

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 (二)





[返回总目录](#)

第一篇 基础

第一章	Web 项目管理	5
1.1	建立 Web 项目	6
1.2	添加文件	10
1.3	删除一个 Web 项目	11
第二章	Web 基础	13
2.1	创建页面	13
2.2	保存页面	15
2.3	预览页面	16
第三章	数据库基础	18
3.1	连接数据库	19
3.2	查询数据库	22
3.3	显示记录	24
3.4	建立事件驱动窗体	26
第四章	编辑基础	32

4.1	编辑 HTML	32
4.2	为页面增添部件	37
4.3	增添脚本	40
4.4	选用其它编辑器	41
第五章	演 练	43
5.1	创建主页	43
5.2	调试脚本	56
5.3	同多个开发者一同工作	69
5.4	部署 Web 应用程序	77
5.5	简化数据录入页面	85

第一篇 基础

“基础”篇中的各章提供 Microsoft Visual InterDev 的重要背景和一些有关信息。为完成通常 Web 应用任务，每一章都列出明确的过程步骤。

第一章 Web 项目管理

一个 Web 项目包括在 Microsoft Visual Studio 中创建并公布单个 Web 应用所需要的文件和信息。这一章教你怎样创建一个 Web 项目，并把新文件或已有文件加到项目中。

第二章 Web 基础

这一章教你怎样创建页面，修改页面，并在你选择的浏览器中预览页面。

第三章 数据库基础

数据库基础描述如何对数据库建立连接、查询数据库，并在你的 Web 页面上显示数据库中的数据；还要建立允许你的用户编辑和更新数据库记录所需要的格式。

第四章 编辑基础

这一章涉及到基本 HTML 编辑，向 Web 页面增添部件，向 Web 页面增添脚本使其更具动态性质；还有选择其它编辑器的内容。

第五章 演练

演练让你通过介绍与插图对常见 Web 应用进行研讨。这一章包括的研讨内容涉及到“创建主页”、“调试脚本”、“与多个开发者共同工作”、“部署 Web 应用”和“简化数据录入页面”。

第一章 Web 项目管理

在 Microsoft Visual InterDev 中，你首先创建一个 Web 项目，这样你就可以：

- 在本地管理工作文件。
- 维护服务器上的主文件。

一个 Web 项目包含一些文件和信息，它们都是在 Microsoft Visual Studio 创建并发布单个 Web 应用所需要的。一个 Web 应用中的文件可以包括几种不同类型的文件格式，可供在设计的时候进行修改：HTML 页面、激活服务器页面（ASP）、图像文件、布局和主样板等。

Web 项目管理 Web 应用的两种副本：本地副本和主副本。所有主 Web 应用程序文件都存储到主 Web 服务器中。在编辑文件之前，你要从该服务器中接收文件，这样，文件的工作副本就放到你本地 Web 应用中。

在多个开发者情况中，开发组的每个成员都要有他们自己的项目，这些项目可以参照同一个主 Web 应用。对于演练的情况，见第五章“演练”中的“同多个开发者一同工作”一节。关于项目、服务器和 Web 应用的更多内容，见第六章“Web 项目概念”中的“项目结构”一节。

注意：你可以使用 Visual InterDev 建立两种类型项目——Web 项目和数据库项目。关于数据库项目的信息，见第二十二章“管理数据库项目”。

1.1 建立 Web 项目

Microsoft Visual InterDev Web 项目存储在你本地计算机中。你计算机中的文件构成了本地 Web 应用。在 Microsoft Visual Studio 中，Web 项目和其他项目类型都由一种解决方案进行访问。

创建新 Web 项目

- 如果你已经有了解决方案并已经在 Project Explorer 中打开，则把一个新项目加到一个解决方案中。

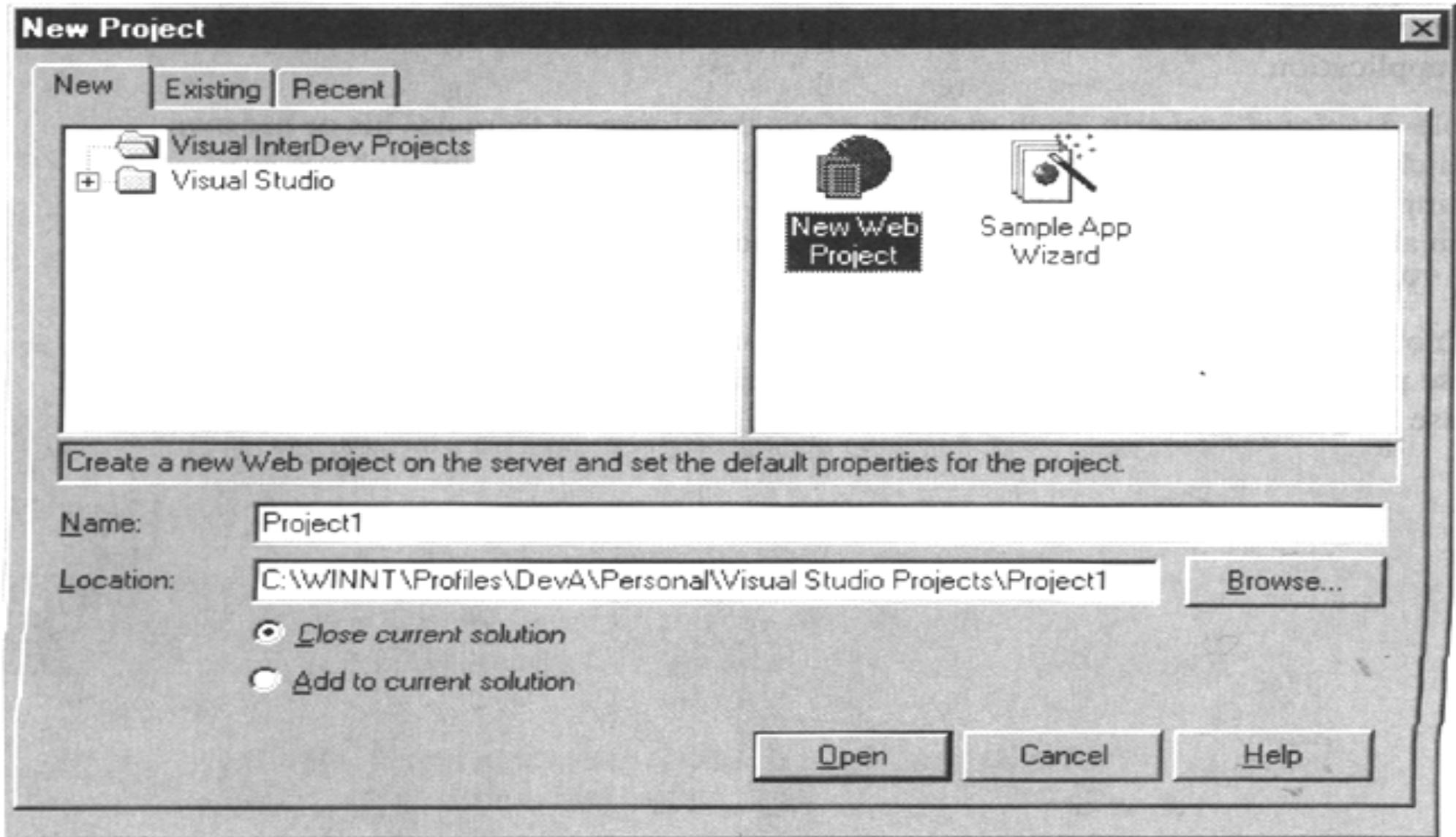
或者

同时建立一种新解决方案和一个新项目。

如果你正在建立你第一个 Web 应用，那么同时建立一种解决方案和一个项目就很容易。

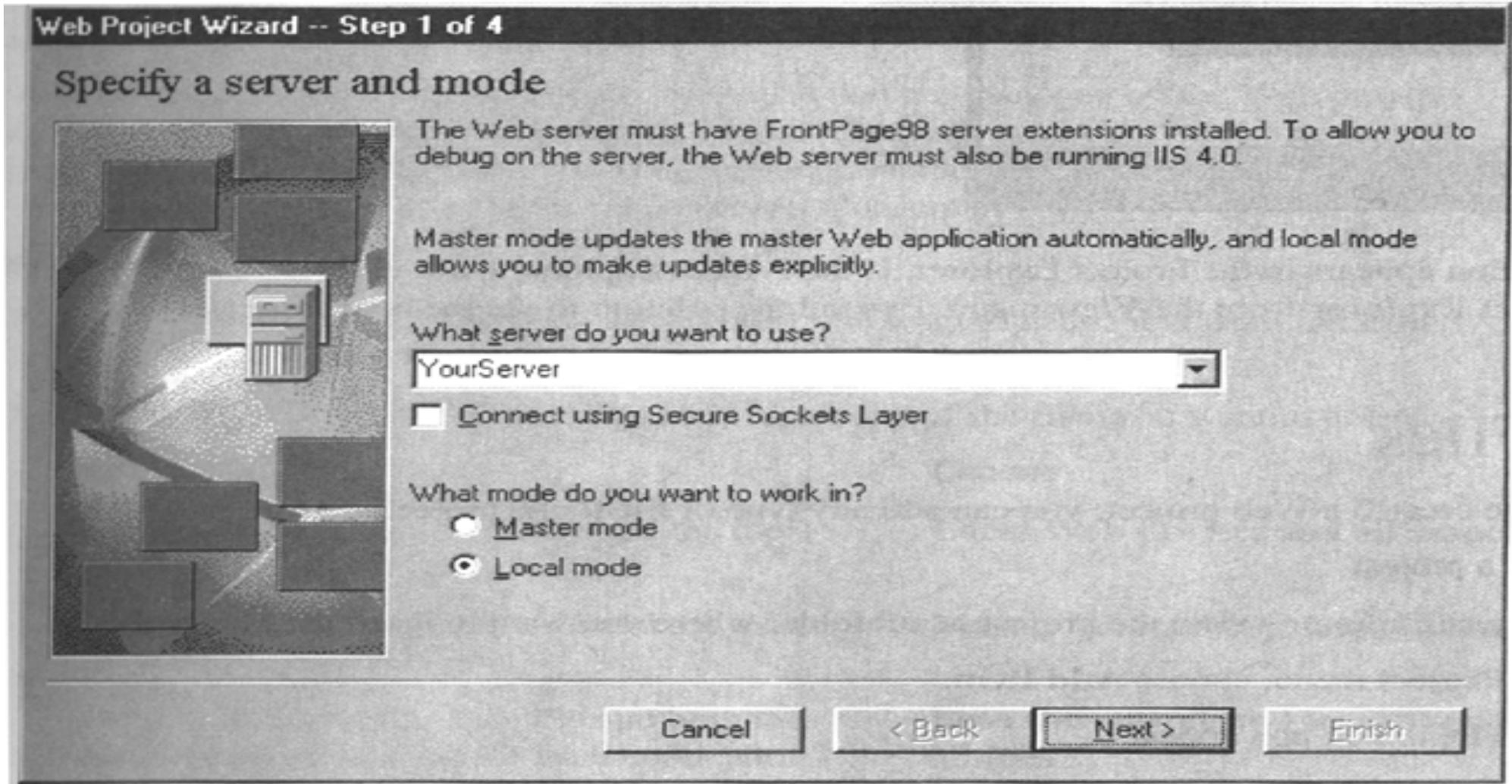
同时建立一种解决方案和一个 Web 项目

1. 从 File (文件) 菜单选择 New Project (新项目)。
2. 从 New 选项卡，选择左窗格的 Visual InterDev Project 和右窗格的 Project。



3. 在 Name 文本框中，键入新解决方案的名字。
如果你已经打开一种解决方案，就选择 Close current solution 选项。

4. 单击 Open, 启动 Web Project Wizard (新项目向导)。



步骤 1: 指定服务器和方式

不管你是否希望用 Secure Sockets Layer 进行连接, 必须指定你的 Web 服务器名字; 还要指定是主方式 (Master mode) 还是本地方式 (Local mode)。

步骤 2：指定你的 Web

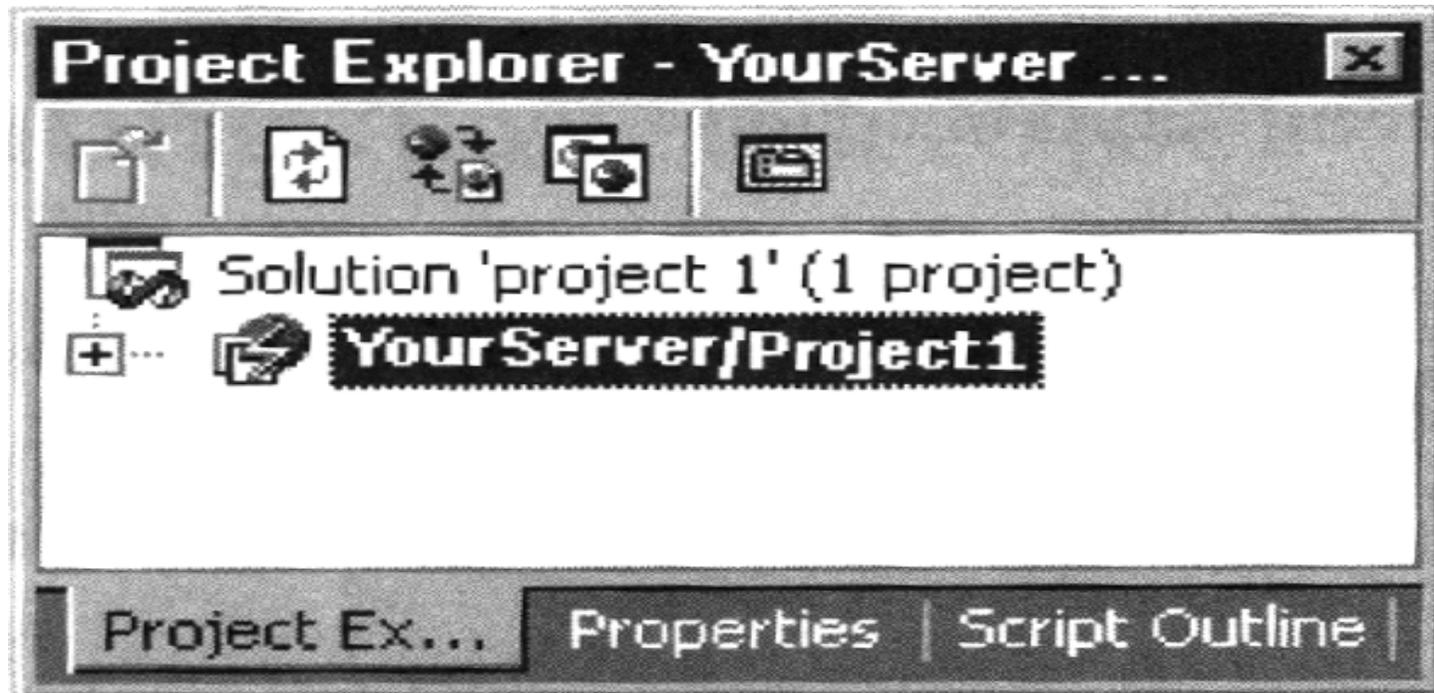
你可以选择用 Visual InterDev 在你的 Web 服务器上建立新的 Web 应用，或选择连接到已经有的 Web 应用程序上。

