视图	Web 页面	不允许。要从数据环境中拖出对象

关于使用数据绑定控件的细节，参见第十九章“查看数据”

数据环境对象模型

数据环境支持自己的对象模型。你可以在编写脚本时使用该模型去操纵希望在 Web 页面上显示的数据。数据环境对象模型是以 AcitveX Data Object（ADO，ActiveX 数据对象）对象模型为基础的，但是使用起来要简单一些。

在 ADO 中，数据环境对象模型的主要对象是连接对象、命令对象、记录集对象、字段对象和参数对象。这些 ADO 对象中的每一个都有自己的特性和方法。

数据环境把这个对象模型加以概括，使它使用起来比较简单。数据环境本身就是一个对象，是可以在脚本中使用的并且包含这些 ADO 对象的对象。在这个数据环境对象中，命令对象是作为脚本中的方法使用的。你可以调用一种命令方法去创建它，并返回该命令所涉及到的记录集，或者去执行它的 SQL 命令

或已存过程。

数据环境中的每个对象还有你可在属性页面上为项目设置的属性，或直接在脚本中设置的属性。关于数据环境对象模型和每个模型的属性有关方面的更多内容，参见第二十一章“直接访问数据库”中的“使用数据环境执行数据库命令”一节。

18.3 数据绑定

由于浏览器、Web 服务器和数据库之间的相互作用，以 Web 页面访问数据库同在传统应用方式中用数据库进行工作是不同的。而且，以服务器为基础的数据库访问和以客户为基础的数据库访问也是不同的。

过一会儿将解释，如果你使用 Visual InterDev 工具为你的应用设计数据库访问机制，你作为应用开发者，不需要担心这些基本差别。尽管如此，了解下列概念还是有用的：

- 服务器访问数据库
- 客户访问数据库
- 数据绑定控件和脚本对象模型。
- 在 Visual InterDev 应用中的数据库访问设计

服务器访问数据库

在以服务器为基础的数据库访问机制中，数据库和用户之间的相互作用同普通客户-服务器数据库应用共享某些特性。然而 Web 服务器位于两者之间，而

且以自己的特性引出一个相互作用层。

下列特征描述了浏览器、Web 服务器和数据库服务器是如何相互作用的：

- 客户（浏览器，给出数据录入窗体和报告）到数据库没有直接通路。客户只能把请求传递给 Web 服务器，Web 服务器再传递给数据库，或把来自数据库的请求传递出去。
- 满足的数据库请求，把记录集发送给 Web 服务器。数据库服务器本身不维护记录集，进而也不追踪记录集中的当前记录。
- 同数据库相互作用是由服务器脚本，而不是由客户脚本执行的。服务器脚本在页面发送给浏览器之前进行处理，因此当用户同页面发生相互作用时（例如通过单击按钮），对该页面的所有数据库访问都已经完成。
- 浏览器和服务器不再继续接触。服务器不保持有关浏览器以前请求的信息。例如，服务器不保持浏览器最后请求数据库中什么记录的有关信息。

因而，在 Web 应用中用户和数据库的相互作用同传统应用的解决方式是不一样的。例如，一个常见的情节是用户要看包含数据库信息的窗体，因而希望在记录之间反复翻动页面。

请求完成这个任务的典型事件顺序如下：

1. 用户请求一个包含该窗体的页面。
2. 在服务器处理这个页面的过程中，执行脚本，连接数据库，然后执行一次查询或已存过程，以便返回一个或多个记录。
3. 服务器把一个具体记录从记录集中移出来。
4. 服务器从当前记录中提取数据，再把数据按照 HTML 文本写到 Web 页面上。关于当前记录位置的有关信息，如书签和主关键字，都存到一个全局服务

器变量中，或者包装起来同页面一起送给浏览器。典型情况是服务器之后丢弃这个记录集。

5. 服务器把页面发送给浏览器，在这里，用户以 HTML 格式查看数据库记录。
6. 用户单击页面上“Next”按钮。这个按钮的脚本把 HTML 格式提交给服务器。因为该服务器没有到浏览器的连接，因而按钮脚本必须把一种指示传递给服务器：该用户向下个记录导航。如果当前记录的位置早已经发送给浏览器，浏览器就把这个信息送回给服务器。
7. 在服务器中，服务器脚本分析用户提交的内容。在这种情况中，服务器发现用户希望导航。然后，服务器重新发出先前的查询，这个查询返回一个新记录集。服务器使用先前保存的书签或主关键字进行导航，到达记录集中正确位置，然后向前移动一个记录，再重复步骤 4 和步骤 5。

这个方案似乎很复杂。实际上，Web 服务器一旦把页面发送给浏览器，就把记录集和用户状态都忽略掉。比较起来，这个过程有点像打电话，同朋友说完话就把电话挂了。每次你拨号叫通电话，你的朋友都把你以前说过的事忘了。

有可能每次查询的记录集都不重复。然而在浏览大量数据的时候，不建议暂时存储。如果你暂时存储，即便用户数目不大也很容易占满服务器资源。但是，如果记录集只包含一行，而且你又要使用乐观锁定(optimistic lock)去更新它，把这个记录集在服务器中暂时存储一下会提高效率。

通过服务器访问数据库，客户环境便不直接控制数据库访问。当服务器从记录集提取信息并把它写到页面上的时候，对客户脚本来说，信息就变得同普通 HTML 文本没有区别了。客户脚本不能直接执行一条命令去在数据库中进行移动。客户脚本必须把足够的信息发送给服务器脚本，以便服务器脚本可以从停

放数据的地方提取出来。

注意:你可以使用设计期间控件和 Visual InterDev 脚本编程对象模型去创建应用，通过这些应用完成用户请求服务器的透明处理过程。详细情况参见第二十四章“用设计期间控件和脚本对象编写脚本”。

客户访问数据库

如果你的开发环境很现实，就可以创建从客户到数据库的直接数据库访问机制。然后你可以从客户脚本全面管理数据库访问机制，这样常常可以得到较快访问数据库的效果。此外，这种开发环境允许你利用浏览器特性的优势创建丰富的用户实践。

要从客户那里访问数据库，你可以使用 Internet Explorer 4.0 特有的动态 HTML 特性。你的所有用户都必须用 Internet Explorer 4.0 作为他们的浏览器。此外，你必须在服务器上保持数据库支持恰当的数据访问软件，至少要使用配置恰当的服务器作为到达数据库的网关。

当你使用客户机访问数据库时，应用和数据库之间的基本相互作用比在服务器上进行的访问简单一些（如果你使用 Visual InterDev 工具去管理数据库访问，差别便消失了，脚本访问数据库的方式对两种情况都是一样的）。

要实现在前边服务器访问数据库一节中描述的导航方案，就要按下述步骤执行应用：

1. 用户请求包含窗体的页面。
2. Web 服务器把该页面传递给浏览器（如果需要，服务器首先处理页面上的脚本，但数据库访问并不需要服务器脚本）。

3.浏览器接收到这个页面时，页面上的 RDS（远程数据服务）控件自动打开记录集，按照 DHTML 4.0 数据绑定规范，把数据绑定到该控件上。

4.用户单击页面上的“Next”按钮，这个按钮脚本就导航到记录集的下一个记录上。

一项重要不同之处是：在 Web 服务器把页面发送到浏览器上之后，应用不再要求进一步请求服务器管理数据库访问，从而大大减少浏览器和 Web 服务器之间信息往返的次数。此外，因为客户机不临时缓冲记录集，数据库的动作，例如导航，不再要求重新产生记录集。最后进行批量更新发送，以便提高效率。

在各种情况中，所得到的结果都是使数据库访问加快。你可以使用微软的“远程数据服务”（RDS）创建基于客户机的数据库访问机制。有关 RDS 的内容，参见微软 RDS Web 站点 http://www.microsoft.com/data/rds/。

数据绑定控件和脚本对象模型

尽管你没有要求使用 Visual InterDev 数据绑定控件访问 Web 应用中的数据库，但还是大大简化了应用环境。数据绑定控件提高一套完整的用户界面元素集，此外还包括连接到记录集、在记录集中进行导航以及更新记录集在逻辑上所需要的一切。

数据绑定控件包括：

- **记录集**。在向数据库对象绑定时起主要控制作用。页面上的其它控制都依次绑定到记录集控制上。
- **单个控件**。显示数据库记录中单个字段的内容，包括文本框、列表框、检查框、选项按钮和标签。

- **栅格**。显示一组记录。
- RecordsetNavbar。包括允许在结果记录集中到下个记录、前一个记录、第一个记录和最后一个记录导航的按钮。

除了易于添加用于显示数据的用户界面元素之外，数据绑定设计期间控件还可以发挥 Visual InterDev 脚本编程对象模型的优势。这个对象模型完成两项任务。第一项任务，为脚本编程数据库访问机制提供一致性界面，不管你进行工作的数据库类型，也不管你是否编写服务器访问数据库或客户机访问数据库的脚本。第二项任务，创建一个高级模型用于数据库访问和操纵，基于 Web 的数据库访问机制所涉及到的大多数复杂性都对你隐蔽起来。

在使用数据绑定控件进行工作时，你主要使用记录集控件。这个控件使用脚本编程对象模型把可以帮助你管理结果记录集中记录属性和方法都展现给你。例如，在一个结果记录集内的各个记录之间进行导航，你可以调用记录集对象的 moveNext 方法或 movePrevious 方法。同样，单个数据绑定控件还把如下属性和方法展现给你：这些属性和方法绑定在记录集上，确定控件的表现行为。