步骤 3：申请一种布局图

你不一定要使用一种布局图。如果你愿意，你现在可以选择 None，以后再申请一种布局图。

步骤 4：申请一个主样板

你不一定要使用一个主样板。如果你愿意，你现在可以选择 None，以后再申请一个主样板。



新解决方案出现在 Project Explorer 中。如果 Project Explorer 是不可

见的，从 View 菜单中选择 Project Explorer。扩充这个解决方案，查看新项目及其文件。

1.2 添加文件

一旦建立起 Web 项目，你就可以在项目中添加任何类型的文件。

向项目中插入文件

1. 在 Project Explorer 中，选择你想插入文件的项目或子文件夹。

2. 选择 Project 菜单中的 Add Item。

出现 Add File 对话框。

3. 在 New 选项卡中，选择右窗格的一种文件类型，为 Name 框提供一个名字。
详见第二章“Web 基础”中“建立页面”一节。

或

在 Existing 选项卡中，浏览该文件或浏览插入的一些文件。要确保在 Files of Type 框中选择相应的文件类型。

4. 单击 Open, 向 Web 项目添加文件。

注意:你还可以把文件或文件夹添加到项目中，办法是从 Windows Explorer 把文件或文件夹拖到 Project Explorer 的一个文件夹中。

1.3 删除一个 Web 项目

你可以删除一个 Web 项目及其相关的 Web 应用，你也可以只从解决方案中去掉项目而不删除它的文件。

一个 Web 应用有两类 Web 文件副本：本地副本和主副本，它们都是由 Web 项目管理的。除了删除项目文件 (.vip) 之外，你还可以删除这两种应用副本中的一种，或两种都删除。关于本地文件和主文件的详细内容，参见第六章“Web 项目概念”中的“Web 文件处理”一节。

删除一个 Web 项目

1. 在 Project Explorer 中，选择你要删除的 Web 项目，再选择 Edit 菜单中的 Delete Project.
2. 在 Delete Project 对话框中，选择你要删除的文件副本。

删 除

选 择

你的本地 Web 项目 (.vip) 文件和 Web 应用的本地副本

本地 Web 项目和所有相关文件

你的本地 Web 项目 (.vip) 文件和 Web 应用的主副本

主 Web 项目和所有相关文件

你的本地 Web 项目 (.vip) 文件和 Web 应用的本地副本和主副本

主和本地 Web 项目及所有相关文件

3. 选择 OK

注意：如果你选择只删除本地 Web 项目和文件，过一会儿你可以建立一个

项目，这个项目指向仍然保存在服务器中 Web 应用的主副本。

如果你选择只删除主副本，你还可以删除本地项目文件 (.vip)，但是该本地目录和 Web 应用文件仍然保留在你的计算机中。

你可能希望保留你的项目及其相关文件，但是要从解决方案中去掉该项目。

从解决方案中去掉项目

1. 在 Project Explorer 中，选择你要去掉的 Web 项目。
2. 选择 File 菜单中的 Remove Project。
3. 在消息中选择 Yes。

该项目在解决方案文件中的引用子 (reference) 也被删除，但是项目文件和本地与主 Web 文件仍然保留。

第二章 Web 基础

一旦开始对 Web 项目进行工作，你会希望对你的 HTML 页面和 Active Server（激活服务器）页面进行试验，然后用新文件更新 Web 服务器。Visual InterDev 使得创建页面、修改页面、并在你选用的浏览器中进行页面预览都变得很轻松。完成这些工作之后再更新服务器。

2.1 创建页面

一旦有了 Web 项目，你就可以加上 HTML 页面和激活服务器页面（.asp 文件），使得你的 Web 站点工作起来。

注意：客户方脚本加到 HTML 页面上，服务器方脚本加到激活服务器页面上。

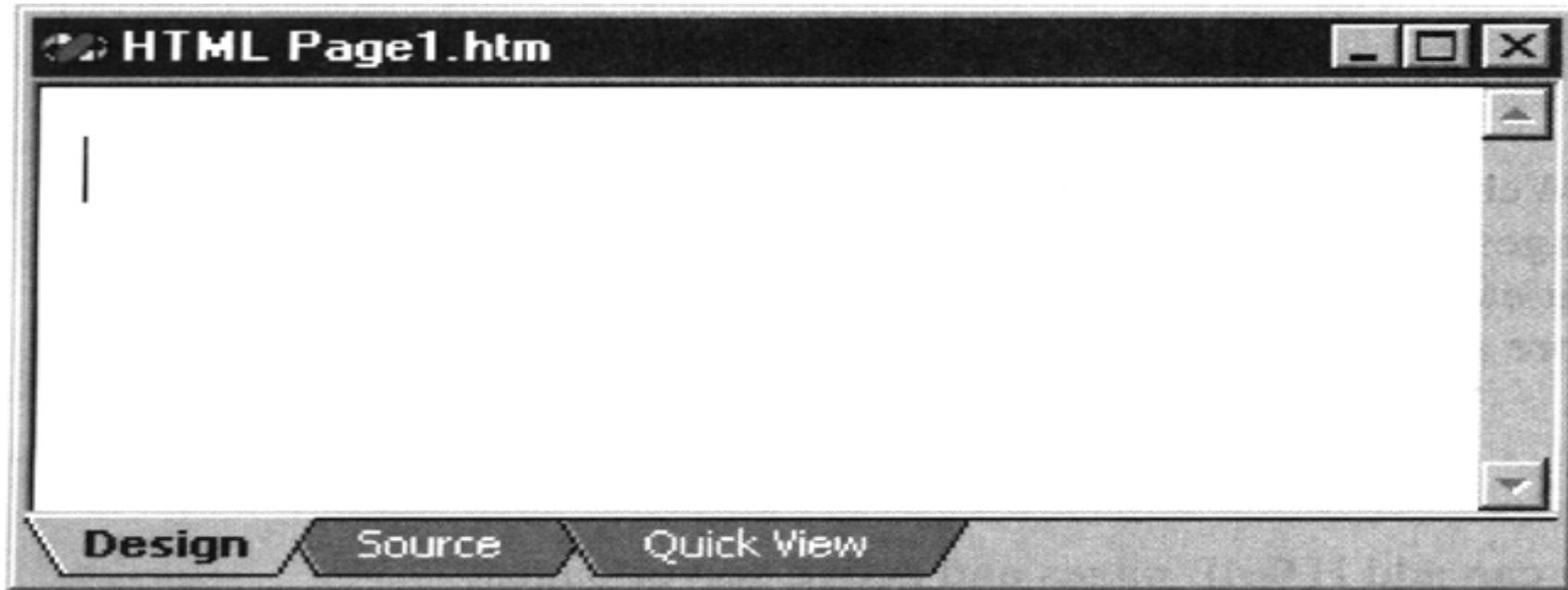
创建新页面

1. 在 Project Explorer 中，选择你要添加新页面的项目或子文件夹。
2. 选择 Project 菜单中的 Add Item。
3. 在 Add Item 对话框中，单击 New 选项卡。
4. 选择右窗格的 HTML 页面或 ASP 页面。
5. 在 Name 框中键入新文件名。

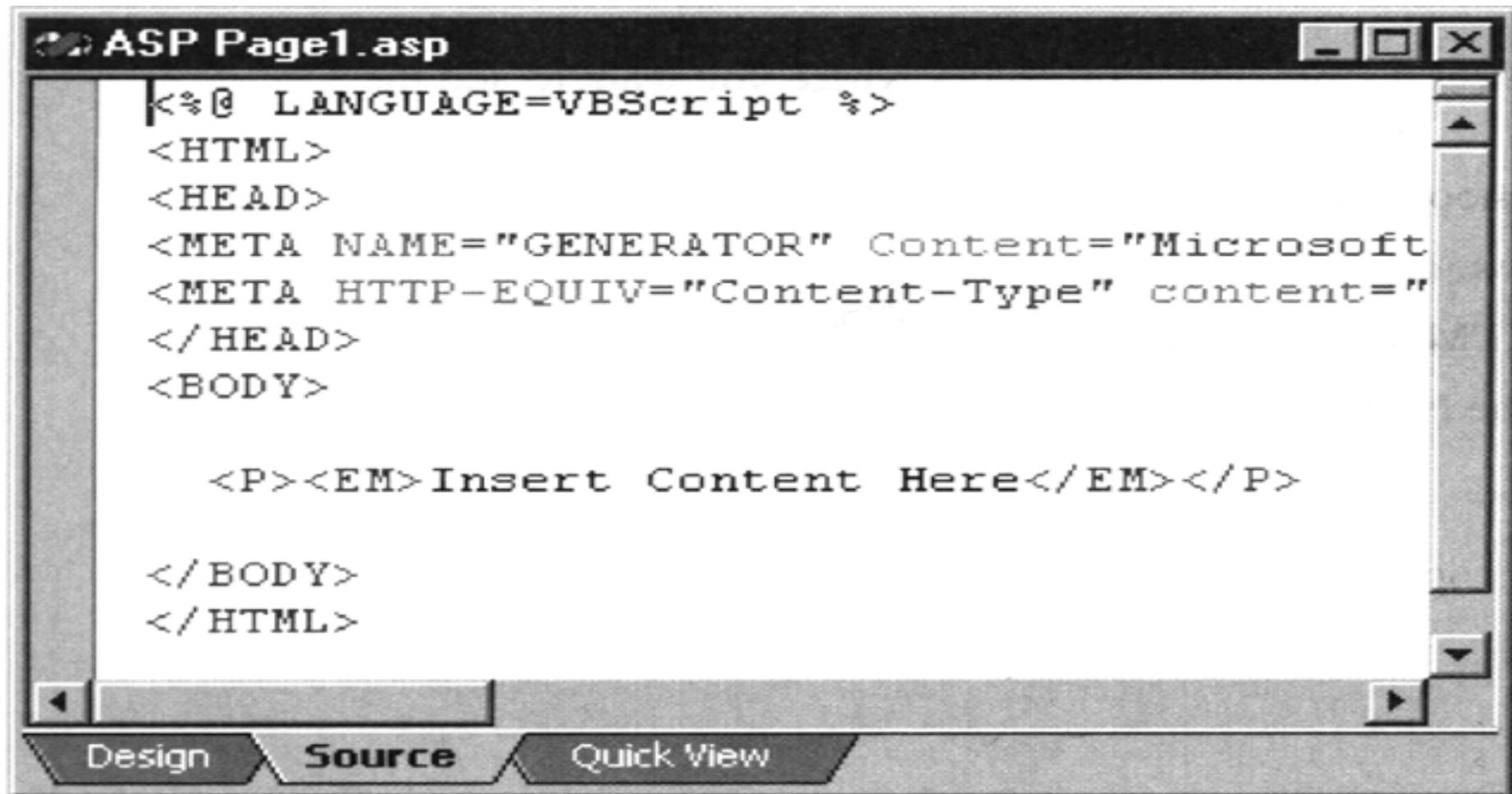
6. 单击 Open.

Visual InterDev 在编辑器中打开新页面。

- 编辑器对 HTML 页面的默认视图是 Design View。当你在 Design View 中编辑页面时候，文本也同时显示出来，并且所施加的格式，其样式同你在词处理软件中见到的文档一样。详见第四章“编辑基础”中“编辑 HTML”一节。



- 对于 ASP 页面的默认视图是 Source View。在你在 Source View 中进行编辑的时候，你的标记和脚本也都进行颜色编码，从而使得它们便于阅读。详见第四章“编辑基础”中“添加页面”一节。



```
<%@ LANGUAGE=VBScript %>
<HTML>
<HEAD>
<META NAME="GENERATOR" Content="Microsoft
<META HTTP-EQUIV="Content-Type" content="
</HEAD>
<BODY>

    <P><EM>Insert Content Here</EM></P>

</BODY>
</HTML>
```

2.2 保存页面

定时保存你的页面，避免损失你的工作，这是个好习惯。

保存当前页面

1. 单击页面，确保该页面是焦点页面。
2. 选择 File 菜单中的 Save.

该文件便保存到本地的计算机中。根据你的网络连接情况，该文件还可能在主 Web 项目中得到更新：

网络	本地状态	主状态
在线	页面只在本地保存	页面在本地保存并在主 Web 服务器中更新
离线	页面只在本地保存	页面只在本地保存

要了解本地状态和主状态之间区别的详细内容，参见第六章“Web 项目概念”中“项目结构”一节。

在保存页面之后，你可以在 Quick View 中或在浏览器中预览这些页面。详见下一节“预览页面”。

2.3 预览页面

在编辑 HTML 或 ASP 页面的时候，你可以借助于预览页面的办法方便地检查你的进程。

预览你正在编辑的页面

- 在编辑器中，单击窗口底部的 Quick View 选项卡。
注意：在 Quick View 中一个文件被查看之前不需要保存。

当预览一个页面时，你可以看见页面如何出现在 Internet Explorer 4.0 中。但因你在本地预览页面，因而有一定局限性：

- 服务器方的脚本不予以处理。
- 数据绑定设计期间控件不显示数据。

要了解这些限制，可在浏览器中直接阅览页面，例如在 Internet Explorer 4.0 中阅览。

在浏览器中预览页面

1. 在 Project Explorer 中选择文件。
2. 选择 View 菜单中的 View in Browser。

如果你以本地方式工作，文件的本地副本被默认浏览器打开。

在不同浏览器中预览页面

1. 在 Project Explorer 中选择文件。
2. 选择 View 菜单中的 Browse With。
3. 在 Browse With 对话框中选择一浏览器。

如果你在计算机中安装了一个浏览器，而这个浏览器没在对话框中列出来，你可以单击 Add，把这个浏览器加到列表中。

注意：你可以在 Browse With 对话框中改变默认浏览器，办法是从列表中选择一种浏览器，然后选择 Set as Default。

第三章 数据库基础

Visual InterDev 赋予你连接大多数数据库中数据的能力。例如，你可以连接到 SQL 服务器、Access 或 Oracle 数据库。一旦建立起对数据库的连接，你就可以选择数据库中一组特定的记录，并在你的 Web 页面上显示这组记录。

为了便于在你的 Web 页面上显示并编辑数据，Visual InterDev 支持范围广泛的数据绑定设计期间控件。

- Recordset (记录集) 控件与数据库有关联，允许你提取一组记录。
- RecordNavbar 控件允许你在 Web 页面上从数据库中的一个记录移动到另一个记录中。
- 除了单个控件，例如标签、文本框、列表框之外，还可以显示数据库中的数据。
- Grid (栅格) 控件使得在 Web 页面上显示你数据库中的数据更加容易。在一个栅格中你可以显示多个记录、对记录进行分类、并控件数据的格式和外形。

当你把一个设计期间控件加到 ASP 或 HTML 页面上时，这种控件能自动加到页面上的脚本中，这样的页面上可以显示数据，还具有诸如编辑数据或在各记录之间进行导航的功能。你还可以扩充 Web 页面的功能（例如增加有效性或客户导航），办法是编写一个事件句柄，用来修改控件的状态。

注意：你还可以使用数据环境添加数据连接，并且建立和管理一个位置的数

据绑定控件。这使你可直接创建强有力的客户解决方案，使用 Visual InterDev 编辑器和调试器来修改用于在 Web 页面上显示数据和操纵数据的脚本。详见第十八章“数据库概念”中的“数据环境”一节。