关于使用控件和脚本编程对象模型的更多内容，参见第十九章“查看数据”和第二十三章“脚本编程概念”中的“脚本编程对象模型”一节。

在 Visual InterDev 应用中设计数据库访问

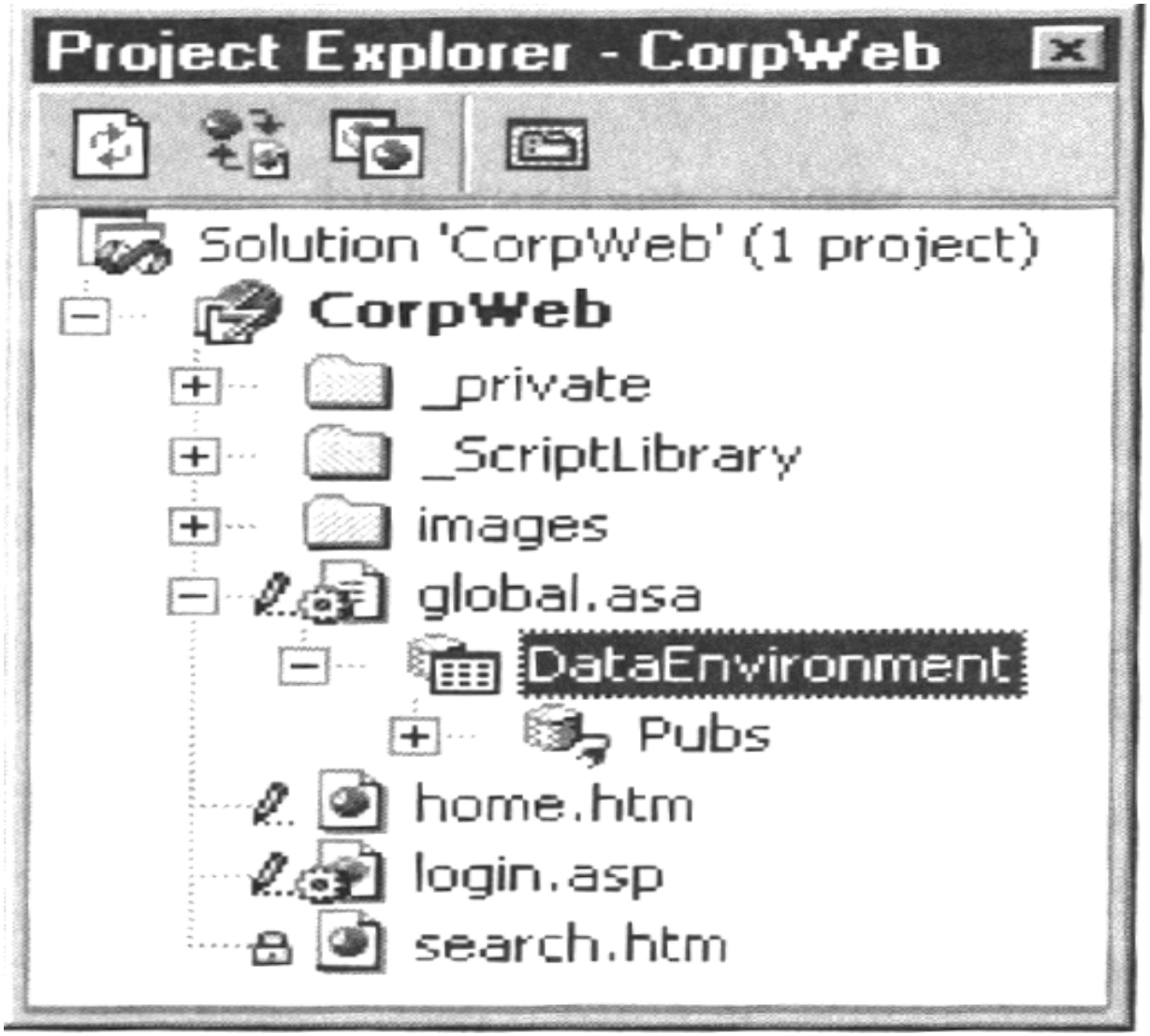
不管你访问什么数据库，Visual InteDev 中包含的工具都可以大大简化这个访问过程。这一节为你介绍如何使用数据库部件创建 Web 应用，强调指出你在哪个方面可以沿着 Visual InteDev 特性的方向发挥优势。

尽管基本过程在基于服务器的和基于客户机的数据库访问中是不同的，Visual InteDev 数据库工具还是产生不同程度的透明性。因为仅仅有很少的不同之处，因而你可以使用同一个过程创建两类访问机制。

在大纲窗体中，创建一个带有具有数据库访问机制的页面所要求的步骤如下：

1. **建立到数据库的连接**。通过在 Project（项目）选择 Add Data Connection（添加数据连接）的办法规定到数据库的连接。

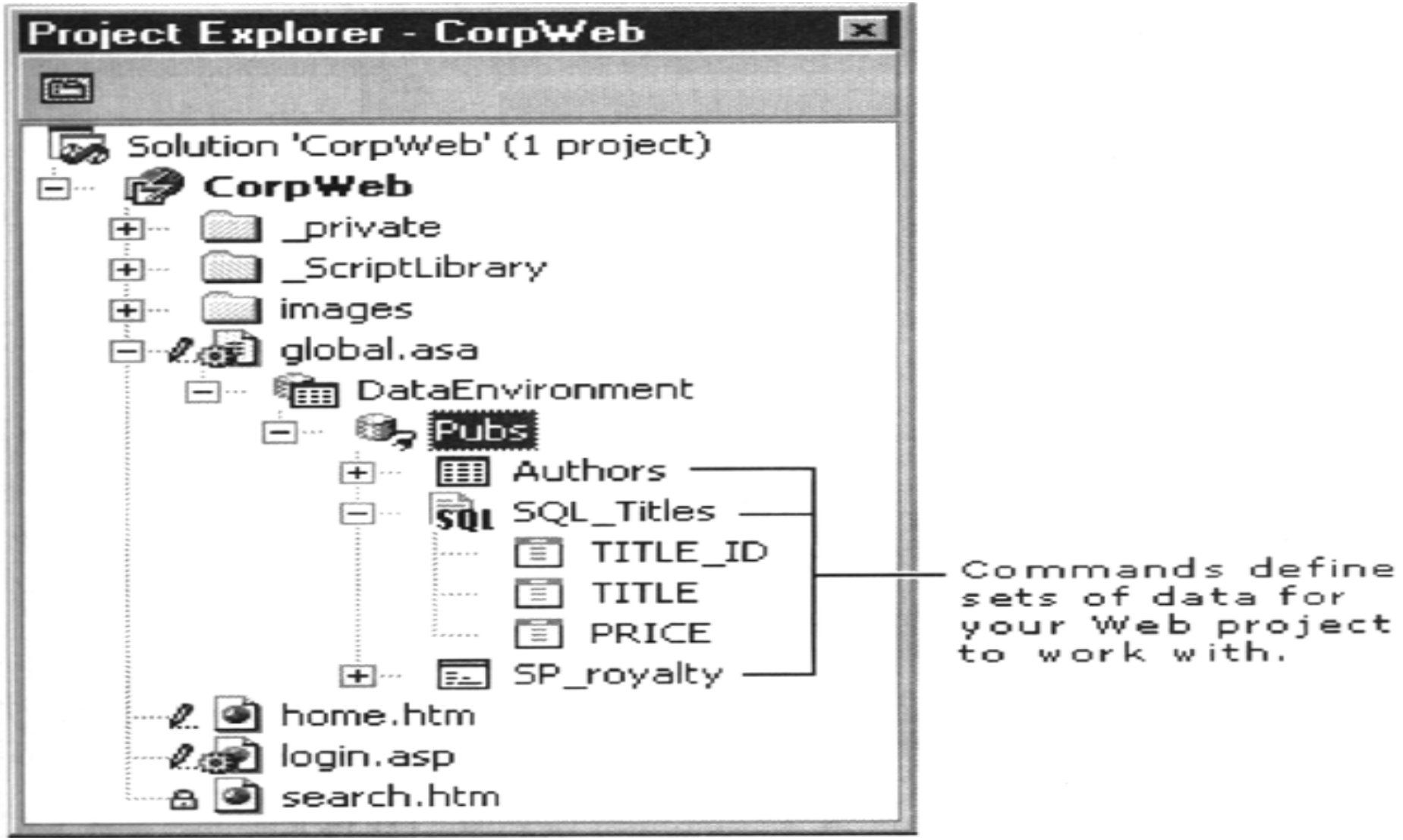
在你这样做的时候，Visual InterDev 创建一个数据环境，这个数据环境对你项目中所有数据连接起存储器作用。因为不同 Web 页面可以共享一条连接，在你 Web 项目的 Global.asa 文件中的脚本便维护这个数据环境。



如果你的应用要求访问多个数据库（包括不同服务器上的数据库），你可以在这个数据环境中建立多个连接。

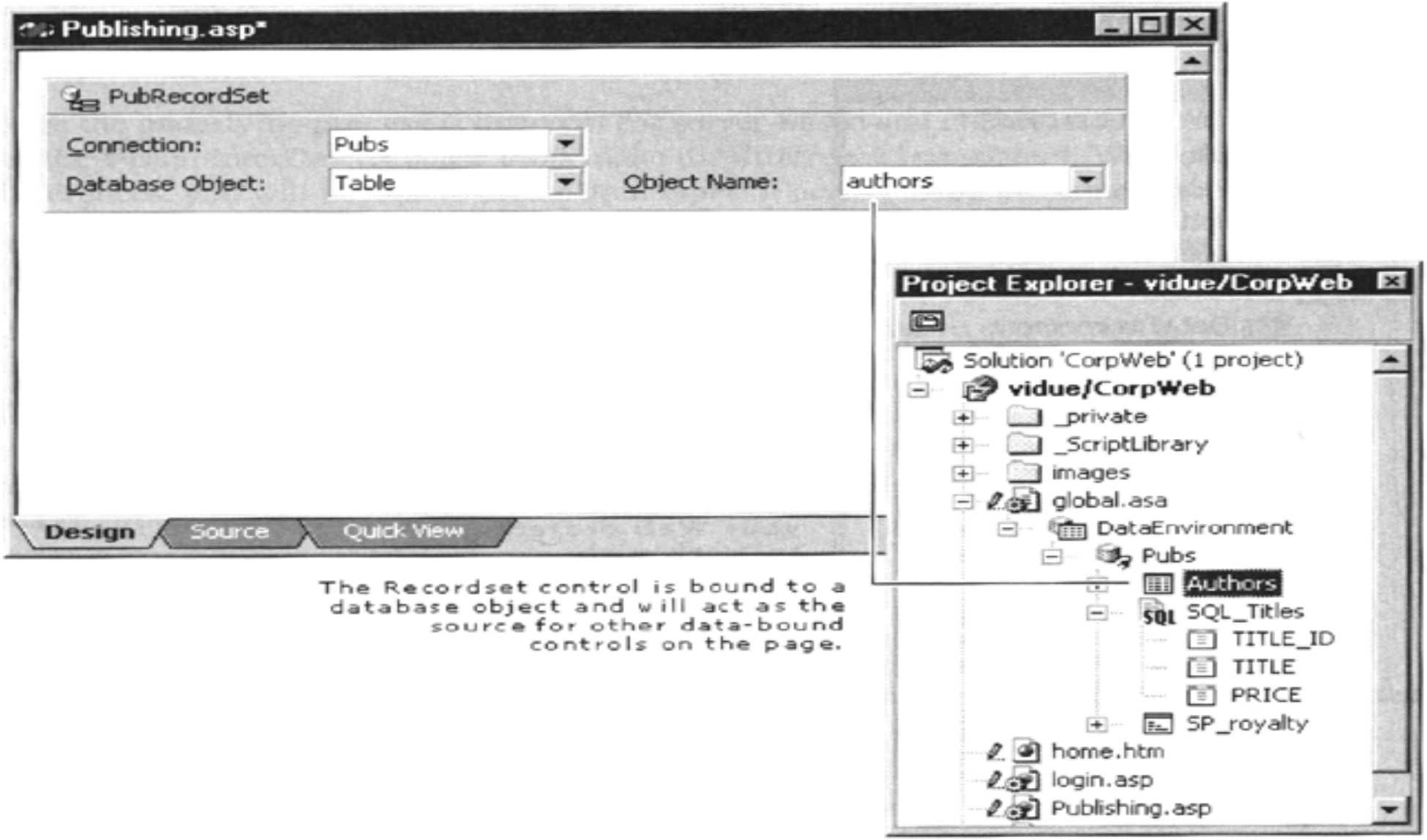
2.**定义使用的数据**。对于数据环境中每条连接，你创建一个或多个“命令”，

这些命令是用于封装引用 SQL 语句、已存过程、表格、视图或同义词的对象。每个命令都生成一个不同的结果集(result set)(结果集也叫做“游标”(cursor))。例如，你的应用从一个表格、一个查询和一个已存过程中读取数据，你就可以定义三个不同命令。