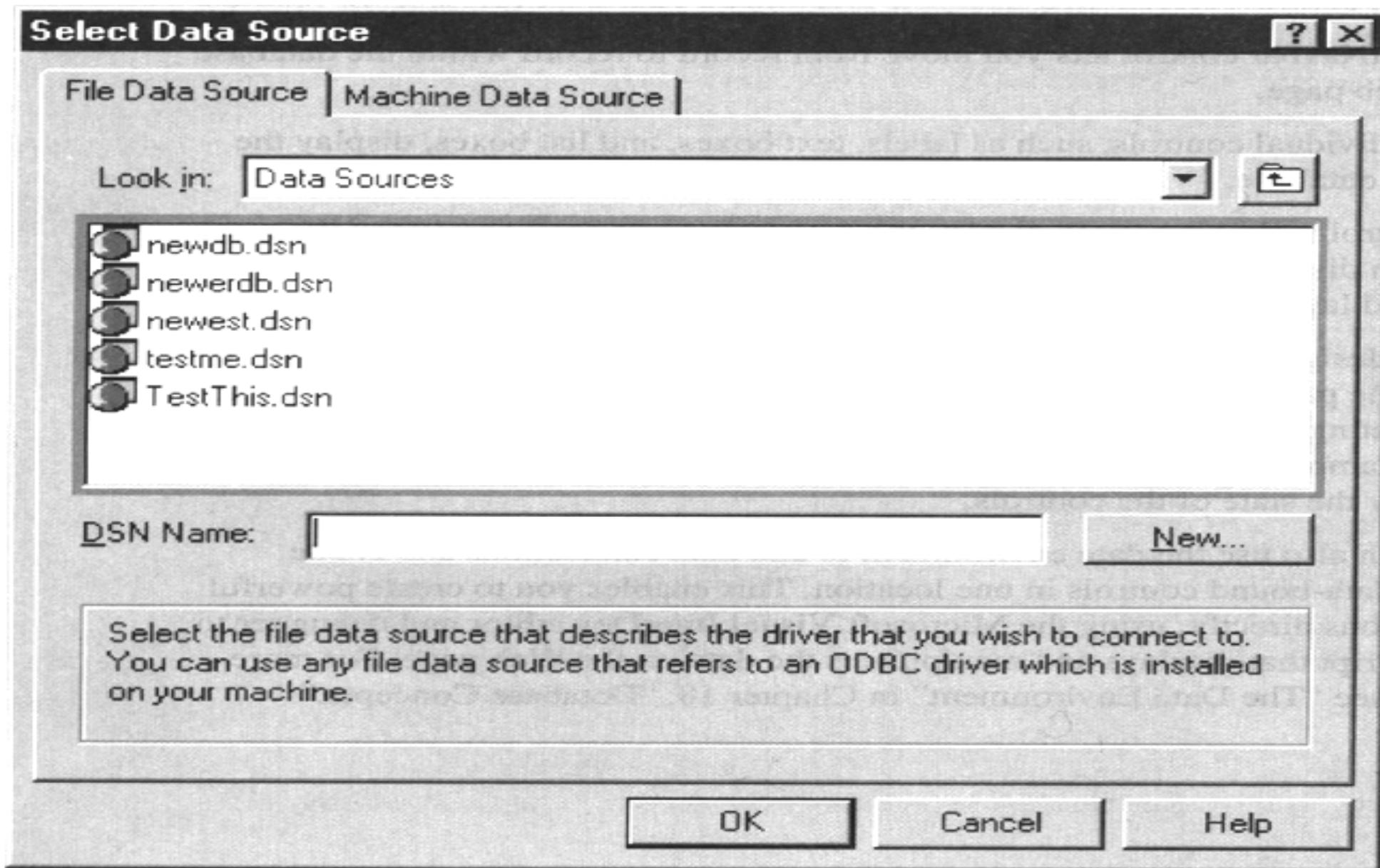
3.1 连接数据库

在 Visual InterDev 中你可以显示或编辑 Web 页面上数据之前，你必须连接到一个数据库上。一条数据连接为你的 Visual InterDev 项目对一个特定数据库提供访问权。然后你就可以使用这条数据连接连通数据库，并在你的 Web 页面上显示其数据。

当连接一个数据库时，你首先要为数据库建立一个“数据源名字”(DSN)或者选择一个已有的名字，然后使用这个 DSN 建立数据连接，并把它加到你的项目上。

建立数据源名字

1. 打开 Visual InterDev 项目，并在 Project Explorer 中选择它。
2. 选择 Project 菜单中的 Add Data Connection。
出现 Select Data Source 对话框。



3. 在 File Data Source 框中选择 New.

4. 选择你希望的数据库驱动软件，并选取 Next。这个数据库驱动软件必须同你要连接的数据库类型相匹配。例如，如果你的数据库是一种 SQL 服务器数据库，就要选择 SQL 服务器驱动软件。
5. 为连接文件键入名字，并选择 Next。
扩展名字 .dsn 自动加到文件名字上。
6. 选择 Finish。
7. 在出现的一个或多个对话框中，填入与驱动软件有关的信息，例如要访问的数据库名字或要打开的文件名字。

注意：当你指定数据库的时候，使用到数据库的相关路径，不是数据库在你开发计算机上的位置。例如，如果数据库位于 Web 服务器上，就使用到该数据库的 UNC 路径。这要保证数据库在你的 Web 服务器上的可用性。

你返回到 Select Datas Source 对话框，你建立的文件数据源名字就显示在列表中。

关于你可建立的数据源类型的详细内容，参见微软 Visual 数据库工具在线文档中“选择数据源名字”部分。

提示：你还可以使用“ODBC 数据源管理器”（在你的控件面板上的 ODBC 文件夹中）建立并命名数据源。这些数据源将出现在 Select Data Source 对话框中，你可以使用它们在 Visual InterDev 中建立数据连接。

一旦为一个数据库建立了数据源名字，就可以使用它为该数据库建立数据连接。

为项目增添数据连接

1. 在 Select Data Source 对话框中的 File Data Source 选项卡上选择你建立的数据源名字，并选定 OK。

显示出的一个对话框中给出数据连接的参数。你所能见到的对话框取决于你所选定的数据源的类型。例如，如果你选择了一个 SQL 服务器数据源，你就可以看见一个 SQL Server Login 对话框。

2. 键入数据连接参数。例如，对于 SQL 服务器数据库，要键入登录标识和口令。

所显示的数据连接在你项目中的 DataEnvironment 文件夹中，在 global.asa 文件夹之下。你还可以在 Data View 窗口中浏览和编辑这个数据库中的数据。

现在你已经做好了准备从数据库中接收并阅览你希望的数据。详细内容，参见下一节“查询数据库”。

3.2 查询数据库

一旦把数据连接加到数据库上，就可以查询数据库，查找你希望由特定 Web 页面使用的一组记录。Visual InterDev 使得一组数据库记录对 Recordset（记录集）设计期间控件具有可用性。

对 ASP 或 HTML 页面增加记录集控件

1. 确保你所设置的选项可以用图形方式进行浏览。选择 View 菜单中的 View Controls Graphically。为把这个选项作为默认方式，使用选项对话框中的 HTML 节点。
2. 打开编辑器中的 ASP 或 HTML 页面。关于建立 ASP 或 HTML 页面的信息，参见第二章“Web 基础”中的“创建页面”一节。
3. 把 Recordset（记录集）控件从 Toolbox 的 Design-Time Controls 选项卡中拖到该页面上。

提示：如果记录集控件没在 Toolbox 中出现，用右键单击 Toolbox，选择 Customize Toolbox 并加上记录集控件。

你现在可以为记录集控件规定一组记录。

规定一组记录

1. 在 ASP 或 HTML 页面上的记录集控件中把 Connection 特性设置为数据连接的名字，这个数据库是你希望查看记录的数据库。
2. 把 Database Object 特性设置成 Table，这个表用来显示数据库表中的所有记录（你还可以把这个特性设置成数据库中的其他记录集，例如一个视图或存贮的一个过程）。
3. 把 Object Name 特性设置成表的名字或其他数据库对象的名字，它们的记录就是你要使用的。

你可以显示这组记录中的数据，办法是把数据绑定控件加到你的 ASP 页面或 HTML 页面上，再把记录集控件设置为控件的数据源。然后这个控件被绑定到

那个表中的记录上。详细内容，参见下一节“显示记录”。

注意：你还可以使用数据环境建立数据连接，并把记录集控件加到 Visual InterDev 项目上。如果希望能够用编程办法在一个位置管理你的数据和记录，数据环境特别有用。详细内容，参见第十九章“查看数据”和第十八章“数据库概念”中的“数据环境”一节。

Visual InterDev 利用 Internet Explorer 4.0 客户方数据绑定的优点以及较传统的服务器方数据绑定的优点。关于客户方数据绑定的讨论，参见第十八章“数据库概念”中的“数据绑定”一节。

3.3 显示记录

你可以使用数据绑定设计期间控件，例如标号、文本框和列表框，在 Web 页面上显示数据；还使用导航控件在各个记录之间进行导航。

在 ASP 或 HTML 页面上增加数据绑定控件

1. 要确保你设置的选项能以图形方式查看控件。选择 View 菜单中的 View Controls Graphically。为把这个选项设置为默认方式，要使用 Options 对话框中的 HTML 节点。
2. 打开一个 ASP 或 HTML 页面，这个页面要在编辑器中包含一个记录集控件。
3. 把数据绑定控件从 Toolbox 的 Design-Time Controls 框中拖到该页面上。例如，你可以把 Textbox 控件拖到页面上，显示一个特定段的内容。
4. 右键单击该控件，并选择 Properties。

5. 把该控件的 Recordset (记录集) 特性设置为当前页面上一个记录集控件的名字。

6. 如果该控件有一个 Field 特性, 就把它设置成记录集一个段的名称, 这个段是你希望用该控件进行显示的段。

7. 按照希望, 设置其他控件特性。

要得到数据绑定控件以及设置它们特性的详细内容, 参见 Visual InterDev 在线文档中“设计期间控件”和“对话框与窗口”两部分。

8. 关闭特性窗口, 保存 .asp 文件或 .htm 文件, 在浏览器中预览该文件。

文本框将显示的段是你在所连接的数据中选择的第一个记录中所选择的段。

有关在浏览器中预览 ASP 或 HTML 页面的详细内容, 参见第十四章“管理站点图”中“在浏览器中预览页面”一节。

使用 RecordsetNavbar 控件, 可以很容易地为在你的 Web 页面上显示的各个记录之间提供导航。

在记录间提供导航

1. 在编辑器中打开 ASP 或 HTML 页面。

2. 把 RecordsetNavbar 控件从 Toolbox 的 Design-Time Controls 选项卡中拖到该页面上。

3. 用右键单击该控件并选择 Properties。

4. 把该控件的 Recordset (记录集) 特性设置成 ASP 或 HTML 页面上记录集控件的名字。

5. 如果需要，关闭特性窗口，保存 .asp 文件或 .htm 文件，并在浏览器中预览该文件。你可以使用 RecordsetNavbar 控件在基本记录集中的各记录之间移动。

作为默认控件，RecordsetNavbar 控件提供 Move to First（移到第一个记录）、Next（下一个记录）、Previous（前一个记录）和 Move to Last（移到最后一个记录）按钮。你可以使用这些按钮显示页面上基础记录集中的不同记录。

你还可以编写事件句柄，使用由这个控件所提供的对象模型，按客户要求改进导航的表现。

你还可以使用 Grid（栅格）设计期间控件，在页面上显示数据库中的多个记录。详细内容，参见 Visual InterDev 在线文档中“数据绑定栅格实例”（Data-bound Grid Sample）和“栅格特性对话框”（Grid Properties Dialog Box）。

要了解在 Web 页面上显示数据的详细内容，参见第十九章“查看数据”、第二十章“修改数据”和下一节“建立事件驱动格式”。

3.4 建立事件驱动窗体

你可以使用 FormManager 控件很容易地在你的页面上建立具有不同模式的一种窗体。例如，你可以建立一个数据键入窗体，具有插入、更新和删除模式。在规定了控件事件出现的时候，例如单击一个按钮，则窗体从一种模式变成另一种模式。

你还可以规定特性设定和方法，在特定窗体模式确立的时候和从一种模式

向另一种模式转变期间执行这些设定和方法。使用 FormManager 控件的特性页面，完成上述所有工作都不需要编写脚本。

关于如何使用 FormManager 控件建立数据录入窗体的详细例子，参见第五章“演练”中的“简化数据录入页面”一节。

使用 FormManager 控件建立 Web 页面的第一步是加上一个记录集控件和任何其它设计期间控件，这些控件都是你希望在页面上以不同窗体模式显示时使用的。例如，你可能希望用文本框显示数据，用按钮在不同窗体模式之间进行转换。对于如何在 ASP 或 HTML 页面上增添数据绑定控件，参见本章前边的“查询数据库”和“显示记录”，以及第十九章“查看数据”。

当在页面上有了按你希望的窗体模式所使用的控件的时候，你就可以添加 FormManager 控件，然后规定它们的模式和转变。

注释：尽管很常见，你还是不一定要使用 FormManager 控件按所建立的窗体显示数据。如果你不在窗体上显示数据，你就不需要使用记录集控件。

向页面添加 FormManager 控件

1. 确保你设置的选项能以图形方式观察控件。在 View 菜单中选择 View Controls Graphically。要把这个选项设置为默认方式，使用 Option 对话框中的 HTML 节点。
2. 在编辑器中打开 ASP 或 HTML 页面。
3. 把 FormManager 控件从 Toolbox 的 Design-Time Controls 选项卡中拖到该页面上。

注意：如果你在页面上正在使用 DHTML，要确保你放置 FormManager 控件

的页面位于你希望使用该格式的任何其它控件的下边。这样就可以保证 FormManager 将访问该页面的所有对象。

增添窗体模式

1. 用右键单击页面上的 FormManager 控件，在快捷菜单上选择 Properties 命令。
2. 如果需要，在 Properties Pages 的 Form Mode 选项卡中键入 FormManager 控件所用的名字。这个名字用于在脚本产生的时候标识该控件。如果你不提供名字，就使用默认名字。
3. 在 States 组中，在 New Mode（新模式）框内键入你希望使用的模式名字。
4. 使用箭头键进到 New Mode 框的右侧。
5. 在 Form Mode 列表中显示出新模式名字。

关于如何观察模式的内容，参见本章“定义模式”一节。

规定默认模式

1. 用右键单击页面上 FormManager 控件，选择快捷菜单中的 Properties 命令。
2. 在 Property Pages 的 Form Mode 选项卡上选择一种模式，这是在 Default Mode 列表中列出的你希望使用的默认模式。