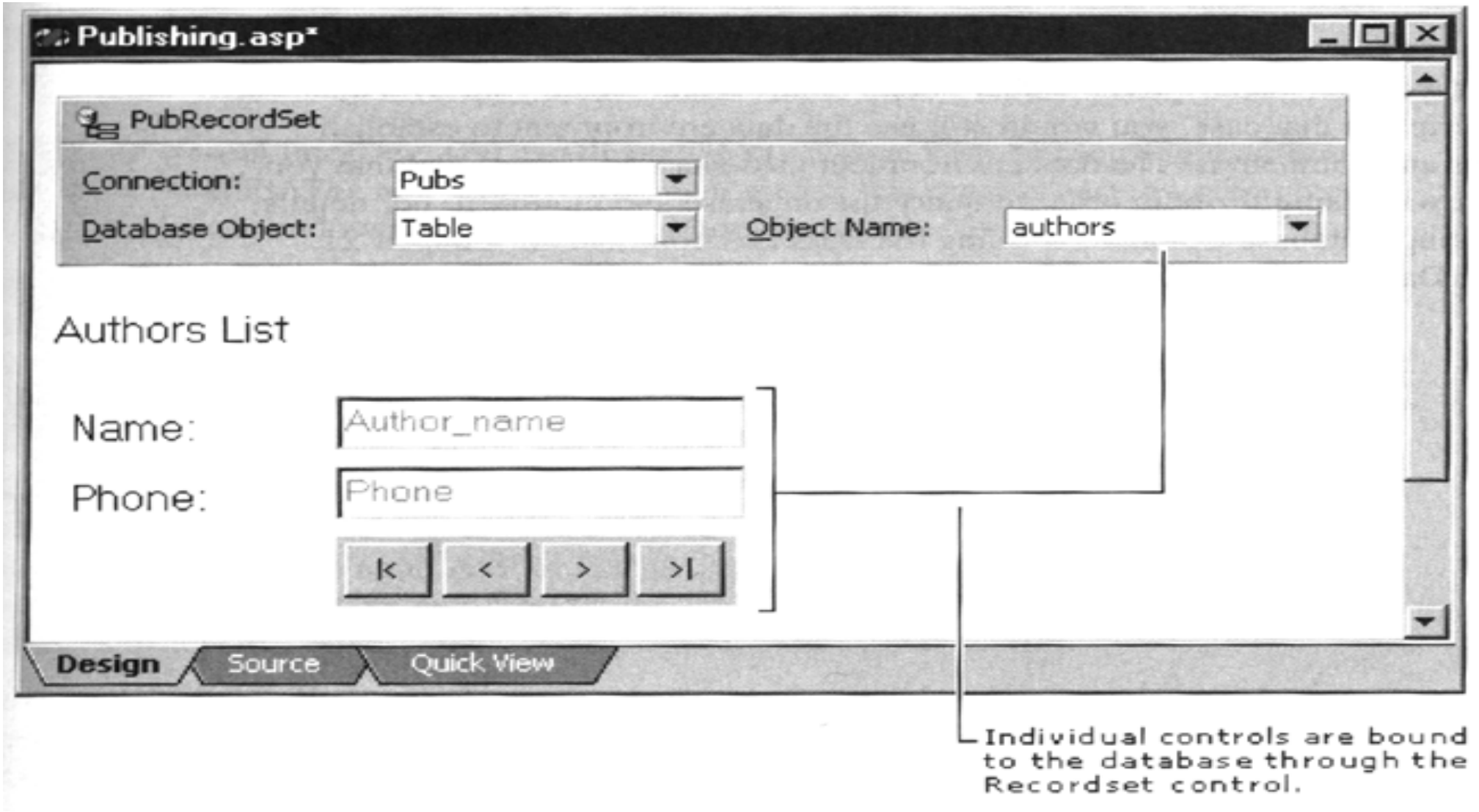
3. **向 Web 页面添加记录集控件**。在你希望使用数据库数据的页面添加一个记录集控件。在页面运行的时候，这个控件起本地控制器的作用，为其它相互作用的数据绑定控件提供一个对象。数据库对象要么直接绑定，要么使

用数据环境中命令对象完成绑定。



4. **向 Web 页面添加数据绑定控件**。在你已经放上记录集控件的 Web 页面上，你可以再添加数据绑定控件，如文本框、复选框、按钮以及导航条。使用

这些数据绑定控件可定义数据录入窗体、报告，以及其它在数据库中用于数据的用户界面元素。每个数据绑定控件都通过记录集控件链接到一个数据库上。



5. **添加你自己的脚本**。如果数据绑定控件不提供在应用中希望使用的数据库所有交互功能，那么你可以编写一个辅助脚本。例如，你可以编写脚本，使用户进入数据绑定控件合法化。你还可以编写脚本，通过调用记录集对象中的方法，完成诸如更新、添加，或删除记录的功能。最后，你还可以编写脚本，引用存储在数据环境中的命令，直接执行数据库命令。

上边按步骤列出的方案是比较简单的，但是更复杂的方案不需要付出更大的努力。例如，一个分类 Web 页面可能包括一个下拉式组合框，客户可以从其中选择一种产品。下拉式列表可以用第二种记录集创建出来（除了记录集被分类页面自己更新之外）。在 Visual InterDev 中的数据访问工具允许你很容易地维护这两种记录集，而且使得把特定控件绑定在每个集合上变得很容易。

如果你希望的话，还可以旁路这些数据绑定控件，在脚本中创建数据库访问机制。在这种情况中，你还要使用数据环境建立链接和命令。数据环境提供一种对象模型，你可以用脚本同这种模型相互作用，去查询数据库或修改数据库。详细情况，参见第二十一章“直接访问数据库”中的“使用数据环境执行数据库命令”一节。

第十九章　查　看　数　据

在 Microsoft Visual InterDev 中，你可以使用数据绑定设计期间控件显示你 Web 页面上的数据。这种新的数据环境使得这项处理变得比较容易：给你在本地创建并管理所有数据绑定控件的能力。

首先，你建立一条到已有数据库的数据连接。这条数据连接出现在 Web 项目的数据环境（DataEnvironmet）文件夹中。然后你可以很容易地使用数据环境中的创建命令对象，添加控件，把这条连接绑定在 ASP 或 HTML 页面上，或者把连接拖放到页面上。Visual InterDev 创建数据绑定控件，显示数据库中的数据。

你还可以把数据绑定控件从 Toolbox 拖放到你的 ASP 页面上以显示数据。例如，你可以创建一条数据连接，并且把基于这条连接的一个“记录集”（Recodset）设计期间控件添加到一个 ASP 或 HTML 页面上。其它多个控件允许你显示由一个“记录集”控件按不同方式定义的记录集中的数据。这些不同方式包括文本框、标签、列表框、复选框和选项组等。

表示你的数据还有更多的灵活方式，你可以发挥“栅格”（Grid）设计期间控件的优势。你可以使用这个控件以栅格格式显示多个记录中数据。

19.1 获 取 记 录

从数据库中获取并显示记录，是 Visual InterDev 关键特性之一。你可使用数据绑定设计期间控件(主要是“记录集”控件)操纵数据。

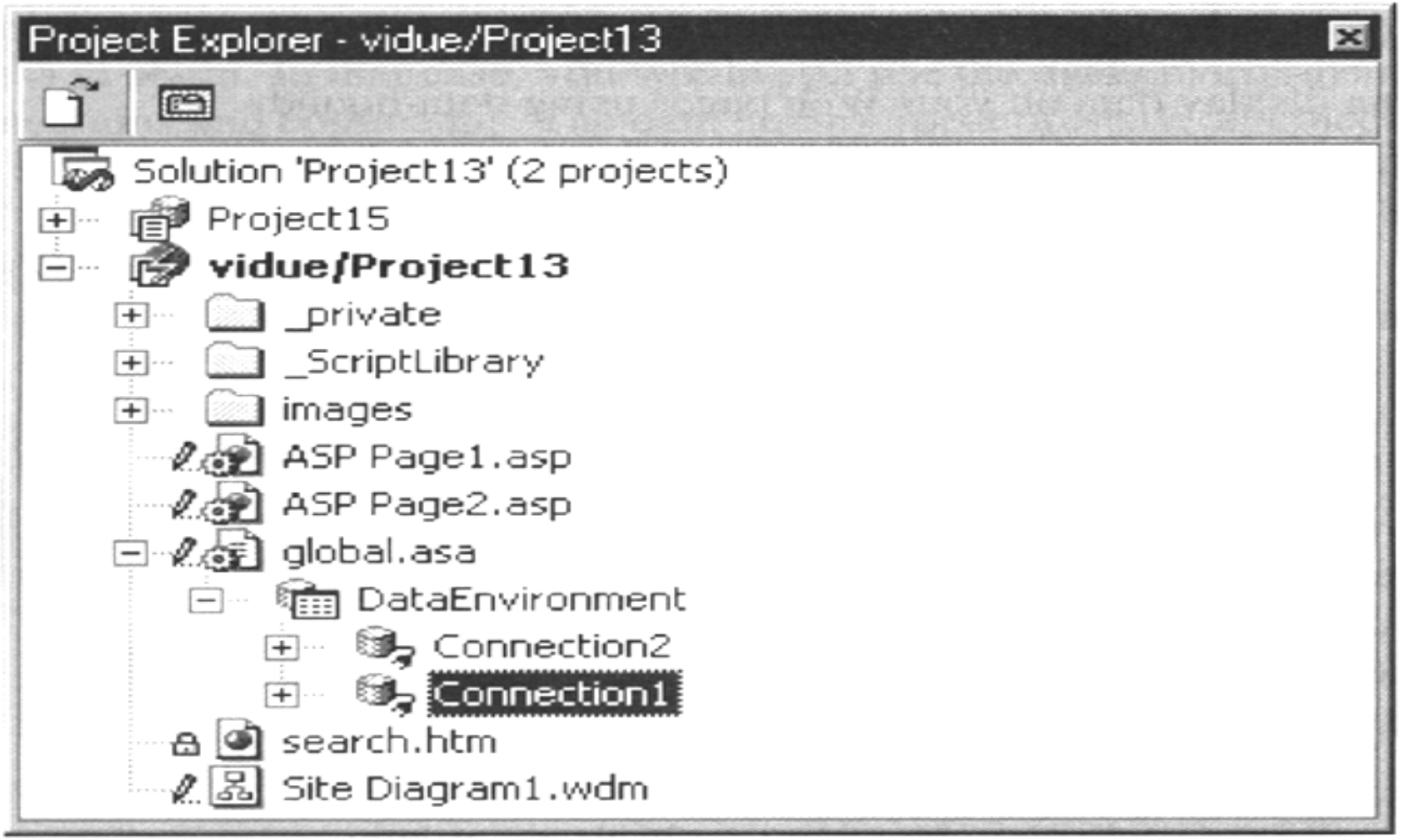
“记录集”(Recordset)控件使用数据库连接绑定到一个具体数据库上，在数据库中规定一个具体的记录集。