定义模式

1. 用右键单击页面上 FormManager 控件，选择快捷菜单中的 Properties 命令。

2. 在 Property Pages 的 Form Mome 选项卡上，选择你希望在 Form Mode 列表中定义的模式。
3. 在 Property Pages 的 Form Mode 选项卡中选择一种控件，其特性是你希望设置的，或者按照这种特性你希望运行 Object 段中的一种方法。例如，你可以把 Textbox 控件的关闭特性设置为真，或使用 addItem 方法把一个值加到 Listbox 控件上。
4. 在 Member 字段，选择你希望设置的特性或希望运行的方法。
5. 在 Value 字段段中键入你希望设置特性的值，或方法的任何参数。
如果设置一个特性，Value 字段就给出 <value>。用特性设置值代替它。
如果设置值是一个字符串，要使用引号。
如果你运行一种方法，这个 Value 字段便显示一对空的圆括号。在这个圆括号中放入该方法的任何参数。在下拉列表的 Member 字段列出各种方法及其参数。
6. 重复步骤 3 到 5，直到你定义完这种窗体模式的所有特性设置和方法。
关于定义窗体模式的更多内容，参见第五章“演练”中的“简化数据录入页面”一节。

规定模式转变出现的时间

1. 用右键单击页面上 FormManager 控件，选择快捷菜单中的 Properties 命令。
2. 在 Property Pages 的 Action 选项卡上选择一种模式，这种模式的转变是你希望在 Form Mode Transitions 表中的 Current Mode 字段进行定义

的。

3. 在 Object 字段选择一个对象，这个对象是你希望用来触发从一种模式到另一种模式进行变更的对象。例如，你可能希望用户单击 Insert（插入）按钮从 Update（更新）模式转变成 Insert 模式。
4. 在 Event 字段选择一个控件事件，这个事件将触发模式转变。例如在 Insert 按钮上的单击事件。
5. 在 Next Mode 字段选择你希望在控件事件出现时进入的模式。例如你可能希望进入 Insert 模式。
6. 重复步骤 2 到 5，直到对窗体定义完所有模式之间的转变。
7. 在 Actions Performed Before Transition 表中，你还可以设置特性，定义方法，它们在出现模式转变的时候产生作用，但是要在进入新模式之前。关于如何进行这些设置的内容，参见本章中前边的“定义模式”一节。

除了 Visual InterDev 和设计期间控件的 Scripting Object Model（脚本对象模型）提供的方法之外，你还可以引用在页面脚本以一种模式定义的方法，或一种模式转变方法。例如，你可能希望运行一种方法，这种方法在按一下 Save 按钮之后，但是要在进入新模式之前使数据才有效。

使用户定义的方法与一种模式产生关联

1. 按照上边“定义模式”或“规定模式转变出现事件”的内容去做。
2. 在 Actions Performed for Mode 表中 Form Mode 选项卡的 Member 字段内，或在 Actions Performed Before Transition 表中 Action 选项卡的 Member 字段内，键入你的用户定义的方法。

3. 在 Value 字段中的圆括号内键入方法参数（如果有的话）。

你的方法在模式激活的时候，或在出现模式转变的时候执行。如果你拥有多个用户定义的方法，这些方法将按表中特性页面上出现的顺序运行。

用户定义方法的数据格式录入举例，参见 Visual InterDev 在线文档中数据录入格式实例。

第四章 编辑基础

Visual InterDev 应用的开发都集中到创建 Web 页面上。为了帮助你，Visual InterDev 提供一组丰富的工具，用于对 HTML 页面进行编辑和编写脚本（.htm 文件），以及对激活服务器页面（Active Server Pages）进行编辑和编写脚本（.asp 文件）。

你的主要工具是 Web 编辑器。你可以使用几种编辑器加工 Web 页面，到底使用哪一种，要取决于你的爱好，以及你要用 Web 页面做什么。默认编辑器是 Visual InterDev HTML 编辑器。这种编辑器允许你以 Design 视图进行工作，可以看见你的页面，就像在浏览器中看见的一样。此外，你也可以在 Source 视图进行工作，它允许你以“原始”HTML 文本和你页面上的对象进行工作。

4.1 编辑 HTML

要在 Web 页面上显示文本，你可以简单启动一个新 HTML 文档（.htm 文件）或 ASP 页面（.asp 文件），并把文本键入到页面上。如果需要对文本赋予一定格式，或加上一些特性，如图像或到页面的链接，就要使用 HTML（超级文本标记语言）。

如果你不是一位 HTML 专家，你大概希望在开始工作以前先熟悉一下 HTML

是如何工作的。熟悉 HTML 可以帮助你获得基于 Web 的应用以外的许多东西。

在建立 HTML 文本的时候，你经常使用编辑器中的 Design 视图，这个视图按指定的格式显示文本，即在文字处理软件中你阅读文档所使用的样式。设计视图是 .htm 文件的默认编辑器。

要对文档上更精确地控制，可以在 Source 视图中进行编辑，这个视图显示文本和和赋予文本格式的 HTML 标记。

使用 HTML 标记

HTML 编写的时候要使用一些标记 (tags)，这些标记是嵌入在文本中的格式命令。标记两边是尖括号 (<和>)，从而把标记与周围的文本区别开来。标记一般是成对使用，在你希望赋予格式的文本上使用的顺序是：一个开头标记、文本和一个结束标记(结束标记带有正斜杠符号 (/))。例如下边一行示出你对一些文字赋予粗体格式 ()，对另一些文字赋予斜体格式 (<I>)：

```
<B>Welcome</B> to my <I>home page</I>.
```

当你在浏览器中显示的时候，这行文字是如下这个样子：

```
Welcome to my home page.
```

HTML 包括用于字符格式化和段落格式化的标记（例如使段落置于中心位置），还有用于规定特性的标记，如规定图像、链接、表格和格式等。下表简单介绍常用标记。

插入	使用	举例
新段落	<P>	<P>这是第一段</P><P>这是第二段</P>
图像	加上要显示图像文件的名字	
链接	<A>加上要跳到的页面地址，后跟链接的文本	Go to Home page
表格	<Table>用于定义表格 <TR>用于每行 <TD>用于每个表元	<TABLE> <TR> <TD>Row 1, Cell 1 </TD> </TR> <TR> <TD>Row 2, Cell 1 </TD> </TR> </TABLE>
表单	<FORM>用于定义表单 <INPUT>用于定义控件，如文本框和按钮	<FORM> Name: <INPUT Type="Text"> <INPUT Type="Submit"> </FORM>

注意：使用 HTML 菜单中的命令，你很容易在文档中创建 HTML 元素。

详细内容参见 Visual InterDev 在线文档中“Inserting Links and Bookmarks in Web Pages”（在 Web 页面中插入链接和书签）和“Creating and Editing Tables in the HTML Editor”（在 HTML 编辑器中创建和编辑表格）。

要创建格式，你可以使用 Visual InterDev 设计期间控件。详细内容，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“用设计期间控件创建格式”一节。

在编辑器中创建 HTML

Visual InterDevb 编辑器允许你在新页面或已有页面中创建 HTML。详细内容参见第二章“Web 基础”中的“创建页面”一节。

在进行编辑的时候，可以在下列不同形式中使用 HTML 进行工作：

- 在 Design 视图中，你可以为文本和节指定格式，其格式形式同文字处理软件的格式；而该编辑器为你嵌入相应的 HTML 标记。
例如，如果你选择文本并单击工具栏中加粗按钮，编辑器便把 和 标记插入到所选择文本的两边。更复杂的标记，例如图像标记和表格标记，你可以用填充选择把整个元素插入到一个对话框中。
- 在 Source 视图中，你可以看见 HTML 标记，并直接编辑这些标记。你还可以使用工具栏按钮或菜单命令嵌入这些 HTML 标记，如在 Design 视图中所做的那样。例如，可以用选择一个菜单命令或指定一个对象页面的办法来插入一条链接。

在 Source 视图中编辑器为 HTML 文本配置颜色可以帮助你醒目地区分

HTML 标记的不同部分。例如，标记名字用一种颜色显示；标记属性，例如 SIZE= 用另外一种颜色显示。未被确认的文本也可以加上颜色，这样可以帮助你迅速发现错误。例如不匹配的括号。

提示：你可以在选项窗口中改变编辑器字体样式和颜色。从 Tools 菜单选择 Options, 扩展 Text Editor 节点，再选择字体和颜色。

- 在 Quick 视图中，你可以看到 .htm 文件看上去就像在 Internet Explorer 中见到的一样（Quick 视图显示的 .asp 文件可能不十分精确）。

在编辑过程中任何地方都可以在不同视图之间进行转换，从而你可以及时看到正进行的编辑效果如何。

转换视图

- 在编辑窗口的底部选择你希望的视图标识符号。

注意：当你在 Source 和 Design View 之间进行转换时，编辑器可能有一个空白调整时间（例如出现空白和标识符号），接着完成任何未完成的在 Web 页面上存在的 HTML 标识符号。详细情况，参见 Visual InterDev 在线文档中“Design View, HTML Editor”（设计视图，HTML 编辑器）。

当你建立或开发一个 Web 页面时，是用 Design 视图还是用 Source 视图显示出来，取决于你正在编辑的 Web 页面的类型所对应的默认视图。

改变默认视图

1. 从 Tools 菜单中选择 Options.

2. 在左窗格展开 HTML.

3. 在 Initial View 范围内为 HTML 页面 (.htm 或 .html) 或为 ASP (.asp) 页面选择默认视图。

4.2 为页面增添部件

除了键入 HTML 之外，你还可以为你的页面增加各种部件，其中包括：

- 控件与对象
- 引用其它文件（包括图像、链接和样式图表）
- 数据库链接
- HTML 图表

一般而言，你可以把部件从 Toolbox 或从 Project Explorer 拖出来加到你的页面上。

添加控件和对象

当你用 Visual InterDev 进行工作时，Toolbox 列出你可以在页面上使用的控件。这些控件包括：

- **Visual InterDev 设计期间控件：** 用户接口控件，如文本框和按钮，这些都是你为创建 Web 页面和为 Web 页面编写脚本而使用标准的面向对象的技术。
- **ActiveX 控件：** 在你的计算机中登记的控件。
- **HTML 控件：** 标准 HTML 控件，如文本区域和按钮。

- **服务器部件**：得到 Internet 信息服务器（IIS）支持的部件和对象列表，这些部件和对象可用于服务器脚本中，其中包括 ActiveX Data Objects(ADO)、Index Server 对象等。

要使用这些部件，把它们从 Toolbox 拖动到页面上希望出现的位置。大多数控件和对象都在 HTML 编辑器中，以 Design 视图和 Source 视图的图形方式显示出来。在 Source 视图中，你可以规定对象的文本版本

注意：你可以增添新元素和符号，使 Toolbox 符合用户要求。详细情况，参见 Visual InterDev 在线文档“Toolbox”题目下的内容。

增添文件引用关系

当向页面上增添图像、链接、文档、样式图表的时候，你不是实际把元素嵌入到页面上，而是增加对指定元素的引用关系。例如，当增添一个图像时，你是增添一个 HTML 标记，这个标记中有包含该图像的 .gif 文件名字。

你可以把一个引用元素从 Project Explorer 拖出来，增添到你的页面上。你可以对下列元素增添引用关系：

- **图像** 如果你把一个图像从 Project Explorer 拖到你的页面上（例如从一个图像节点），HTML 就建立一个 标记。
- **Web 页面** 把一个 .htm 或 .asp 文件拖到页面上，你就建立一条到那个文档的超级链接。
- **文档** 你可以把文本文件或文字处理文档从 Project Explorer 拖到页面上，建立一条允许用户下载该文档的链接。
- **样式图表** 如果你把一个样式图表（.css 文件）从 Project Explorer

拖到文档的抬头上，你就自动建成一条到该样式图表的链接。

你可以把元素从任何拖动源，包括 Windows Explorer 拖到页面上。当这样做的时候，元素的 URL 就同你的项目根建立了关系。

注意：对你的开发项目，你可能需要在 Source 视图中为链接确定一个正确的 URL。

增添数据库连通性

如果你的 Web 页面同一个数据库相互作用，你就可以很方便地在你的 Web 项目中建立数据命令，并在页面上建立数据绑定控件。使用拖动和下拉，你可以做到：

- 增添数据链接到你的页面项目。一条数据链接包含连接到特定位置特定数据库所需要的信息。
- 增添数据命令到一条已有的数据链接。一个数据命令要指向一个数据库对象：表或视图、存储的过程、数据库图或查询。
- 在页面上建立数据绑定控件。

关于数据连通性的详细内容，参见第三章“数据库基础”和第十八章“数据库概念”。要了解数据绑定控件的细节，参见第十九章“查看数据”中的“获得记录”和“显示你 Web 页面上的数据”两节。

存储和重复使用 HTML 文本

如果你希望重复使用页面上的任何元素：HTML 文本、脚本、链接或你可选择的任何其它元素，则可以把它们存储到 Toolbox（工具框）中。选择文本，

然后把它拖到工具框上。当你拖动的时候，工具框建立一个新的称为 HTML Fragment 的元素。然后你就可以把这个元素从工具框拖到任何其它页面上。

如果你希望用这种方式存储多个元素，你可以给它们一个有意义的名字，甚至在工具框中为它们建立一个专门的选项卡。

4.3 增添脚本

可以只用文本和 HTML 标记建立基本 Web 页面。然而，如果你希望创建功能完善的数据驱动应用，可以在 Web 页面上增添脚本。

脚本是用户显示你的 Web 页面时运行的程序。这些脚本可以很简单，也可以很复杂，取决于你的需要。你可以把客户脚本和服务器脚本都包括进去。

例如，你可以使用脚本创建下列各种类型的 Web 页面：

- 包含文本、当前时间和日期的页面。
- 显示 Web 站点被访问次数的页面。
- 显示需要用户填充信息的格式，然后返回询问信息或更新数据库的页面。
- 完成数据库操作的页面，其中包括事务处理以及其它复杂的数据管理操作。

可以用不同的脚本编程语言编写脚本。有两种常用的脚本编程语言：Microsoft Visual Basic、Scripting Edition (VBScript) 和 JScript，后者是微软公司实现的 ECMAScript 语言。

你可以选择任何一种你希望用的语言，甚至可在同一个 Web 页面上对不同脚

本使用不同的语言。

Visual InterDev 编辑器帮助你创建的脚本具有下列特点：

- Source 视图，在这里你可以直接编写脚本。
- 为文本配置颜色，这样可以清晰显示脚本语句中的不同元素。
- IntelliSense，只要你键入对象名字，方法名字和特性名字就立刻呈现在你面前，从而帮助你创建无错脚本语句。
- Script Outline Windows（脚本离线窗口），在你的页面中显示客户和服务对象，以及用于编写脚本的事件列表。

注意：你可以使用 Visual InterDev 设计期间控件，为普通任务产生脚本，例如查询数据库、表示输入格式或者显示一个报告。详细内容，参见第十九章“查看数据”、第二十章“修改数据”和第二十四章“用设计期间控件和脚本对象编写脚本”。

关于脚本编程的详细内容，参见第五篇“编辑与脚本编程”，第二十三章“脚本编程概念”中“Web 应用中的脚本”。

编完脚本之后，可以使用内建的调试命令查找脚本中的错误。详细内容，参见第二十三章“脚本编程概念”中“脚本调试过程”和第二十六章“调试你的页面”。

4.4 选用其它编辑器

Visual InterDev 允许选用多种不同编辑器，在项目中加工文件。默认的编辑器有：

- **HTML 编辑器**，用于编辑 Web 页面的布局图并创建脚本。在创建或打开一个 HTML 页面或 ASP 页面的时候，这种 HTML 编辑器便自动打开。
- **样式图表 (style sheet) 编辑器**。在创建或编辑一个级联样式图表 (.css 文件) 的时候，这种编辑器便打开。
- **文本编辑器**。在创建或编辑文本文件时打开此编辑器。