Visual InterDev 使得使用数据环境创建“记录集”变得容易。你还可以使用 Toolbox 创建“记录集”控件，然后把这些控件同数据环境联系起来。数据环境如何同控件和 Visual InterDev 其它数据库特性发生联系，更多内容参见第十八章“数据库概念”。

使用数据环境创建一个“记录集”控件

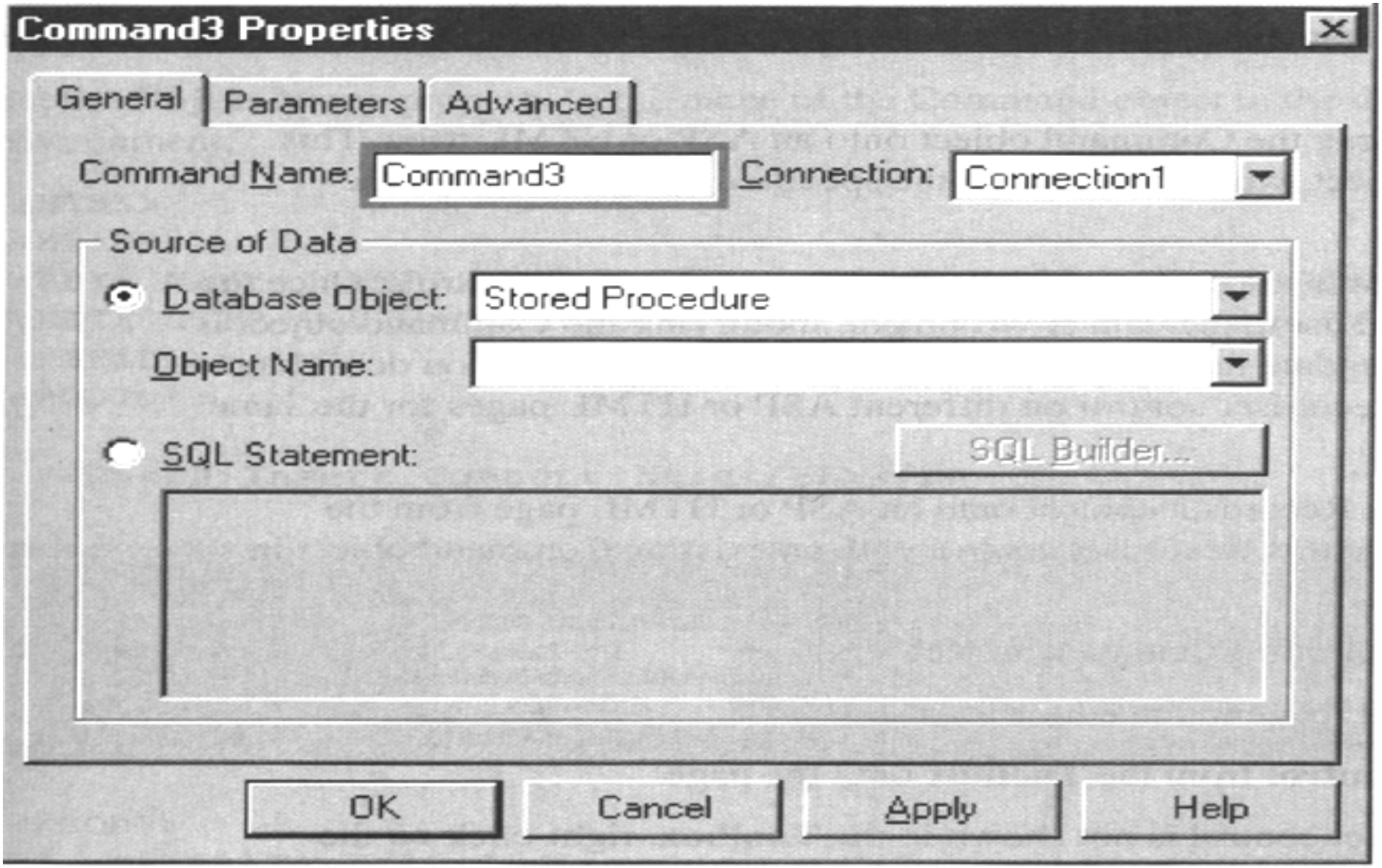
1. 在你的 Visual InterDev 项目中，把一条数据连接加到你希望显示其记录的数据库上。详见第三章“数据库基础”中的“连接数据库”一节。

在 Project Explorer 显示的数据连接是在 DataEnvironmet 文件夹中的 Global.asa 文件中。



提示:你还可以使用右键单击 DataEnvironmet 文件夹，然后单击 Add Data Connection,以创建一条数据连接。当你使用 Project 菜单中的 Add Data Connection 命令时，会出现同样一组对话框。

2.右键单击 DataEnvironmet 文件夹，选择 Add Data Command。
显示 Command Properties 页面。



3．在 Command Name 框中为这个命令对象键入一个名字，例如CustomersTableRecords。

4. 在 Connection 框中选择数据连接。如果在你单击 Add Data Command 之前选择了 DataEnvironment 文件夹中的一条数据连接，这个数据连接便已经在 Connection 框中。
5. 如果你希望这命令对象包含一组记录，而这组记录在在一个表格中、查询中、已存过程中或这在其它类型的数据库对象中，选择 Database Object 框后边的按钮。
   然后在 Database Object 框中选择这个数据库对象类型，并在 Object Name 框中选择这个数据库类型的对象名字。
6. 如果你希望使用一个显式 SQL 语句选择你所希望的记录集，就选择 SQL Statement 框中的按钮，然后选择 SQL Statement 框中的 SQL 语句。
   要构造这个 SQL 语句，你可以按 SQL Builder 按钮，并使用 SQL Builder。
   在任何时候你都可以把 Command 对象拖到一个 ASP 或 MTML 页面上。这就创建出一个“记录集”对象，这个对象绑定在数据库中一组特定的记录上。

使用数据环境是创建“记录集”控件的推荐方法。一旦使用数据环境把 Command 对象创建出来，修改这个 Command 对象就简单了。你只需要在数据环境中把该对象更新一次就行了。你不必对同一组记录不同 ASP 页面或 HTML 页面上继续重新创建“记录集”控件。

然而，你可以把一个“记录集”从 Toolbox 插入到一个 ASP 或 HTML 页面上，然后使这个“记录集”控件同数据环境中的一个已存在 Command 对象发生关联。

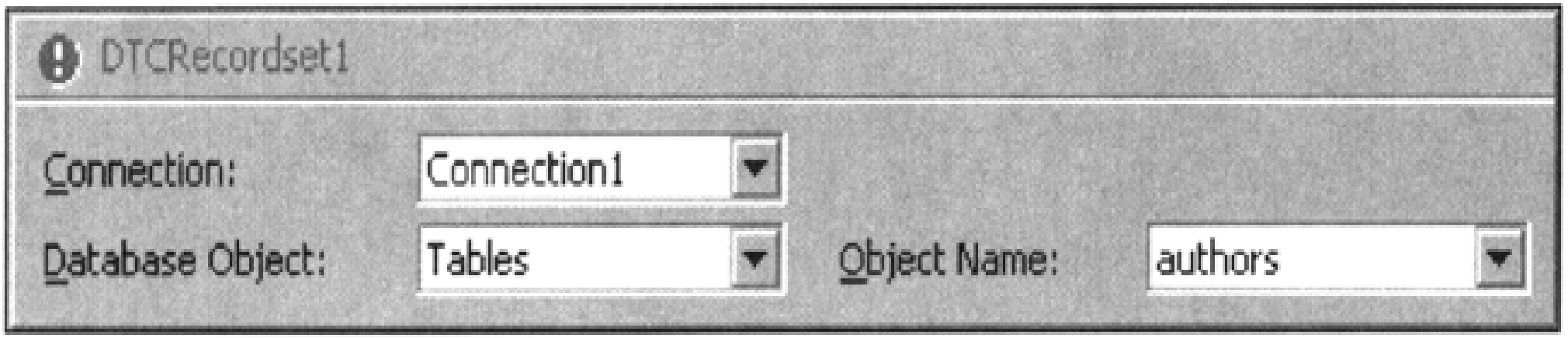
使“记录集”控件同“命令”对象发生关联

1. 打开编辑器中一个 ASP 或 HTML 页面。

2.把 Recordset control 从 toolbox 拖到该页面上。

提示:如果 Recordset 没出现在 Toolbox 中，就用右键单击 Toolbox，选择 Customize Toolbox，然后再添加 Recordset 控件。

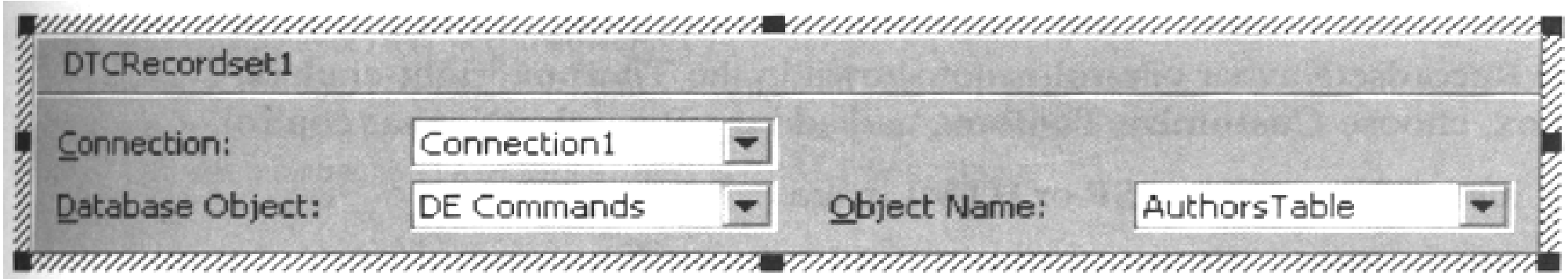
这个“记录集”控件显示 Connection、Database Object 和 Object Name 各字段。这些字段是“记录集”对象的属性。



3.在 ASP 或 HTML 页面上的“记录集”控件中，把 Connection 特性设置成数据连接的名字，而这个数据连接是对应于你希望查看记录的数据库的。这也就是 Command 对象要连接的数据库。

4.把 Database Object 特性设置成 DE Commands。

5.把 Object Name 特性设置成数据环境中 Command 对象的名字。



现在你已经创建了一个记录集并规定了它的记录，你可以在一个 Web 页面上显示这个记录集中的数据了。

19.2 显示 Web 页面上的数据

使用 Visual InterDev 显示数据是很容易的事。你可以连接到一个数据库上，规定要显示的一组记录，然后使用数据绑定设计期间控件显示这组记录，或者你可以使用脚本直接显示数据。你还可以在栅格中显示数据，显示出一个记录集中的多个记录。

使用数据绑定控件显示数据

1. 创建一个数据环境 Command 对象，用来规定数据库中一组记录。详见这章前边的“获取记录”一节。
2. 在编辑器中打开一个 ASP 或 HTML 页面。
3. 把 Command 对象从 DataEnvironment 文件夹拖到该页面上。这就在该页面上创建了一个“记录集”控件。

4.把 Command 对象中的一个或多个字段拖到页面上。

这些字段显示在 DataEnvironment 文件夹中 Command 对象之下。

被拖的每个字段都创建一个数据绑定控件，这个控件将显示该记录集中那个字段的数据。Text 和编号字段创建 Textbox 控件。Yes/No 字段或 True/False（布尔）字段创建 Checkbox 控件。要连接更多内容，参见第十八章“数据库概念”中的“数据环境”一节。

在带有 RecordsetNavbar 控件的 Web 页面上你要显示的记录之间，你可以很容易地提供导航（从一个记录到另一个）功能。

在记录之间提供导航

1.在编辑器中打开一个 ASP 或 HTML 页面。

2.把 RecordsetNavbar 控件从 Toolbox 拖到该页面上。

提示：如果 RecordsetNavbar 控件不在 Toolbox 中出现，用右键单击 Toolbox，然后选择 Customize toolbox，再添加 RecordsetNavbar 控件。

这个控件插入到你的 ASP 或 HTML 页面上。

3.右键单击该控件，再单击 Properties，以便打开其属性页面。

4.把该控件的 Recordset 属性设置成“记录集”控件的名字，这个“记录集”的记录显示在该页面上。

按照默认设置，RecordsetNavbar 控件提供 Move 到 First、Next、Previous 的按钮和 Move 到 Last 的按钮。你可以使用这些按钮在该页面上显示的记录之间进行移动。你还可以通过修改按钮创建的脚本的办法定制这些按钮的功能。

使用脚本进行导航

你可以通过在记录集脚本对象中调用导航的方法在记录集的记录之间进行移动。在移动的时候，移动到达的记录变成当前记录。

重要事项:当你导航到另外一个记录时，当前记录不能自动保存。要保证你已经把数据复制到当前页面上，并在调用一种导航方法之前保存好这个记录。参见第二十章“修改数据”。

使用脚本进行导航

- 调用记录集脚本对象的诸多方法中的一种方法：moveNext，movePrevious，moveFirst，moveLast 和 moveAbsolute 等。
  例如，下边给出你可能为“Next”按钮编写的处理程序：

```
Sub btnNext_onclick
    rsEmployeeList.moveNext
End  Sub
```

如果你希望完成一个以导航为基础的过程，你可以为 onrowenter（行上进入）或 onrowexit（行上离开）事件编写处理程序。例如，你的应用不仅显示字段的内容，你还希望使用单个文本框显示第一个和最后一个名字结合的内容。

要这样做，你可以为 onrowenter 处理程序编写一个处理程序，从记录集中获取数值，把这些数值级联起来，再把它们放到文本框中。这个处理程序看上去像这样：

```
Function rsEmployeeList_onrowenter()
    firstName = rsEmployeeList.fields.getValue("lastName")
    lastName = rsEmployeeList.fields.getValue("firstName")
    txtFullName.value = lastName & ", " & firstName
End function
```

你还可以使用 Grid（栅格）设计期间控件在 Web 页面上创建一个数据栅格。

第二十章　修改数据

除了简单显示数据库中的数据之外，你还可以创建 Web 页面，允许用户修改数据：更新已有记录、添加记录和删除记录。使用“记录集”控件可在你的应用中建立数据修改功能，每个修改功能都创建一个记录集——一种虚拟表格，用户可以在这种表格中进行导航和更新。修改功能产生一个记录集，然后把这个集合传递给基础数据库。

另外，你可以使用记录集显示你页面上的记录，然后调用已存过程，完成更新、插入或删除动作。完成更新的方法同你使用记录集或使用已存过程所使用的方法是一样的。关于把“记录集”控件绑定在已存过程的细节，参见第二十一章“直接访问数据库”中的“使用数据环境执行数据库命令”一节。

不管使用哪种方法，你都必须首先保证可以全面更新数据库。要考虑的因素包括：

- **许可**　许多数据库都要求更新的显式许可。即便你是开发者也要有许可才能进行更新。你必须保证你的 Web 应用的用户也要得到许可。
- **足够的数据**　一般说来，记录集必须包含足够的数据库信息，寻找相应的记录进行修改。这些信息通常是主关键字或独有的检索值。此外，只是作为记录集组成部分的字段才能更新。例如，一个记录集可能包括雇员的 ID 号、名字、姓。尽管雇员表格还有更多字段，在这种情况下，你能够修改的只有这三个字段，或者在添加一个新记录的时候只提供这三

个字段。

- **更新数据源** 如果你借助于记录集进行更新，必须有 Keyset 或 Dynamic 光标类型和一个只读的锁定类型。此外，记录集必须以你可以更新的数据库对象为基础。例如，如果记录集以包含主关键字和独有索引的视图为基础，它就不能更新，而不管你选择的光标类型如何。准确内容取决于你如何产生一个结果集合和你使用的数据库特性。

注意:如果你使用已存过程更新数据库，你记录集的光标类型就不重要了，因为更新不能在记录集中传递。

20.1 从当前记录中获取数值

按照默认设置，在页面运行的时候，记录集脚本对象打开表格或执行在其数据绑定特性中规定的查询，然后创建一个记录集。在该记录集的第一个记录上放上一个指针(pointer)，这个记录就变成当前记录。

要显示数据，你需要从当前记录中获取数值。如果你正以设计期间控件进行工作，而且为它们建立了数据绑定，那么这些控件就自动进行更新，反映数据库数值。

然而，如果你正用非数据绑定控件进行工作，你必须把当前记录中的数值复制到控件中。当前记录的数值可以在记录集的字段集合中得到。

从当前记录获得数值

- 为字段集合调用 getValue 方法，并指明哪个字段是你希望的字段。例如，

下列语句从称为 rsEmployeeList 的记录集对象字段集合中提取“Name”字段的数值：

```
name = rsEmployeeList.fields.getValue(“Name”)
```

提示：从当前记录中获取数值的较好时机是在对应于记录集 onrowenter 事件的处理程序中。

在得到一个数值之后，你可以在一个控件中，例如在一个文本框中或在标签控件中，显示它。例如下列脚本说明你如何在称作 txtName、txtAddress 等的一组文本框脚本对象中显示当前记录的数值。

```
txtName.value = rsEmployeeList.fields.getValue("Name")
txtAddress.value = rsEmployeeList.fields.getValue("Address")
txtCity.value = rsEmployeeList.fields.getValue("City")
areaCode = rsEmployeeList.fields.getValue("AreaCode")
phone = rsEmployeeList.fields.getValue("Phone")
txtPhone.value = areaCode & " " & phone
```

注意：如果这个控件和记录集使用同一种目标脚本编程平台，你只能把记录集的信息复制到一个控件中。更多细节，参见第二十三章“脚本编程概念”中的“脚本编程对象模型”和第二十四章“用设计期间控件和脚本对象编写脚本”中的“用设计期间控件创建窗体”两节。

如果你在可编辑控件中显示数据，例如在文本框中显示数据，用户就可以改变这些数据。要保存他们的变更并把变更写到数据库中，参见下一节“更新记录”。

20.2 更新记录

“记录集”(Recordset)控件包括一些特性，使你很容易允许用户在记录集中更新当前记录。例如，如果你在页面上使用一个 RecordsetNavbar 控件，就可以设置一个选项，在用户导航到其它记录上时以用户的变更自动更新当前记录。

不是所有记录集都允许变更(向数据库中写入)。“记录集”对象光标类型必须设置为 Keyset 或 Dynamic。你还必须拥有更新数据库的许可。最后，查询可以产生记录集，但是不包含允许更新的足够信息。有关细节，参见本章开始部分。

开启自动导航更新

- 把 RecordsetNavbar 控件的 updateOnMove 属性设置为 True（真）。

  注释:把这个选项设置为 True（真），便使得记录集在用户每次导航时得以更新，即便不作出变更也是一样。如果你预料到用户很少进行变更，你可能希望使用系统开销较少的一种策略。

如果你正用数据绑定设计期间控件进行工作，这些控件便在作出更新以前把它们的数值复制到当前记录中。但是，如果你工作时使用的控件不是数据绑定设计期间控件，那么你必须在作出更新以前手动更新当前记录。

在当前记录中设置数值

- 为字段集合设置 setValue 方法，指明那个字段是你希望的字段。规定要

更新的字段和数值。

提示:从当前记录获取数值的较好时机是在记录集 onbeforeupdate 事件的处理程序中。

例如，下边语句把当前记录的 Name 字段设置为称为 Name 文本框的数值。

```
rsEmployeeList.fields.setValue("Name",txtName.vlaue)
```

注释:如果控件和记录集使用同一个目标脚本编程平台的话,你只能从一个控件向记录集中复制。更多细节，参见第二十三章“脚本编程概念”中的“脚本编程对象模型”和第二十四章“用设计期间控件和脚本对象编写脚本”中的“用设计期间控件创建格式”两节。

你只能在脚本中更新记录。基本过程是调用把当前记录写入数据库的方法。

更新脚本中的记录

1. 要保证在你的页面上有一个“记录集”（Recordset）控件。有关细节，参见第十九章“查看数据”中的“获取记录”，注意“记录集”控件的名字。
2. 如果你在页面上拥有一些控件，而这些控件不是数据绑定设计期间控件，就要按照前边描述的过程把它们的数值复制到当前记录中。
3. 在当前记录中完成你所需要的所有数值设置之后，调用 updateRecord 方法。

下例给出一个 Save 单击事件的处理程序，这个处理程序把当前记录保存到数据库中。

```
Sub btnSave_onclick
```

```
    rsEmployeeList.updateRecord
End sub
```

下例给出同一个过程，但是把数值从非数据绑定控件复制到当前记录上，然后再保存这些数值。

```
Sub btnSave_onclick
    ' Copying data to the current record is required only for
    ' controls that are not data-bound design-time controls
    rsEmployeeList.fields.setValue("Name", txtName.value)
    rsEmployeeList.fields.setValue("Insured", chkInsured.value)
    rsEmployeeList.fields.setValue("LastUpdate", date)
    rsEmployeeList.updateRecord
End sub
```

为了帮助你在更新记录集的时候捕获差错，你可以为记录集对象的onbeforeupdate 和 onafterupdate 事件编写处理程序。典型情况是你为onbeforeupdate 事件编写一个处理程序以保证在当前记录中的数据是正确的。你可以使用 onafterupdate 事件来确定更新是否成功。