如果愿意，可以使用其它编辑器，可以选用：

- Microsoft FrontPage。如果你已经熟悉 FrontPage，或者希望使用 FrontPage 专有特性的长处，例如 WebBots，就可使用这种编辑器。
- 其它编辑器。

根据需要，你可以选用一种专门的编辑器打开一个页面，或者针对某种页面类型确定默认编辑器。

在一种选定的编辑器中打开 Web 页面

1. 在 Project Explorer 中，用右键单击文件名字，然后选择 Open With。
2. 选择要使用的编辑器。如果希望选择默认编辑器，就选用 Set as Default。
3. 选用 OK。

第五章 演 练

这一章的一些题目通过操作练习，使你了解使用 Visual InterDev 的 Web 应用开发的一般方案。这些演练提供如下内容：

- 指导你完成工作中主要任务的指南。
- 在学习练习中使用的步骤介绍。
- 熟悉 Visual InterDev 的方法手段。
- 指出其它论题的详细信息。

5.1 创建主页

为你的 Web 站点创建主页是熟悉新 Web 编著和开发产品的重要举措。如果你已经知道如何建立导航栏，如何设计页面布局图，以及如何使用 JavaScript 编码，那么这是了解如何使用 Visual InterDev 完成这些任务的一个机会。

在 Visual InterDev 中，创建主页由下列步骤或部分步骤完成：

- 创建 Visual InterDev Web 项目
- 增添主页
- 加入图形
- 使用一致的可视设计

- 在你的页面上增添导航
- 定制主页
- 维护主页

创建 Visual InterDev Web 项目

在开始草拟一个主页之前，必须首先建立一个 Web 项目。一个 Visual InterDev Web 项目把 Web 应用所使用的文件，如图像、文档和 HTML 页面等都组织起来。

在创建一个 Web 项目时，你要为 Web 应用指定一个 Web 服务器和项目名字，还要规定页面布局图和所有页面可视表示样式。

建立 Visual InterDev Web 项目

1. 在 File 菜单中选择 New Project。

出现 New Project 对话框。

2. 在 New 选项卡上，选择左窗格的 Visual InterDev Project，然后选择右窗格的 New Web Project。

3. 在 Name 文本框中键入 Web 项目标名字。

4. 单击 Open。

Web Project Wizard (Web 项目向导) 出现。

Web 项目向导 (Project wizard) 引导你一步步通过选择主 Web 服务器、Web 应用名字、以及默认主样板和 Web 应用布局图的过程。

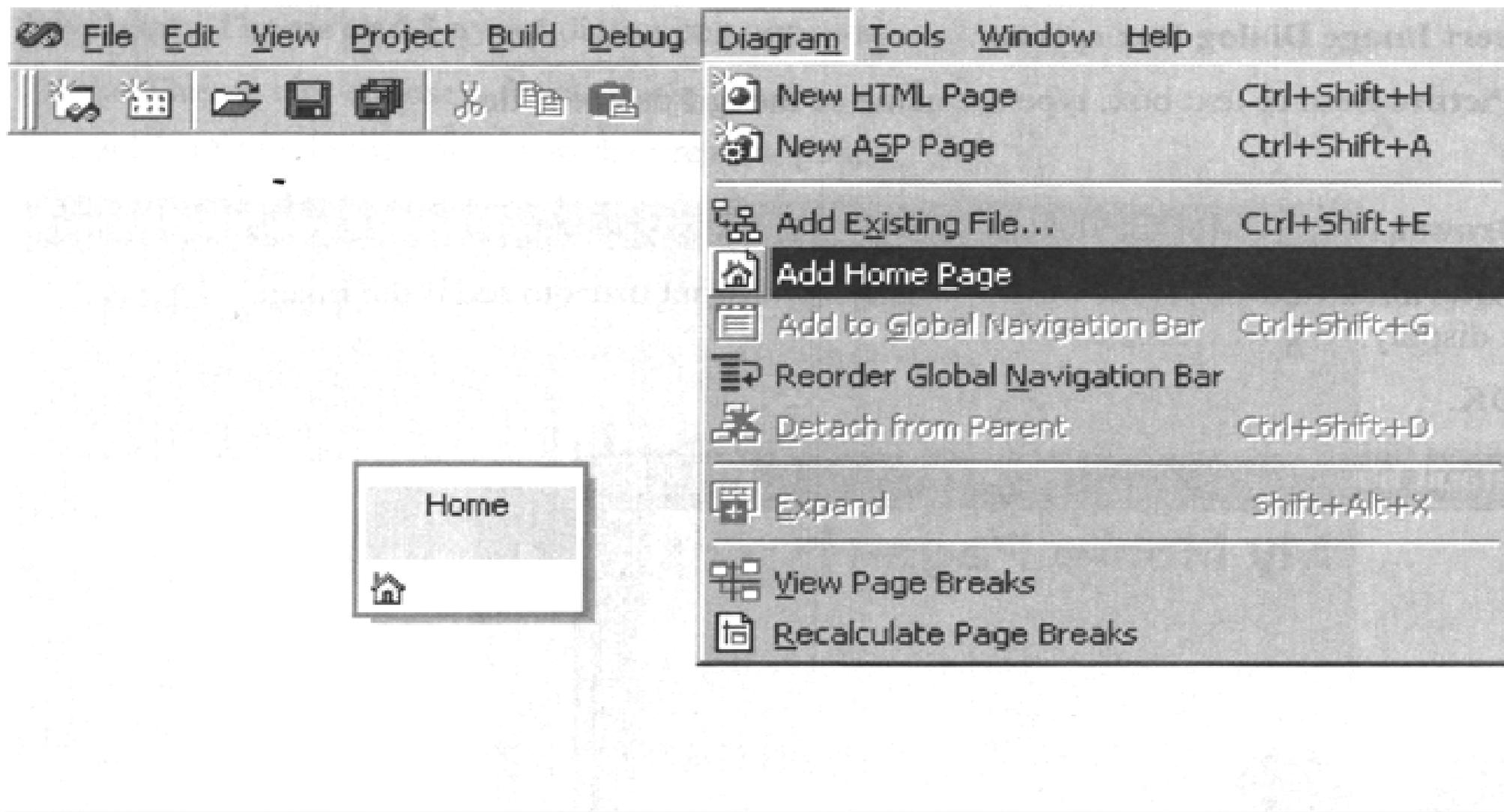
详细内容，参见第一章“Web 项目管理”中的“建立 Web 项目”和第十章“管理 Web 项目”。

增添主页

使用站点图，你很容易把主页添加到你的 Web 应用程序上。一个站点图是一个电子白板，这个白板允许你很快产生一个新的 Web 应用程序，带有以后可定制的框架 HTML 或 ASP 页面。

建立新主页

1. 在 Project 菜单中选择 Add Item。
出现 New 选项卡（Add Item 对话框）。
2. 在 Web Project Files 文件夹中选择 Site Diagram。
3. 在 Name 文本框中键入站点图的名字。
4. 单击 Open。
一个新站点图打开。
5. 在 Diagram 菜单中选择 Add Home Page。
一页图形出现在站点图中，名字为“Home”。



6. 在 File 菜单中选择 Save。

站点图的变更已经保存下来，Visual InterDev 建立一个新主页，称为 Default.htm 或 Default.asp，到底叫什么名字，取决于你使用的 Web 服

务器。

注意：Internet 信息服务器规定主页名字为 default.asp，而个人 Web 服务器规定主页默认名字为 default.htm。

详细内容，参见第十二章“设计 Web 站点”和第十四章“管理站点图”。

添加图形

向 Web 页面上增添多媒体，例如 .gif、.wav 或 .avi 文件可使页面显得有生气。在 HTML 编辑器的 Design 视图中使用 WYSIWYG（所见即所得）技术很容易在主页上添加图形和其它 HTML 元素。

在主页上添加图形

1. 在 Project Explore 中用右键单击你的主页，选择 Open。

注意：Internet 信息服务器规定主页名字为 default.asp，而个人 Web 服务器规定主页默认名字为 default.htm。

2. 在 HTML 编辑器中选择 Design 选项卡。

3. 在 HTML 菜单中选择 Image。

出现 Insert Image Box。

4. 在 Picture Source 文本框中键入 .gif 或 .jpeg 文件名字，

或

单击 Browse 搜索图像文件。

5. 在 Alternate Text 框中，键入在不显示图像时你希望用户看见的信息。

6. 单击 OK。



使用一致性可视设计

为页面建立一致性可视显示，使你的 Web 站点具有专业化设计的印象。你可以在 Web 应用程序的各个页面使用一个主样板（theme），从而保持可视一致性。主样板可以施加于整个设计，也可以施加于单个页面。

使用主样板建立一致的可视外观

1. 在 Project Explorer 中选择你的主页

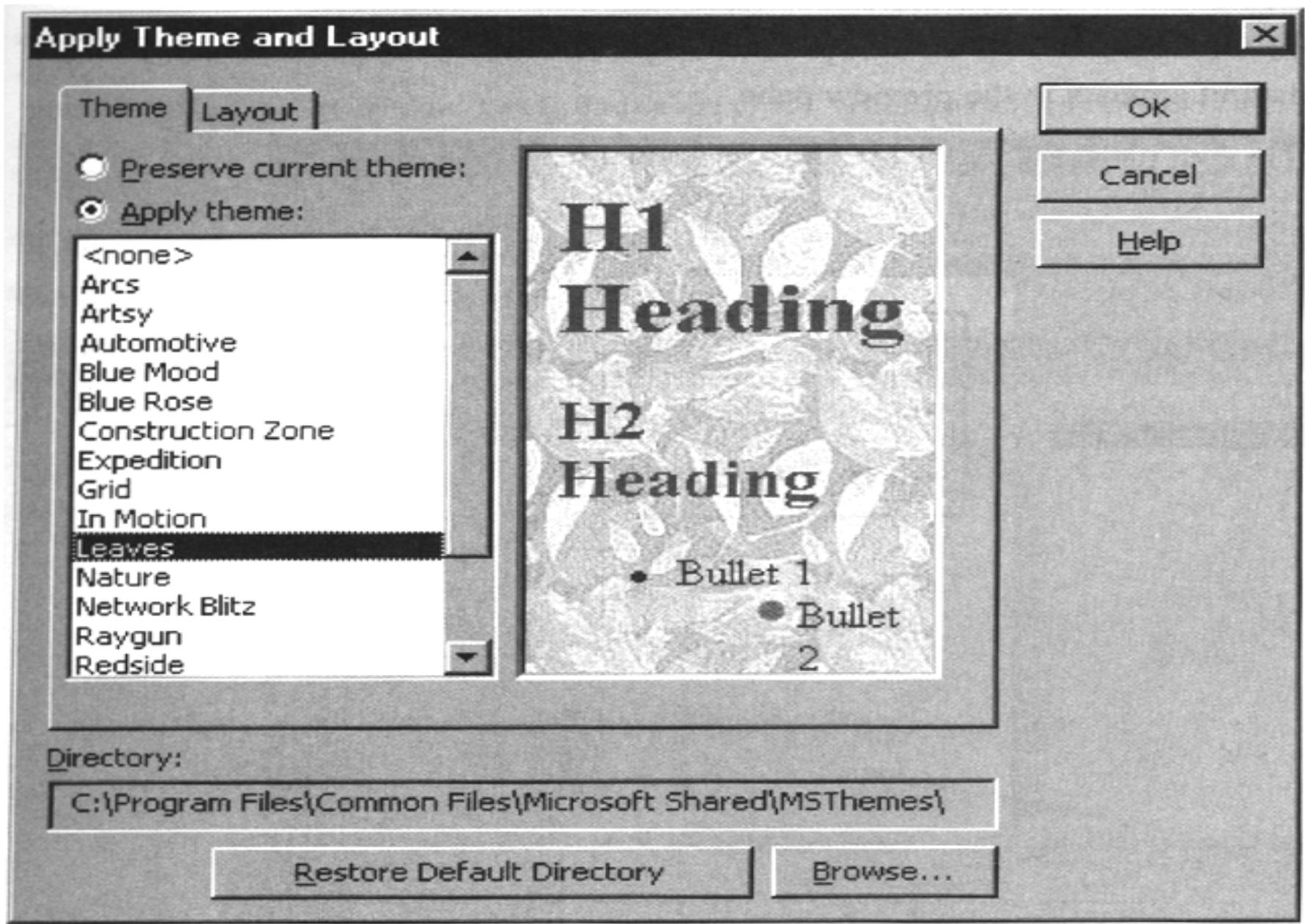
注意:Internet 信息服务器规定主页名字为 default.asp, 而个人 Web 服务器规定主页默认名字为 default.htm。

2. 在 Edit 菜单中选择 Apply Theme and Layout。

出现 Apply Theme and Layout 对话框。

3. 在 Theme 选项卡中选择主样板名字。

一个主样板图形出现在预览窗格中。



4. 单击 OK。

详细内容，参见第十一章“站点设计”中的“站点一致性”和第十七章“页面布局定制”。

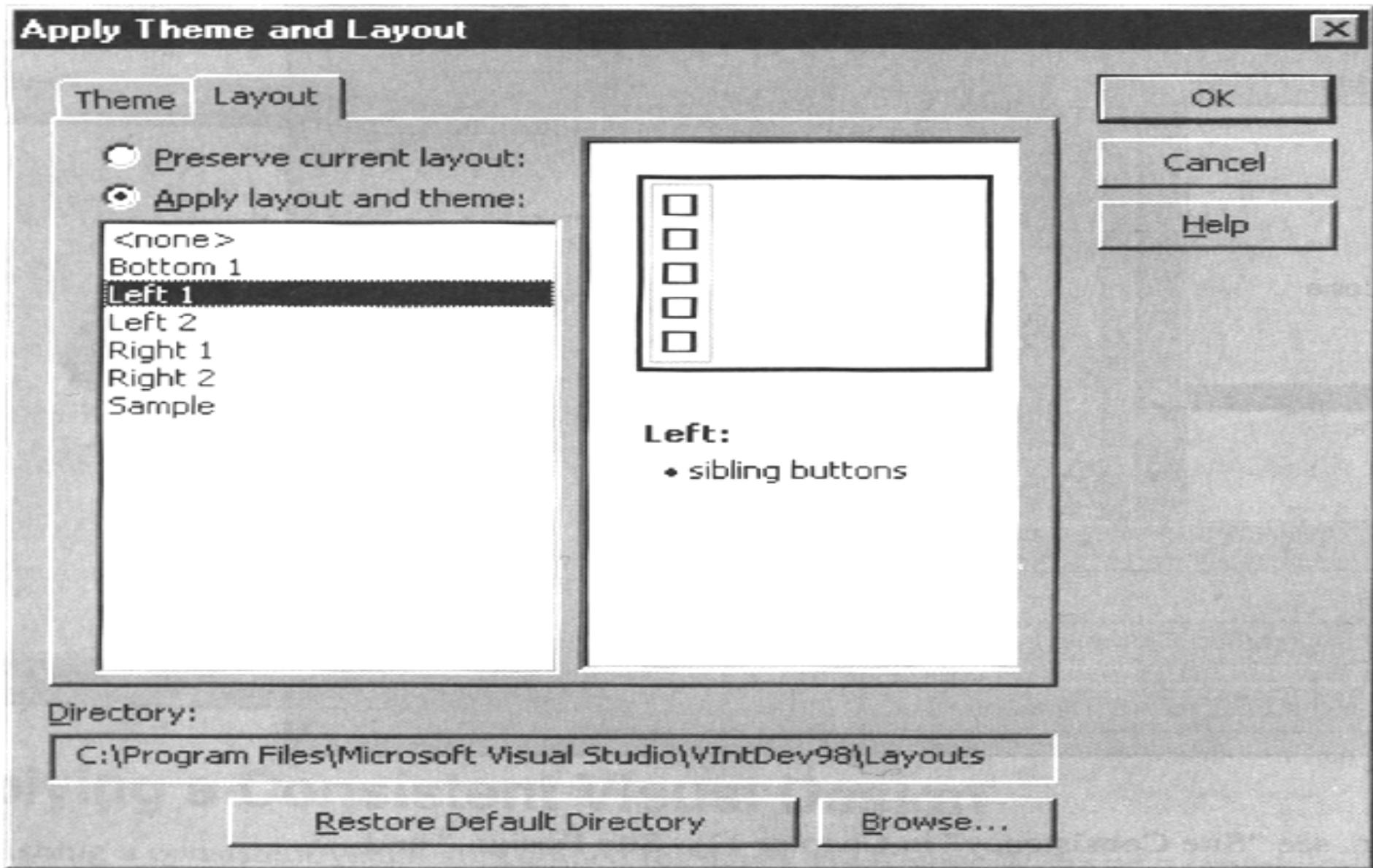
为页面添加导航

导航栏 (Navigation bar) 包含一些链接，这些链接允许你的用户在你的 Web 站点周游。保持导航的一致性使得用户浏览你的站点变得容易。你可以使用布局图，或使用一个设计期间控件与站点图，自动生成导航栏。