关于为设计期间控件编写事件处理程序的内容，参见第二十四章“用设计期间控件和脚本对象编写脚本”中的“为脚本对象编写脚本”一节。

20.3 添加记录

当你使用脚本中的记录集对象进行工作时，你可以有两种方式向数据库中

添加记录。第一种方式分两步操作。首先，你启动一个新的空白记录。用户可以填写这个记录。在记录完成的时候，用户可以单击 Save 按钮（或类似按钮），把当前记录写入到数据库中，就像一个已经存在的记录被更新一样。当你用一种窗体进行工作，而用户可以使用这种窗体编辑已经存在的记录或添加新记录，在这种情况下上述策略是很有用的。

另外一种情况，你可以创建一个新记录，并把它移入单个操作之中。如果你的目标脚本编程平台是服务器，立即添加这种模式会更有效，因为这样可以避免服务器启动新记录的往返过程。在你不使用某种窗体，而是在脚本中创建一些新记录时，这种策略特别有用，尤其当你把几个记录一次添加到数据库中时。

你不可能总是可以创建新记录。你需要对你的记录集有正确设置，还要对数据库有适当的许可以及唯一标识记录的足够信息。更详细信息，参见本章开始部分。

先预置后更新

在用某种窗体进行工作的时候，你经常使用预置和更新的策略。这种窗体通常有一个 New 按钮和一个 Save 按钮。使用预置和更新策略，你可以使用同一个 Save 按钮更新已经存在并已经编辑过的记录，还可用于插入新记录。

在记录集中预置一个新记录，你要使用“记录集”对象的 addRecord 方法。如果你用数据绑定设计期间控件进行工作，这个记录集便在你预置一个新记录的时候自动清除控件。当你保存这个记录时，用户键入的数值便自动复制到当前记录中，然后再把记录写入数据库。

如果你工作用的控件不是数据绑定设计期间控件，则你必须完成一些这方面的工作。在预置一个记录之后，你必须手动清除控件，这样用户便能录入新数据。然后在你的用户单击 Save 按钮或类似按钮的时候，你必须保证把数据从控件复制到当前记录中，然后才把这些记录写入数据库。

预置并添加记录

1. 保证在你的页面上有一个“记录集”（Recordset）控件。详见第十九章“查看数据”中的“获取记录”，注意“记录集”控件的名字。
2. 在脚本中，调用记录集脚本对象的 addRecord 方法，在这个记录集中准备一个新的空白记录。例如，用于 New 按钮的下列事件处理程序调用 addRecord 方法：

```
Sub btnNew_onclick
    rsEmployeeList.addRecord
End sub
```

如果你的页面包含非数据绑定控件，便在预置新页面之后清除这些控件，如下例所示：

```
Sub btnNew_onclick
    rsEmployeeList.addRecord
    txtName.value = ""
    chkInsured.value = false
End sub
```

注意:如果控件和记录集都使用同样的目标脚本编程平台，在同一个时机你只能使用控件和记录集进行工作。详见第二十三章“脚本编程概念”中的“脚本编程对象模型”和第二十四章“用设计期间控件和脚本对象编写脚本”中的“用设计期间控件创建窗体”两节

在用户单击 New 按钮时，用户便把信息键入到新记录中。页面上的“记录集”控件设置一个旗标〔flag〕，表示这是一个新记录。在用户单击 Save 按钮时，“记录集”控件便自动创建这个新记录，以用户的数据更新这个记录，再把记录写入数据库。

新记录填写之后，同已有记录一样的方式（以记录集对象的 updateRecord 方法）进行保存。详见本章前边的“更新记录”一节。

即 刻 添 加

如果你正在脚本中创建记录，你会经常立即把记录添加到数据库中。把记录立即添加到数据库中，同先预置一个记录然后保存是类似的。然而，因为添加和更新功能是在一个步骤中完成的，因而对服务器只要求往返一次。

在你创建一个即刻记录时，你必须把各字段的信息收集在一起，然后在你添加新记录时，把这些信息作为参数传递给记录。

添加即刻记录

1. 把要写入数据库的信息收集在一起。你可以使用页面上的控件做这件事情，数值要计算出来，或通过其它方式得到。
2. 创建两个数列，一个数列包含字段的名字，另一个数列包含数值。例如，

下列脚本创建一个字段数列，跟踪事务处理（transID、transData 和 transTime）和每个字段的数值。

```
Dim fieldsArray(2)
Dim valuesArray(2)
fieldsArray(0) = "transID"
fieldsArray(1) = "transDate"
fieldsArray(2) = "transTime"
valuesArray(0) = getTransactionCounter()
valuesArray(1) = Date
valuesArray(3) = Time
```

3.调用“记录集”对象的 addImmediate 方法添加记录，并把记录传递给你刚刚创建的两个数列，如下例所示：

```
rsTransactionLog.addImmediate( feldsArray. ValuesArray )
```

关于为设计期间控件编写事件处理程序，参见第二十四章“用设计期间控件和脚本对象编写脚本”中的“为脚本对象编写脚本”一节。

20.4 删除记录

要从数据库中删除一个记录，你可以在页面上添加一个控件，例如 Delete

（删除）按钮。这个按钮调用记录集的 deleterecord 方法，删除当前的记录。

你不是总可以删除记录。你需要对你记录集正确设置、数据库中相应权限和唯一标识记录的足够信息。有关细节，参见本章开头部分。

在脚本中删除记录

1. 保证在你的页面上有一个“记录集”控件。详见第十九章“查看数据”中的“获取记录”一节。
2. 在脚本中，调用记录集脚本对象的 deleterecord 方法，就像下例中称为 btnDelete 按钮的处理程序一样：

```
Sub btnDelete_onclick
    rsEmployeeList.deleteRecord
End sub
```

按照默认设置，上边说明的脚本使得页面显示的记录是刚刚被删除的记录后边紧跟着的记录。如果你愿意让页面显示一个不同记录，则可以在返回语句的前边插入一个导航方法。下边的例子说明如何移到前边的记录。

```
rsEmployeeList.deleteRecord;
rsEmployeeList.movePrevious;
```

关于为设计期间控件编写事件处理程序的更多内容，参见第二十四章“用设计期间控件和脚本对象编写脚本”中的“为脚本对象编写脚本”一节。

第二十一章　直接访问数据库

许多数据库应用不是只依赖绑定在数据库中表格和视图上的控件。这些应用借助于具有一些优点的已存过程或查询完成它们的数据访问。

使用已存过程可以限制用户作出的改变，也可以增强验证和其它要求。如果你使用查询进行工作，你可以按照你应用的要求动态地改变查询。

在这一章，你将学习如何创建 Microsoft Visual InterDev 应用，以便直接用已存过程和查询进行工作。

21.1　调试已存过程和触发器

如果你用 Microsoft visual Studio 的企业版进行工作，Microsoft Visual Interdev 包括 SQL 调试器，你可以使用这种调试器调试 SQL 服务器已存过程和触发器，调试方式同你调试其它类型的脚本或程序非常相似。但是在你安装 SQL 调试和运行调试器方面，还是有些不同。

安装 SQL Debugging(调试)

SQL Debugging（调试）要求如下：

- 你必须有 Visual Studio 的企业版。
- 你必须运行 SQL Server 6.5 并有 Service Pack 2。
- SQL Server 必须运行微软 Windows NT 4.0 或更高的版本。

- 你的工作站必须运行 Windows 95 或 Windows NT 4.0 或更高的版本。

要使用 SQL Debugging（调试），你必须适当配置你的服务器和工作站。你应当：

- 保证已经在你的 SQL 服务器上安装了 SQL Debugging 部件。
- 建立一个 Windows NT 用户，这个用户在 SQL Server 运行的服务器计算机上有管理员特权。
- 在服务器上为 SQL 调试配置分布式 COM(DCOM)。
- 保证在客户机上配置的 DCOM 支持 SQL Debugging（仅仅是 Windows 95 工作站）。

安装 SQL Server Debugging 部件

SQL 调试要求的部件要安装到你 SQL 服务器上。这些部件是 Visual Studio，企业版的组成部分。

安装 SQL 调试部件

1. 在安装 SQL 服务器的计算机上，启动 Visual studio Enterprise Edition（企业版）安装程序。
2. 安装向导稍有不同的选择，取决于在计算机上你是否已安装了服务器部件。
   - 如果已经安装了服务器部件，在 Add/Remove Options 之下，选择 Server Applications and Tools。
   - 如果没安装服务器部件，开始安装向导，直到你到达提供给你

Enterprise Setup Options 页面为止。选择 Server Applications。

3. 开始安装向导软件，直到你到达提供给 Enterprise Setup Options 页面为止。选择 Server Applications。

4. 在下一页上，选择 Launch BackOffice Installation Wizard，然后选择 Install。

5. 在显示 BackOffice Business Solution 向导的时候，选择 Custorm，然后选择 Next。

6. 接着进行，直到你看见为你提供安装的部件列表页面为止。除了下列部件之外，所有部件都不进行检查：

- SQL Server Debugging
- MS Data Access Components
- Visual InterDev Server

7. 继续安装。

设置调试用户

使用 SQL Debugging，你必须可以提供 Windows NT 用户的名字和口令，这个用户在 SQL Server 运行的服务器计算机上拥有管理员特权。

为 SQL 调试设置用户

1. 在服务器的 Windows 控制面板上，选择 Settings，然后选择 Services。
2. 选择 MSSQLServer，然后选择 Startup。
3. 检查 Log On As 设置。如果该选项置为 System Account，把它改变为 This

Access，为拥有管理员特权的用户键入有效域名和帐号（窗体是：域名/帐号），然后键入口令。

4. 如果你已经改变了设置，则重新启动 SQL 服务器。

为 SQL 调试安装 DCOM

SQL 调试使用分布式 COM（DCOM）在你的客户计算机和服务器之间进行通信。因此你必须配置 DCOM，以允许远程用户连到服务器上调试器的一个进程上。

根据默认设置，正确的 DCOM 设置在 SQL Server 安装到服务器上的时候就配置好了。然而，基于对运行 SQL Server 的计算机安全的考虑，你可能希望限制对调试器的访问。使用下列一般过程，在 SQL 服务器计算机上安装 DCOM。

1. 从服务器的 Windows Start 菜单上，选择 Run，然后在 Open 框中安装提示键入 Dcomcnfg.exe。
2. 在 Distributed COM Configuration Properties 窗口中，选择 Default Security 选项卡。在 Default Access Permissions 下，选择 Edit Default。
3. 如果 Everyone 组还没有得到许可，选择 Add，然后为拥有管理员特权的用户加上域名和用户帐号（窗体是：域名\帐号）。
4. 加上帐号之后，检查 SYSTEM。如果在列表中没有，就把 SYSTEM 加上：在 Add Names and Groups 对话框中的用户列表中选择它。
5. 如果你对这个过程所描述的任何设置做了改变，就要重新启动 SQL Server。