在使用布局图或使用 PageNavbar 控件的时候，你可以使用站点图设计并维护 Web 页面的导航结构。然后 Visual InterDev 使用你设计的导航结构在导航栏中为你的页面生成链接。

在页面上添加布局图

1. 在 Project Explorer 中选择你的主页。
2. 从 Edit 菜单中选择 Apply Theme and Layout。
出现 Apply Theme and Layout 对话框。
3. 在 Layout 选项卡中选择布局图名字。
一个布局图形出现在预览窗格中。



4. 单击 OK。

关于站点导航的详细内容，参见第十三章“设计站点导航”。

定制主页

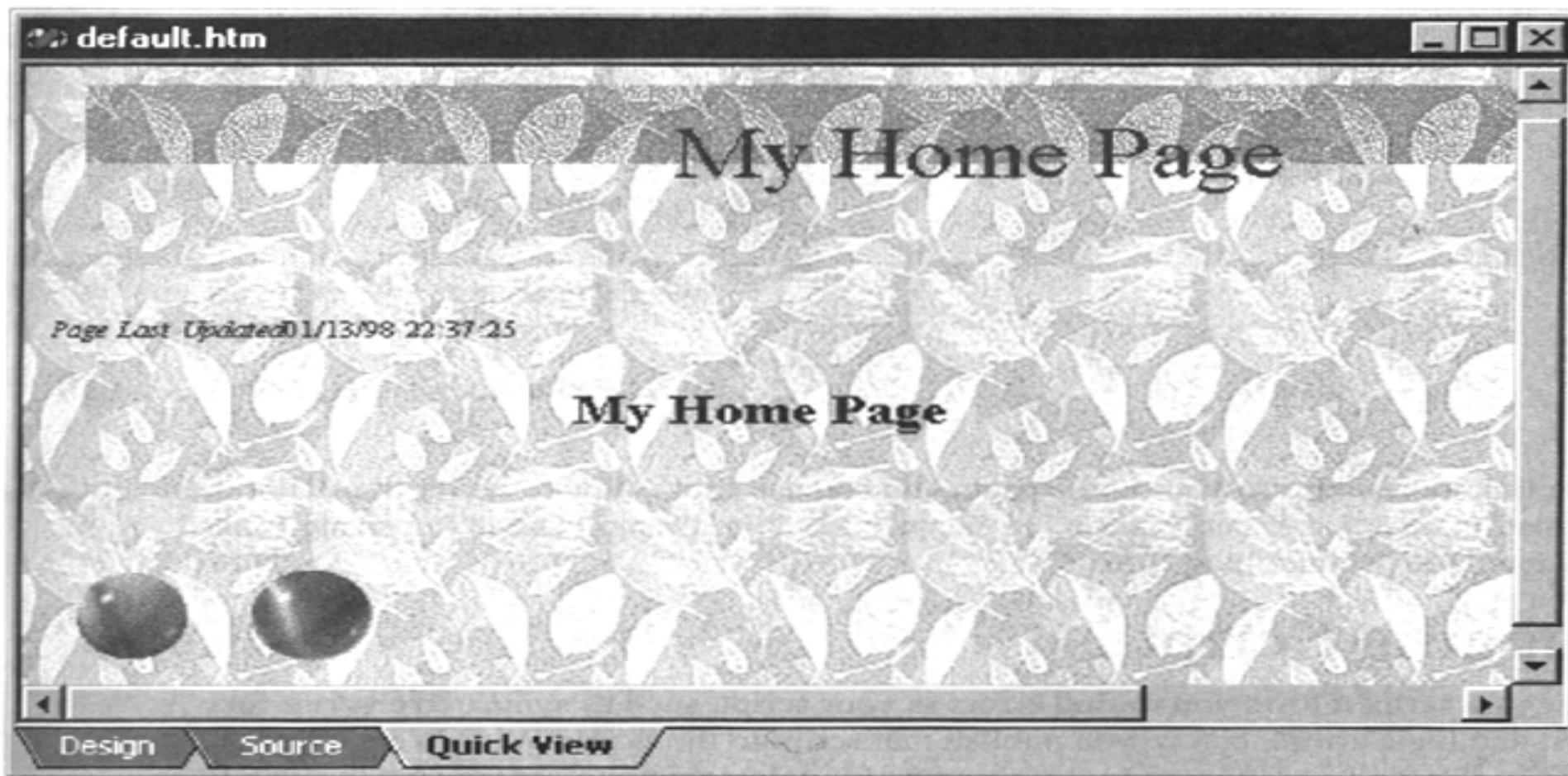
一旦使用内建特性设计并建立起主页基本结构，就可以很容易地定制主页。使用 HTML 文本和标记为你的页面创建内容。还可以使用脚本完成增强 Web 页面的任务。例如，可以使用客户方脚本产生文本，告诉观众该主页最后一次更新在是什么时候。

添加更新消息

1. 在 Project Explorer 中用右键单击主页并选择 Open。
该主页在 HTML 编辑器中打开。
2. 在 Source 选项卡上，在 <BODY> 标记之后键入下列代码

```
<Script language="JavaScript">
<!--
document.write("<I>Page      Last      Updated:</I>"      +document.
lastModified);
// -->
</Script>
```

3. 在 Standard 工具栏中单击 Save。
4. 要查看最后更新的文本，单击 Quick View 选项卡。



详细内容，参见第四章“编辑基础”中的“添加脚本”和第二十五章“用 HTML 元素编写脚本”。

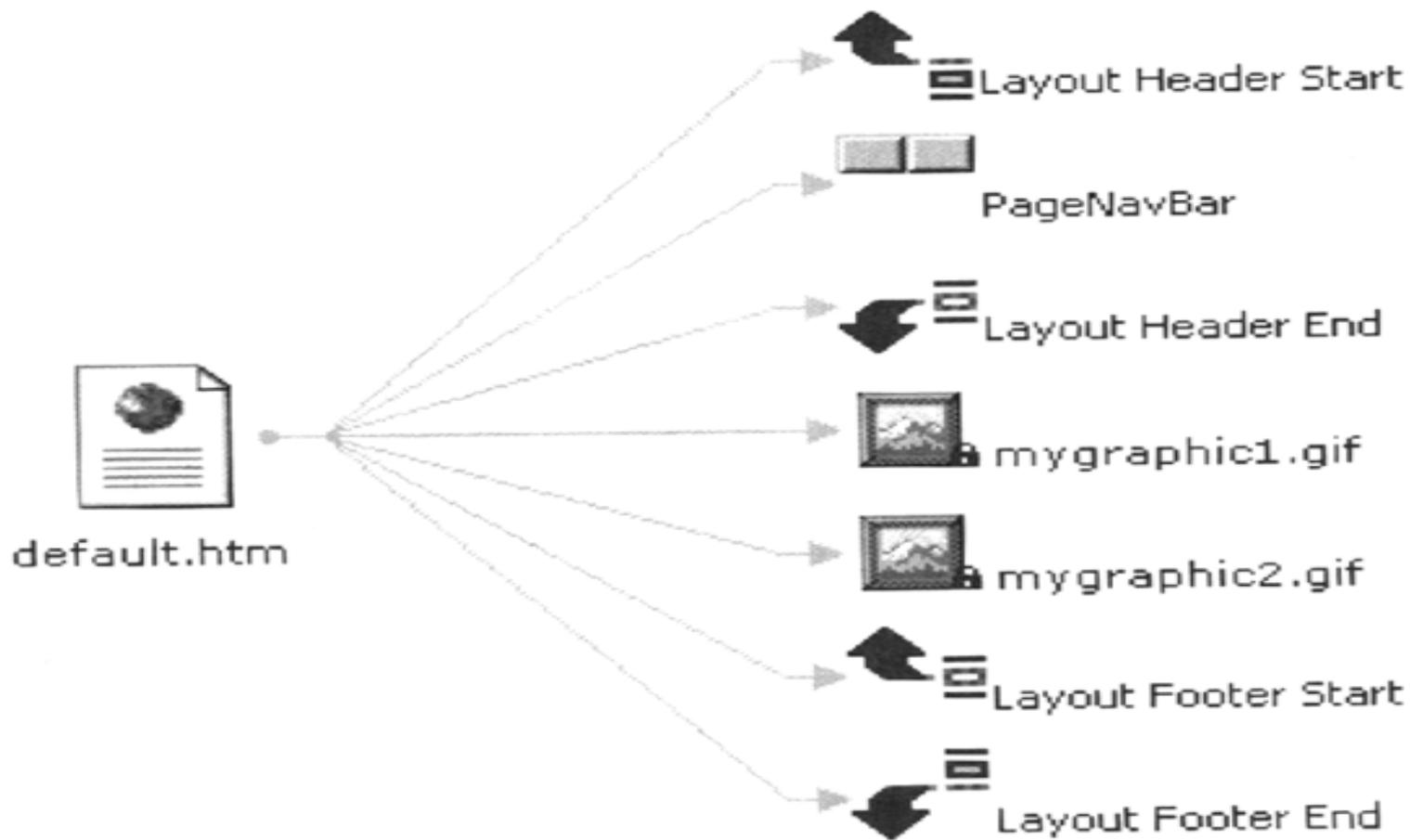
维护主页

主页一旦建立起来，任务便转变成维护主页。对你的用户来说，链接的正确工作是很重要的。你可以使用链接视图维护主页中各项目之间的链接。你可

以快速查看链接是否已经断开，读到链接错误。

查看主页的链接

- 在 Project Explorer 中用右键单击主页并选择 View Links。出现一个链接图，在图的中间部位显示主页。这个链接图可以过滤以显示不同类型的链接，从而使得管理链接变得容易。



如果你把更多的项目链接到主页上，则可以使用 Link View 来检验它们。详细信息，参见第十一章“站点设计”中的“链接检验”和第十六章“维护链接”中的“维修链接”。

5.2 调 试 脚 本

调试脚本可以使你在把脚本发行给 World Wide Web（万维网）之前，发现脚本中的错误，例如语法错误、运行期间错误以及逻辑错误。你大概已经知道为调试设置断点，知道如何一步步通过一行行的脚本。现在你可以看到如何使用 Visual InterDev 完成这些任务。在 Visual InterDev 中，你把脚本放到页面上进行调试，调试脚本包括下述步骤或部分步骤：

- 把客户脚本加到一个 HTML 页面上
- 设置断点
- 一步步通过脚本的各行
- 改变变量的值

把客户脚本加到一个 HTML 页面上

调试客户脚本是熟悉在 Visual InterDev 环境中进行调试的轻松方式。客户方脚本在浏览器中进行处理，而服务器方脚本在 Web 服务器中进行处理。要了解客户和服务脚本的详细内容，参见第二十六章“调试页面”中的“调试客户脚本”和“调试服务器脚本”两节。

要进行演练，你需要创建一个文件，其中带有要调试的脚本。有一个样板

客户脚本，在下述例子中放到<SCRIPT>和</SCRIPT>两个标记之间，以简单格式设定了长度合适的口令项。其余的 HTML 标记定义格式，但不特别指定脚本调试过程。

要使用下述脚本，就要建立一个新的.htm 文件，并把 HTML 编辑器转换到 Source 视图中。复制脚本和 HTML，然后单击右键，再选择快捷菜单中的 Paste As Text（按文本粘贴）。详细内容，参见第二十五章“用 HTML 元素编写脚本”。

```
<SCRIPT LANGUAGE="JavaScript">
function validatePassword()
{
    var password;
    password = document.frm1.txtPassword.value;
    // debugger;
    alert("You entered " + password);
    if (password.length < 4) {
        alert("You must enter 4 characters or more!");
        document.frm1.txtPassword.select();
        return false;
    }
    else {
        alert ("Your form is being submitted!");
        return true;
    }
}
</SCRIPT>
<P><H2>Please enter a password.</H2>
<P>
<FORM NAME="frm1" METHOD="Post" ACTION="Process.asp" OnSubmit="return validatePassword()">
    <INPUT NAME="txtPassword" TYPE="Password">
    <INPUT NAME="btnSubmit" TYPE="Submit" VALUE="Submit">
    <INPUT NAME="btnReset" TYPE="Reset" VALUE="Reset">
</FORM>
```

注意:JavaScript 区分大小写, 因而上述样板脚本出现在你的 .htm 文件中, 严格保持原样。

在用户单击 Submit (提交) 按钮时, 脚本核实口令长度是否是四个字符或更多字符。脚本产生一消息框, 列出用户键入的口令。如果口令的长度为四个字符或更多字符, 用户便用<FORM>标记中的 Action 元素重新定向到该页面上。

如果你还没有建立 Process.asp, 文件便调用格式, 而且浏览器将显示“对象没找到”错误。如果口令长度少于四个字符, 便出现一个消息框, 指明要求多加字符。



设置断点

要在脚本中指定一个你希望停下来位置，查看一下过程状态，这时你就要使用断点。断点是脚本中的一行，你希望应用程序在这里停下来。然后你可以一步一步进行，或一步跨过几行脚本，以便查找差错。在 Source 视图中，断点在脚本行的左边显示一个红色八角形。

注意：要在微软 Internet Explorer 中调试客户脚本，必须使用 Internet Explorer 4.0，调试功能必须打开。详细信息参见 Internet Explorer 文档。

设置断点

1. 打开 .htm 文件，选择 Source view。
2. 把光标放到脚本中你希望建立断点的行上。
3. 在 Debug 菜单中选择 Insert Breakpoint。
一个红色八角形出现在脚本该行在左边。

注意：你还可以把光标放到脚本的一行上，按 F9，用这个办法设置或去掉断点。

启动调试软件

1. 在 Project Explorer 中用右键单击 .htm 文件，选择 Set As Start Page。
2. 启动 Internet Explorer 4.0，把 .htm 页面装载到浏览器中。
3. 在 Visual InterDev 的 Debug 菜单中选择 Processes。
出现 Processes 对话框。
4. 在 Process 区域选择 Microsoft Internet Explorer，然后再选择 Attach。
在 Debugged Processes 区域出现 Microsoft Internet Explorer 和你计算机的名字。
5. 在 Internet Explorer 中键入口令，选择 Submit（提交）。
脚本开始执行，直到 Internet Explorer 遇到断点为止。当 Internet Explorer 遇到断点时，便停下来，在 Souece 视图中显示源脚本。

覆盖在断点图标上的黄色箭头指示调试程序在脚本的这一行上停下来。
详细内容，参见第二十六章“调试页面”中的“调试客户脚本”一节。

一步步通过每行脚本

要一次执行一行脚本，你可以使用 Step Info。一步步通过每行脚本之后，你还可以在 Internet Explorer 中浏览页面，查看每个语句的效果。

步进每行脚本

1. 在 Source 视图中打开 .htm 文件，在下述一行脚本上设置断点：

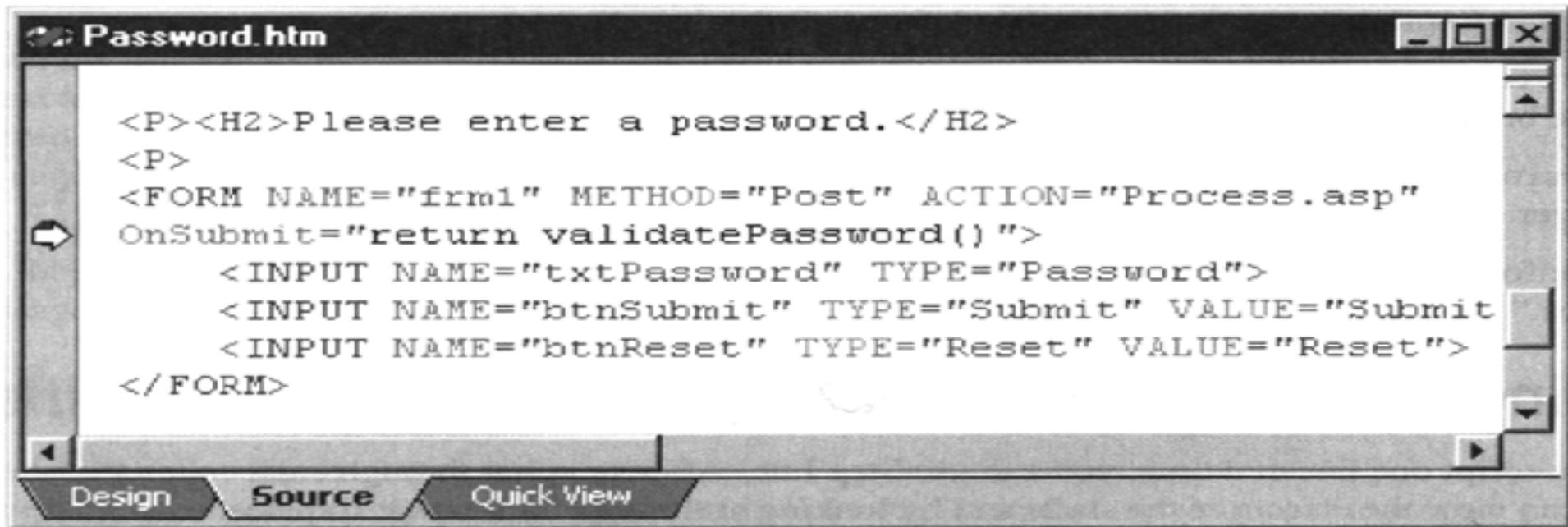
```
<FORM NAME= " frm1 " METHOD= " Post " ACTION = " Process.asp "  
OnSubmit= " return validatePassword() " >
```

```
<P><H2>Please enter a password.</H2>
<P>
<FORM NAME="frm1" METHOD="Post" ACTION="Process.asp"
OnSubmit="return validatePassword()">
  <INPUT NAME="txtPassword" TYPE="Password">
  <INPUT NAME="btnSubmit" TYPE="Submit" VALUE="Submit"
  <INPUT NAME="btnReset" TYPE="Reset" VALUE="Reset">
</FORM>
```

注意：当你把一个断点放到上述一行 HTML 上，实际上你规定了在单击 Submit 按钮时停止执行脚本，OnSubmit="return validatePassword()"。

2. 启动调试程序。

3. 键入两个字符的口令，选择 Submit。



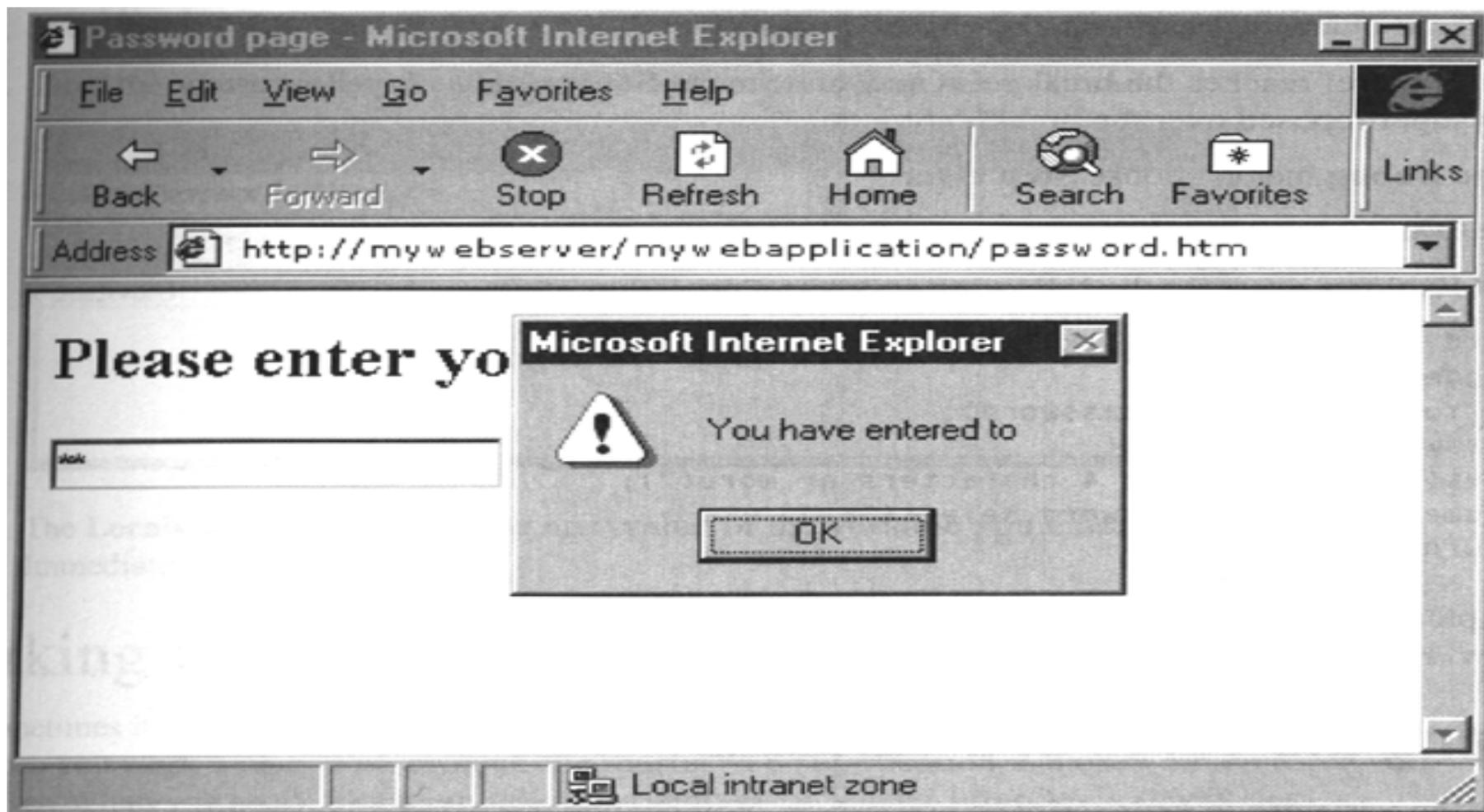
Internet Explorer 到达断点，转换到 Source 视图中。出现一个黄色箭头，覆盖在断点图标上。

4. 在 Debug 菜单中选择 Step Into，然后再选择一次 Step Into。

下面一行脚本变亮：

```
alert(" You entered " + password);
```

5. 在 Debug 菜单中再一次选择 Step Into。



Internet Explorer 执行提示脚本，并显示一个消息框。

6. 选择 OK。

Internet Explorer 到达下一行脚本，转换到 Source 视图中。

你可以继续一步步执行每行脚本，查看在 Internet Explorer 中执行的结果。

要使脚本过程作为一个单元来执行，你可以使用 Step Over(跨越)。Step Over 允许你跳过一个过程或函数，以及跳过后边的过程和函数。

在跨越一个过程之后，你可以通过在 Internet Explorer 中观察页面的办法来观察过程的效果。

跨越每行脚本

1. 在 Source 视图中打开 .htm 文件，在下述一行脚本中设置断点：

```
<FORM NAME= " frm1 " METHOD= " Post " ACTION = " Process.asp "  
OnSubmit= " return validatePassword () " >
```

注意：当你把一个断点放到上述一行 HTML 上，实际上你规定了在单击 Submit 按钮时脚本停止执行，OnSubmit=“return validatePassword()”。

2. 启动调试程序。

3. 键入三个字符的口令，选择 Submit。

Internet Explorer 到达断点，转换到 Source 视图。出现一个黄色箭头，覆盖到断点图标上面。

4. 在 Debug 菜单中选择 Step Over。

5. 在每个提示消息中选择 OK。

Internet Explorer 把下述脚本作为一个过程单元来执行：

```
password = document.frm1.txtPassword.value;
//debugger;
alert("You entered " + password);
if (password.length < 4){
    alert("You must enter 4 characters or more!");
    document.frm1.txtPassword.select();
    return false;
}
```

Internet Explorer 到达该过程单元后边的一行脚本，转换回 Source 视图。要了解详细内容，参见第二十六章“调试页面”。

改变变量值

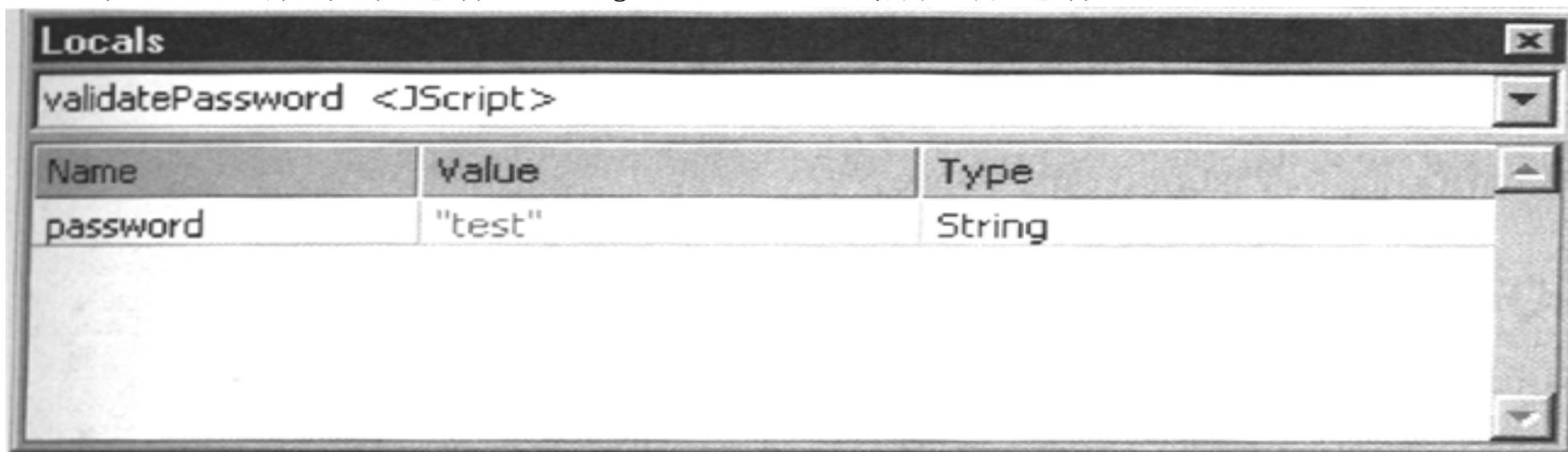
就像调试那样，你可以为你的脚本分配变量。你可以在 Immediate 窗口中改变这些变量值。当 Internet Explorer 到达一个断点并转换到 Source 视图时，你就可以改变这些变量值，并在 Locals 窗口中观察结果。

改变变量值

1. 在 Source 视图中打开 .htm 文件，在下述一行脚本中设置断点：
<FORM NAME= " frm1 " METHOD= " Post " ACTION = " Process.asp "
OnSUBMIT=" return validatePassword () " >
2. 启动调试程序。
3. 键入三个字符的口令，选择 Submit。