注意：如果你把你的帐号加到远程服务器上，但是该远程服务器上的当前帐号却没加上，那么这个帐号不能进行调试，尽管拥有那个帐号名字的

用户正在服务器计算机上运行 Visual InterDev。

运行 SQL Debugging

与调试其它类型进程不同，你不能调试正在运行的已存过程或触发器。你要在编辑器中打开要调试的过程，然后在这里调试它。

调试已存过程

- 在 Data View 窗口中，用右键单击要调试的已存过程，然后选择 Debug。

  编辑器窗口打开，里面有这个已存过程文本和 Debug 菜单中可以使用的调试命令。

在 Debug 方式中打开编辑器窗口之后，你就可以像使用通常命令一样使用调试器命令。例如，你可以设置断点，一步步通过该过程。你可以查看变量值，并把参数传递给 Locals 窗口。你还可以把表达式拖到 Watch（监视）窗口中，对你通过过程的步骤或运行过程情况进行追踪。SQL PRINT 语句的执行结果显示在 Output（输出）窗口中。

这种 SQL 调试器同对脚本使用的调试器有以下差别：

- Auto 和 Immediate 窗口在你调试已存过程时功能失效。尽管你可以显示这两种窗口，但是 Auto 窗口将是空的，而 Immediate 窗口不允许你放入表达式。
- 你不能使用 Set Next Statement 命令改变执行顺序。

如果是用简单 SELECT 语句(每次只能返回一个值)进行工作，返回值在 Locals

窗口内你可查看的变量中。然而，如果 SELECT 返回一个结果集合，这个集合不显示在调试器中。但是你可以在 Output 窗口中查看这个结果集合。

21.2 执行参量化查询

在许多应用中，你希望用数据集合进行工作。这些数据集合是用在应用中支持的条件创建的。例如，你的应用可以显示有关一个部门中所有雇员的报告。你可以组建一种窗体，提示用户键入部门的名字，然后根据用户键入的值执行一次查询。这种类型的查询叫做参量化查询（parameterized query）。

对于参量化查询，你使用一个“记录集”设计期间控件，就像你使用一个表格或其它数据库对象的控件一样。其区别是这种“记录集”控件是绑定在一个已存过程上或一个 SQL 语句上，而不是绑定在一个表格上。

你可以直接把控件绑定在过程或语句上，或者把控件绑定在一个数据命令上。这个数据命令指向这些类型对象中的一个。关于数据命令的细节，参见第十九章“阅览数据”中的“获取记录”一节。

创建参量化查询

1.把“记录集”设计期间控件拖到页面上。有关细节，参见第十九章“查看数据”。

2.右键单击控件，选择 Properties。在 General 选项卡中，设置控件的名字和连接。

3.在 Source of data 下，规定绑定：

- 如果是绑定到一个数据命令上，选择 Database object，然后选择 DE Commmands。从 Object name 列表中，选择要使用的数据命令名字。
- 如果直接绑定到一个已存过程上，选择 Database object,然后选择 Stored Procedure。从 Object name 列表中选择要使用的已存过程名字。
- 对于 SQL 查询，选择 SQL statement，然后键入 SQL 文字。选择 SQL builder，启动 Query Designer。在 SQL 语句中，使用一个问号，指示你查询中的参数。例如，下边 SQL 语句创建一个查询，在这个查询中，部门名字是参数：

  SELECT * FROM employee WHERE department = ?

  注意:不要使用命名参数。

4. 选择 Parameters 选项卡。在 Values for paraneters 之下，你会看见为查询规定的参数列表。对于 Type 值为 In（传递给已存过程或查询的一个参数）的每个参数，指定一个传递给已存过程的值。这些值可以是：
   - **Literals**（文字） 在单括号中键入字符值和没有问号的数字值。
   - **Variables**（变量） 键入服务器代码中定义的变量名字，服务器代码中有你希望传递的值。
   - **Object references**（对象引用子） 键入一个对象引用子和一个特性值，例如 Textbox1.value。对象必须是服务器脚本中可以得到的一种对象。要保证正确使用大写字母，因为表达式将作为一个 JavaScript 表达式来对待。
   - **Expressions**（表达式） 键入文字、变量、对象引用子和功能调用

的任何结合形式。这个表达式是按照 JavaScript 表达式来对待的，因而使用 JavaScript 约定，包括用于文字字母的单括号和用于级联用的加号(+)。

你必须保证在运行查询的时候，参数可以成功地进行计算。作为默认设置，页面第一次下载时，“记录集”设计期间控件将执行查询。在这种情况下，参数值不能是收集的值，或者只在页面显示之后赋给的值。如果你传递的变量值作为查询参数，那么你可以在记录集打开之前，使用两个要进行处理的事件：

- 对于页面使用 onenter 事件。
- 对于记录集使用 onbeforeopen 事件。

此外，你还可以规定，在页面打开时“记录集”控件不自动打开记录集。如果你对于数据录入窗体和结果窗体使用相同页面，这种策略是很有用的。

防止“记录集”控件自动打开记录集

- 右键单击“记录集”(Recordset)控件并选择 Properies。在 Implementation 选项卡中，清除 Automatically open the Recordset 选项。

你可以启动显示没有记录集的页面，并使用一种窗体提示键入一个数值。当用户填写这个窗体并单击一个按钮时，你可以打开这个记录集，并把一个变量值或控件传递给它。

打开记录集

- 调用记录集脚本对象的开启方法。

例如你可能创建一种窗体，带有一个文本框设计期间控件，还带有一个按

钮设计期间控件。用户可以用搜索的值填充这个文本框，然后单击这个按钮。

在“记录集”控件的“特性页面”窗口的参数选项卡中，规定文本框脚本对象的值作为使用下述表达式的参数，把文本框的名字提交给 textbox1:

```
textbox1.value
```

于是按钮的 onclick 事件处理程序简单调用该记录集脚本对象开启方法，如下例所示：

```
Function Button1_ondick
    Recordset1.open()
End Function
```

执行动态查询

如果记录集脚本对象是绑定在一个 SQL 查询上，你可以在运行时改变查询并重新执行它。如果根据用户提供的信息运行一次查询，这是特别有用的。

SQL 命令的当前文本，可以通过调用记录集脚本对象的 getSQLText 方法来使用。你可以通过调用其 getSQLText 方法来设置 SQL 文本。在设置 SQL 文本之后，你可以调用这个记录集脚本对象的开启方法来重复执行查询。

下边的例子说明在记录集中你如何动态改变记录的排列顺序。例子中取得一个 SQL 文本，再把一个新 ORDER BY 分句附在命令的末尾，再一次更新这个记录集，然后运行查询。在查询运行之后，处理程序恢复成原来的查询文本。

```
Sub btnByFName_onclick()
    oldSQL= rsEmployeeList.getSQLText
    newSQL = oldSQL & " ORDER BY firstName"
    rsEmployeeList.setSQLText(newSQL)
    rsEmployeeList.Open    ' Executes the new query
    rsEmployeeList.moveFirst
    ' Restores the original query
    rsEmployeeList.setSQLText(oldSQL)
End Sub
```

你可以确定查询什么时候完成，办法是编写一个 ondatasetcomplete 事件处理程序。例如，你可以等待，直到查询完成，然后显示查询的数据，并开启页面上的控件。完成这些工作的处理程序如下所示：

```
Sub rsEmployeeList_ondatasetcomplete()
   ' Enables controls
   btnNext.disabled = False
   btnPrevious.disabled = False
   btnSave.disabled = False
   btnNew.disabled = False
   ' Refreshes the fields on the page
   txtLastName.value = rsEmployeeList.fields.getValue("au_lname")
   txtFirstName.value = rsEmployeeList.fields.getValue("au_fname")
End sub
```

使用数据环境执行数据库命令

向你的 Web 页面上添加数据库访问机制的最方更的方式是使用“记录集”控件和数据绑定控件。这些工具很容易加到你的页面上，而且还具有一种对象丰富的模型，提供强有力的访问数据库的功能。有关细节，参见 Visual InterDev 在线文档中“设计期间控件”。

然而，在某些情况下，你可能希望创建更直接执行数据库命令的脚本。这样做可使你的页面比使用“记录集”控件时要小。如果创建了自己的用户界面，而且不希望依赖设计期间控件的话，你还可能希望脚本直接访问数据库。

你可以使用脚本在整个服务器中访问数据库，或者直接从客户机脚本上访问数据库。关于服务器和客户机访问数据的讨论，参见第十八章“数据库概念”中的“数据绑定”。

对于以脚本服务器为基础的数据库访问，在连接信息和数据库命令方面，你可依赖数据环境。这个题目将在后边讨论。对于以客户为基础的数据库访问，可使用“远程数据服务”（RDS）。关于 RDS 的内容，参见 Microsoft RDS Web 站点：http://www.microsoft.com/ data/rds/。

编写 DE 对象脚本

对于以脚本服务器为基础的数据库访问，你可使用一种专门的对象——DE 对象，进行工作。这种对象具有一种对象模型，用于执行命令并管理它们的结果。DE 对象模型是 ActiveX Data Object(ActiveX 数据对象)模型的一种容易使用的版本。

在编写数据库访问脚本之前，你必须把一条数据连接加到你项目数据环境中。在你做这件事的时候，Visual InterDev 把脚本加到 Global.asa 文件上，创建一个包含连接信息的 DE 对象。当你在页面上编写数据访问脚本时，你可以引用这个 DE 对象，Visual InterDev 会自动得知如何连接到数据库上。

在添加数据连接之后，你再把一个或多个数据命令加到这条数据连接上。每个数据命令都为你提供对数据库对象访问的机制。这种访问机制包括一个表格或视图、一个 SQL 命令或一个已存过程。

执行命令

你可以在脚本中引用相应命令来执行数据库命令。

执行数据库命令

1. 要保证你希望执行的命令定义为数据环境中的一条数据命令。有关细节，参见第十九章“查看数据”中的“获取记录”一节。
2. 在你的 Web 页面上，使用服务器脚本去引用 DE 对象。如果你已经开启了该页面的脚本对象模型，则使用下述脚本：

```
<%
    thisPage.createDE()
%>
```

如果你不使用这种脚本对象模型，则使用下述脚本：

```
<%
set DE = Server CreateObject(" DERuntime.DERuntime " )
DE.Init(Application(“DE”)
%)
```

3. 在服务器脚本中，通过引用命令名字的办法执行你希望执行的命令。使用的语法是：

De.*commandObjectName*

例如，要执行与称作 Authors 命令对象有关的 SQL 查询，就使用这样的脚本：

```
<%DE.Authors%>
```

4. 如果该命令要用参数（例如引用一个已存过程或参数化查询的命令），你就在调用这个命令的时候传递参数，使用的语法是：

DE.commandObjectName(*parameter1*,*parameter2*,[...])

你可以按照变量、文字（在你编写脚本使用的语言中，字符串应当使用相应的问号进行传递）或其它任何有效表达式传递参数。
例如，要执行需要两个参数的已存过程，就要使用下述脚本引用自己的相应命令对象：

```
<%DE.updateAuthors( " A101 " ,txtAuthorLname.100.00, " London " )%>
```

5. 如果该命令返回一个值，在你执行该命令的时候，用下述脚本把这个值分配给一个变量：

```
<%iRetValue = DE.updateAuthors(" A101 " ,txtLname,100.00)%>
```

用结果集合进行工作

所有命令都返回一个集合，尽管在有些情况中（例如使用更新查询）结果集合是空的。但是，如果你的命令返回一个有用的结果集合，你就可以在结果集合中导航，提取它的内容，显示在页面上。

要提供对结果集合的访问机制，DE 对象就要动态创建一个以该命令对象为基础的结果集合对象，并使用下述语法命名：

DE.rs*commandObjectName*

在产生一个结果集合的时候，应当包含一个指示符，指示哪个记录是当前记录。你完成的任何操作，例如显示数据、导航或更新，在完成过程中都与当前记录有关。要使用不同记录进行工作，你必须首先导航到那个记录。

下边描述的过程允许你用一个结果集合进行工作，而这个结果集合完全用在一个页面上。例如，你可以在这里使用过程创建一个页面，这个页面列出一个页面的一个表格中结果集合的所有记录。

提取结果集合的内容

1. 在创建一个结果集合之后，设置一个变量，指向 DE 结果集合对象。这个 DE 结果集合对象是按照你的命令对象命名的，但是具有“rs”前缀。例如，下述两个语句创建一个结果集合，然后设置变量以指向它：

```
<%
DE.Authors
Set rs = DE.rsAuthors
%>
```

2. 使用类似下述语法，从结果集合的 Field 集合中提取一个一个值：

```
<%
DE.Authors
Set rs = DE.rsAuthors
1name = rs.Fields(“au_1name”)
%>
```

要移到不同的记录，你就要使用记录集上的导航方法。

在结果记录中导航

1.调用结果集合对象的 moveNext、movePrevirous、moveFirst 或 moveLast 方法。

2.要确定你是在结果集合的开始还是结束位置，要测试 EOF 或 BOF 属性。

下边的脚本给出一个完整的例子，说明如何编写一个 DE 对象脚本去到 Authors 表格中获取信息。这个脚本打开一个基于称作“Authors”的数据命令对象的记录集。然后从结果集合的开头导航到末尾，显示每个记录中 au_lname 字段的内容。

```
<%
Set DE = Server.CreateObject("DERuntime.DERuntime")
DE.Init(Application("DE"))
DE.Authors
set rs = DE.rsAuthors
rs.moveFirst
Do While Not rs.EOF
%>
   Next name = <%=rs.Fields("au_lname")%><br>
   <%
   rs.moveNext
Loop
%>
```

记录分页

如果你希望在一个页面上显示结果集合中的单个记录，就要对记录之间的页面提供导航控制。这种情况比较复杂，如第十八章“数据库概念”中的“数据绑定”所描述的那样。

如果这就是你的目的，你就应当找到一种更方便的方法，即使用一个“记录集”（Recordset）、数据绑定控件和一个 RecordsetNavbar 控件的混合方法。

有关细节，参见第三章“数据库基础”中的“显示记录”一节。

有关设计分页的其它信息，参见 Microsoft Visual Studio Web 站点 http://www.microsoft.com/vstudio/。

第二十二章　管理数据库项目

如果你在 Microsoft Visual InterDev 中管理企业数据库，数据库项目会为你提供一个很重要的有利条件：就是对你的数据库、数据库结构和其中的数据进行全面控制。数据库项目通过下述三个方面使这个有利条件成为可能是：SQL 脚本、可以用来建立数据库的文和植入数据。在数据库项目中包含的这些文件和文件夹都置于源控制之下，以便进行更大的控制。

在你创建一个数据库项目并加上一个数据连接的时候，这个数据库显示在 Data View（数据视窗）窗口中，因而你可以使用微软 Visual Database Tools（Database Designer 和 Query Designer）在数据库中操纵这些项目。

你可以在一个 Web 项目和一个数据库项目中使用同一个数据连接，以便对你希望在 Web 页面上显示数据的数据库保持控制。

除了对 Web 项目之外，使用数据库项目还为你带来下列好处：

- 在数据库项目中不要求数据连接，直到你希望对数据库运行一个 SQL 脚本为止。此外，你还可以在数据库项目中对数据连接重新命名。
- 你可以使用源控件，保证数据库及其结构都在管理员控制之下。
- 你可以很容易地把数据库配置到其它项目上。

22.1 植入数据库项目

一个数据库可以包含一个或多个到数据库的连接。每个数据库除了数据库数据之外，还都包含多个数据库对象，如表格、查询、视图、关系和索引。

你可以用下列三种方式创建和修改数据库对象和数据：

- 使用你加到数据库项目上的 SQL 脚本。
- 直接在 Data View（数据视图）中编辑数据库项目。
- 使用“可视数据库工具”（Database Designer 和 Query Designer）。

使用 SQL 脚本为你对数据库、数据库项目和数据库数据提供全面控制。SQL 脚本是包含 SQL 语句的文件，这些 SQL 语句是用来创建和修改数据库和数据库对象的。

使用 SQL 脚本文件植入数据库项目

1.把 SQL 数据库文件加到你的数据库项目中。

2.针对你已经建立一个数据连接的数据库执行 SQL 脚本文件。

关于如何做这些事情以及 SQL 脚本文件包含什么，参见 Visual InterDev 包括的在线文档 Visual Database Tools 中的“Working with SQL Script”。

提示:Visual InterDev 提供多种样板，你可以使用这些样板把 SQL 脚本文件加到数据库项目上。当你用右键单击数据库项目或它的多条数据连接中一条，并选择 Add SQL Script 命令的时候，下列样板便准备好供你使用：New SQL Script、New Stored Procedure Script、New Table Script、New trigger Script 和 New View Script。

New SQL Script 样板把通用 SQL 脚本文件加到你的数据库项目上，你可以使用这个样板创建或修改任何类型的数据库对象。其它样板创建指定为 SQL Server 数据库项目的 SQL 脚本。

你还可以在 Data View 窗口中用右键单击 SQL Server 的一个数据库项目，并选择 Copy SQL Script 命令。然后你就可以把这个 SQL 脚本贴到 SQL 编辑器中，或直接贴到数据库项目上。这就创建了一个 SQL 脚本，在对数据库运行这个脚本时，就创建你所选择的数据库项目。

你还可以在 Data View 中选择多个数据库项目，并使用 Copy SQL Script 命令创建一个 SQL 脚本，你可以把这个脚本加到一个数据库项目上，还可以使用这个脚本创建所有选中的数据库项目。除了 SQL Server 数据库项目之外，这个 Copy SQL Script 命令还用 Oracle 视图、触发器、已存过程和若干功能进行工作，但是不用 Oracle 表格和类似的表格。

当你把数据连接加到数据库项目上时，数据连接和数据库项目显示在 Data View（数据视窗）中。你可以直接编辑数据库对象。这是创建和修改数据库项目和数据的最容易的方式，但是你对数据库的控制不像使用 SQL 脚本那么多。

使用数据视图植入数据库项目

1.选择你希望修改的数据库项目，或选择一个命令，创建一个新项目。

2.在项目中直接编辑数据，或使用 SQL 编辑器修改项目中的 SQL 语句。

Query Designer 和 Database Designer 允许你创建和修改查询、表格和与表格有关的项目，如索引、约束和关系。

使用 Database Designer 植入数据项目

1. 在 Data View 中，选择包含数据库的数据连接。
2. 选择你希望修改的数据库项目（表格或数据库图表），从快捷菜单中选择 Design 命令。
   或
   在快捷菜单中创建一个新表格或新数据库图表。
3. 修改并保存数据库项目。

使用 Query Designer 植入数据库项目

1. 在 Project Explorer 中，选择包含数据库的数据连接。
2. 选择你希望修改的查询，在快捷菜单中选择 Design 命令
   或
   在快捷菜单中选择 New Query 命令。
3. 修改并保存查询。

22.2 同时使用数据库项目和 Web 项目

数据库项目和 Web 项目出现在一个解决方案的不同节点上，这是因为这两个项目是不同类型的项目。使用 Web 项目创建动态 Web 页面并显示信息（包括数据库中的数据）。使用数据库项目创建、管理并配置企业数据库。

然而，如果你希望维持对数据库的控制并配置一个数据库，而这个数据库上的数据是你希望在 Web 页面上显示的，那么你可以同时使用数据库项目和 Web 项目。你可以在 Web 项目中创建一个数据库连接，然后把这个连接加到数据库项目上。这个数据库项目中的 SQL 脚本控制数据库中的结构和数据。

使用数据库项目中的 Web 项目数据连接

1.创建一个 Web 项目。有关细节，参见第一章“Web 项目管理”中的“创建 Web 项目”。

2.把数据连接加到 Web 项目上。有关细节，参见第三章“数据库基础”中的“连接数据库”。

3.创建数据库项目。有关细节，参见 Visual InterDev 包含的在线文档 Visual Database Tools 中的“Creating a Database Project”。

4.把数据连接从 Web 项目加到数据库项目上。

5.在数据库项目中创建管理数据库及其项目的 SQL 脚本。

6.如果你希望，把数据库项目配置到其它解决方案中。

注意:不要在 Web 项目和数据库项目之间拖放数据库项目。

22.3 把源控件加到数据库项目上

你可以把你的数据库项目文件加到源控件上，来维持对数据库及其结构的控制。数据库项目由文件夹和 SQL 脚本文件组成。一旦你把这些文件夹和文件加到源控件上，你就可管理这些文件夹和文件，就像在 Web 项目中对待文件夹

和文件一样。关于如何在文件夹和文件中使用源控件的更多内容，参见第八章“同多个开发者一起工作”。

注意：当你把文件夹和文件加到数据库项目上时，这些文件夹和文件实际加到你的硬盘上。此外，当你把一个数据库项目放到源控件之下的时候，在数据库项目中的文件和文件夹之间与源控件项目中的文件和文件夹之间，有一对一的关系。

你还可以对数据库项目中的数据连接重新命名。因为连接出现在硬盘中的文件夹内，又在源控件项目中，因而这些文件夹也随之重命名。

把数据库项目加到源控件上

1. 在 Project Explorer 中，选择数据库项目。
2. 用右键单击数据库项目名字，从快捷菜单中选择 Add to Source Control。现在数据库项目及其文件都在源控件之下。

如果你后来希望把它们从源控件中去掉，则可以使用快捷菜单上 Remote from Source Control 命令。

在 Data View 窗口中，你可以把已存过程放到源控件之下。这就为你对这些数据库项目提供附加控制。在数据库项目中，已存过程可以保存在 SQL 脚本中；对于 Web 项目而言，这些项目作为已存过程出现在 Data View 中。要了解更多内容，参见 Visual InterDev 包含的在线文档 Visual Database Tools 中的“Adding a Stored Procedure to Source Control in Data View”。

如果你在本地工作，Web 服务器上或数据库服务器上的数据库是现成的。然而，如果你离线工作，在你的本地计算机上必须有数据库。当你在 Visual

InterDev 中离线模式下，则没有办法归并你作出的任何改变。