Internet Explorer 到达一个断点，并转换到 Source 视图。出现一个黄色箭头，覆盖到断点图标上面。

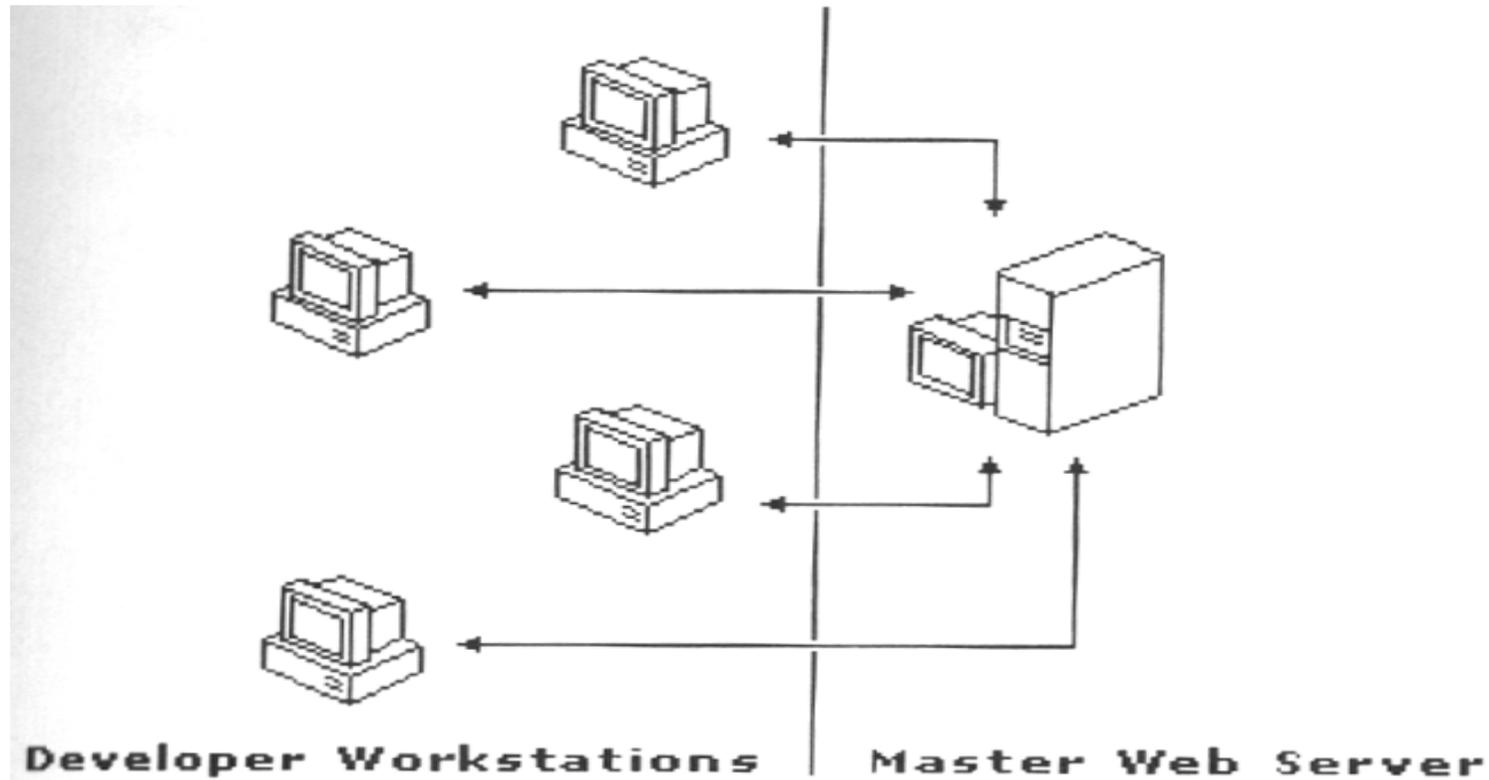
4. 在 Visual InterDev 中的 Debug 菜单中选择 Step Into。
5. 在 View 菜单中选择 Debug Windows，然后再选择 Immediate。
6. 在 Immediate 窗口中键入下述代码：
password= "test"
7. 按 Enter。
8. 在 View 菜单中选择 Debug Windows，然后再选择 Locals。



Locals 窗口显示你在 Immediate 窗口中键入的新变量值。

5.3 同多个开发者一同工作

创建和维护 Web 应用有时候不只一个开发者。当你在 Web 项目中工作时，主 Web 应用存储在主 Web 服务器中。当你在一个小组中工作，为发挥 Visual InterDev 开发者隔离特性的长处，小组各个成员应使用主应用的本地副本，以本地方式进行工作。这样就可以使得多个开发者从本地 Web 项目中访问同一个主 Web 应用。



除了以本地方式工作之外，为了便于以小组为基础对 Web 应用进行开发，

可以使用 Visual InterDev 的内建源控件特性。

在这个演练中，要讨论下述方案：

- 使用源控件特性或 Visual InterDev
 - 创建主 Web 应用
 - 共享主 Web 服务器中的 Web 应用文件
- 用 Visual SourceSafe 控制 Source（源）。

使用 Visual InterDev 的源控件特性

Visual InterDev 内建源控件特性允许多个开发者同时在一个文件上工作。当不同文件版本发生冲突时，Visual InterDev 源控件可以帮助你解决这些差别。

Visual InterDev 允许多个开发者建立他们自己独有的可以访问同一个主 Web 应用的本地 Web 项目，而主 Web 应用位于主 Web 服务器中。每个开发者都可以得到文件的工作副本，编辑这些副本，然后更新主服务器。

如果你有 Visual SourceSafe 一个副本，你可以使用它控制你的源，而不是控制 Visual InterDev 的内建源控件特性。Visual SourceSafe 将在本章稍后“用微软 Visual SourceSafe 控件源”讨论。

创建主 Web 应用程序

你可以在主 Web 服务器中创建主 Web 应用，然后让你们开发组的其它成员共享它。

创建主 Web 应用

1. 在 File 菜单中选择 New Project。出现 New Project 对话框。
2. 在 New 选项卡中左窗格选择 Visual InterDev Projects，然后在右窗格选择 New Web Project。
3. 在 Name 文本框中为 Web 项目键入一个名字。
4. 在 Location 框中，为新本地项目键入你计算机的位置。
注意：名字和位置设置只适用于你的本地项目。这些设置不保存到服务器中，因而对 Web 应用的名字和 Web 站点名字没有影响。
5. 单击 Open。

出现 Web Project Wizard (Web 项目向导)。

步骤 1：规定服务器和方式

规定你的主 Web 服务器名字并选择 Local (本地) 方式。对在多个开发者环境中进行工作的每个开发者来说，本地方式是典型的推荐方式。这样可以允许开发者独立于主 Web 应用编辑与测试文件。

步骤 2：指定你的 Web

选择 Create a New Web application (创建一个新 Web 应用)，并键入名字。

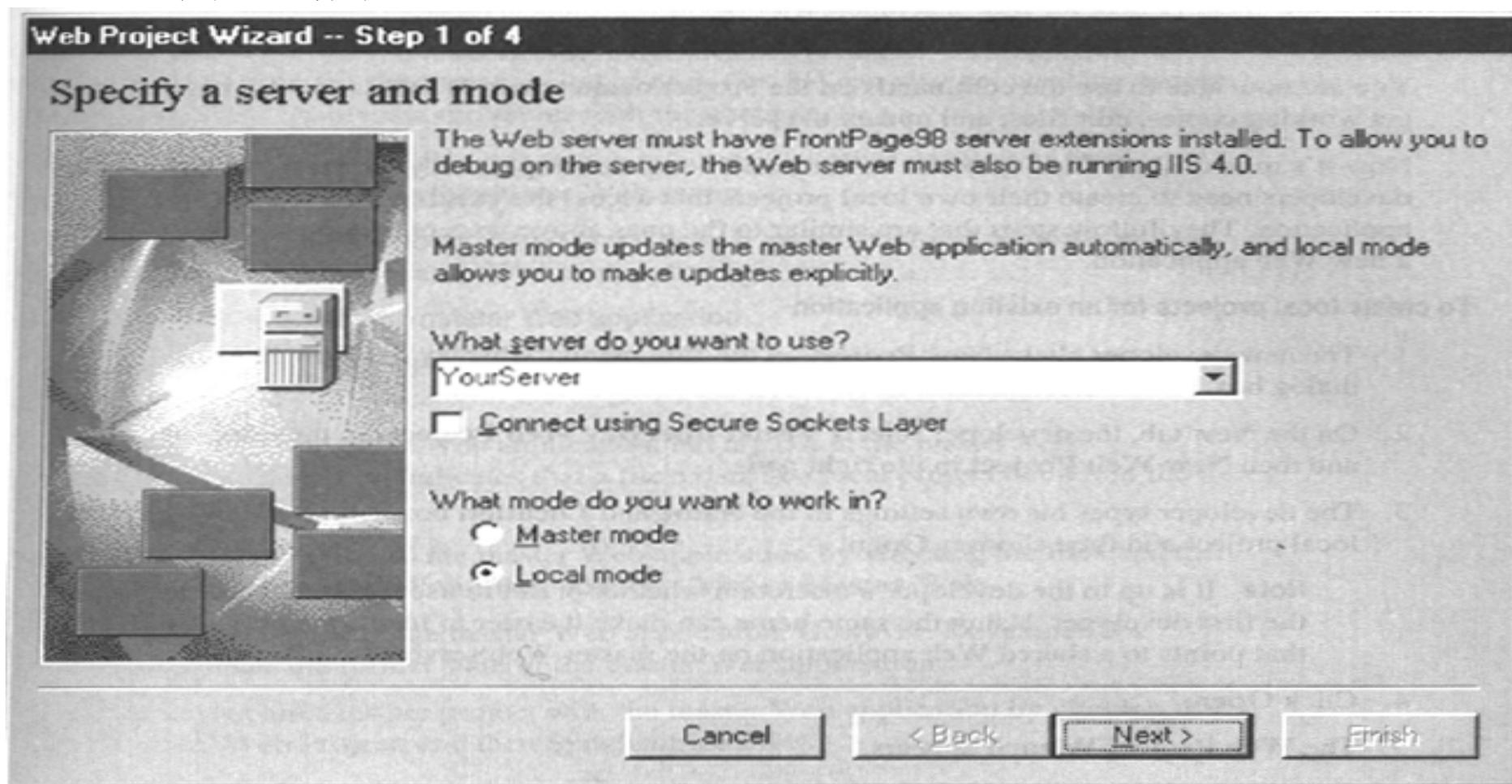
步骤 3：添加布局图

不一定需要使用布局图。如果需要，现在可以选择 <none>，过一会儿再添加布局图。

步骤 4：添加主样板

不一定需要主样板。如果需要，现在可以选择 <none>，过一会儿再

添加主样板。



6. 单击 Finish。

Visual InterDev 在主 Web 服务器上创建新 Web 应用的同时，也可以在你的计算机上创建一个新的本地项目。在新项目创建起来之后，它显示在 Project Explorer 上你的解决方案中。现在你可以使用 Project 菜单中的命令添加 HTML 和 ASP 页面，获得工作副本，编辑文件以及更新服务器。

现在到了你们小组的一个或多个其它开发者共享文件的时候了。所有新开发者都需要建立他们自己可以访问已有主 Web 应用的本地项目。他们所要执行的步骤与上述步骤类似，所不同的只是他们不需要创建新的 Web 应用程序。

为已有应用创建本地项目

1. 新开发者单击 File 菜单中 New Project，显示 New Project 对话框。
2. 在 New 选项卡中，开发者选择左窗格的 Visual InterDev Web Projects，再选择右窗格的新 Web Project。
3. 开发者在 Name 和 Location 框中为本地新项目键入他自己的设置，然后选择 Open。

注意：由开发者自己决定是否与第一个开发者使用相同的名字。使用相同的名字，使标识一个项目指向主 Web 服务器上的共享 Web 应用变得很容易。

4. 单击 Open。

出现 Web Project Wizard。

步骤 1：规定服务器和方式

新开发者要规定与第一个开发者相同的服务器，还要选择本地方式。

步骤 2：指定你的 Web

新开发者要选择 Connect to an existing Web application <server name> 选项。

下拉列表包含指定服务器上所有可用的 Web 应用。新开发者要从中选择相应的 Web 应用。

5. 单击 Finish。

Visual InterDev 访问主 Web 服务器上的主 Web 应用，并在开发者计算机上创建新的本地项目。此后该开发者就可以在 Internet Explorer 中看到新项目。现在开发者就有权访问主 Web 服务器上 Web 应用的所有文件。

关于使用 Web 应用本地副本进行工作的详细内容，参见第七章“本地工作”和第六章“Web 项目概念”中的“项目结构”一节。

共享主 Web 服务器上 Web 应用文件

现在你和其他开发者可以访问主 Web 服务器上的相同文件。每个开发者都可以得到每个文件的最新版本。他们还可以得到工作副本，在本地编辑这些副本，然后用变更的副本更新服务器。

Visual InterDev 的源控件特性防止你和其他开发者在主 Web 服务器上相互覆盖工作成果。

在下面的方案中，开发者 A 和开发者 B，每个人都在自己的计算机上有自己的本地项目，都可以访问同一个主 Web 应用。

本地项目与主 Web 应用同步

1. 开发者 A 在 Project 菜单上选择 Add Item，把多个文件加到他的项目上。这些文件现在都在他的本地 Web 应用中，而不在主 Web 应用中。这个 flag 图标指明一个文件在本地项目中，不在主项目中。
2. 开发者 A 选定一些文件，把这些文件加到主 Web 应用中，然后单击 Project 菜单、Web 文件，再单击 Add to Master Web（加到主 Web）。

这些新文件复制到主 Web 应用中。然而，开发者 B 的项目也反映出前者的主 Web 应用状态。

3. 开发者 B 单击 Project 菜单、Web 项目，再单击 Synchronize Files，使他的项目同主 Web 应用同步起来。

提示：当与其他开发者一起工作的时候，同你的本地项目经常保持同步是个好主意。

开发者 B 的本地项目现在反映的改变是由开发者 A 对主 Web 服务器产生的。

解决文件之间的冲突

1. 开发者 A 希望在 Filename.htm 的本地副本上工作。他选择了 Project Explorer 中的文件，选用 Project 菜单上的 Web File，然后他再选用 Get Working Copy。接着他开始编辑该文件。
2. 开发者 B 也希望用 Filename.htm 进行工作。她选择了 Project Explorer 中的文件并单击 Project 菜单、Web Files，然后单击 Get Working Copy。尽管开发者 A 已经获得了该文件的工作副本，Visual InterDev 还是允许开发者 B 也获得工作副本。
3. 开发者 A 已经保存了文件，希望用他的最新版本更新主 Web 应用。他单击 Project 菜单，Web Files，再单击 Release Working Copy，于是主 Web 应用更新，反映了开发者 A 所做的改变。
4. 现在开发者 B 也已经保存了她的改变，并结束该文件。她单击 Project 菜单、Web Files、再单击 Release Working Copy。

Visual InterDev 把开发者 B 的工作副本同主 Web 应用的版本进行比较，

并归并冲突。于是显示出 Merge（归并）对话框，要求解决冲突。

5. 开发者 B 可以看到所有冲突，并进行选择来解决冲突。冲突解决之后，主 Web 服务器用解决冲突的文件版本进行更新。
6. 开发者 A 在 Project Explorer 中的文件列表现在已经同主 Web 应用失去同步。要在他本地 Web 应用中获得最新版本，他需要选择 Project 菜单中的 Web Files，然后选择 Get Latest Version。

有关同其他人在 Web 应用文件上一起工作的详细内容，参见第十章“管理 Web 项目”和第七章“本地工作”。

Visual SourceSafe 对源进行控制

使用 Visual SourceSafe，你就可以扩充 Visual InterDev 的内建源控件能力，包括版本控件、滚回、及其它源控件特性。

使用 Visual SourceSafe 和 Visual InterDev

1. 为了与主 Web 应用一起使用 Visual SourceSafe，你首先要把它安装到主 Web 服务器上。
因为 Visual InterDev 同 Visual SourceSafe 兼容，就不需要在客户机（你的本地计算机）上安装 Visual SourceSafe。
有关安装 Visual SourceSafe 的详细内容，参见第八章“同多个开发者一起工作”中的“在 Web 服务器上组建源控件”一节。
2. 在服务器上安装 Visual SourceSafe 之后，你需要在主 Web 服务器上把它置于开启状态。只要有一个开发者在主 Web 服务器上使它处于开启状

态，其他开发者就必须刷新他们的本地项目，以便利用源控件的长处。

为了使 Visual SourceSafe 作为源控件，参见第八章“同多个开发者一起工作”中的“把源控件加到 Web 应用上”一节。

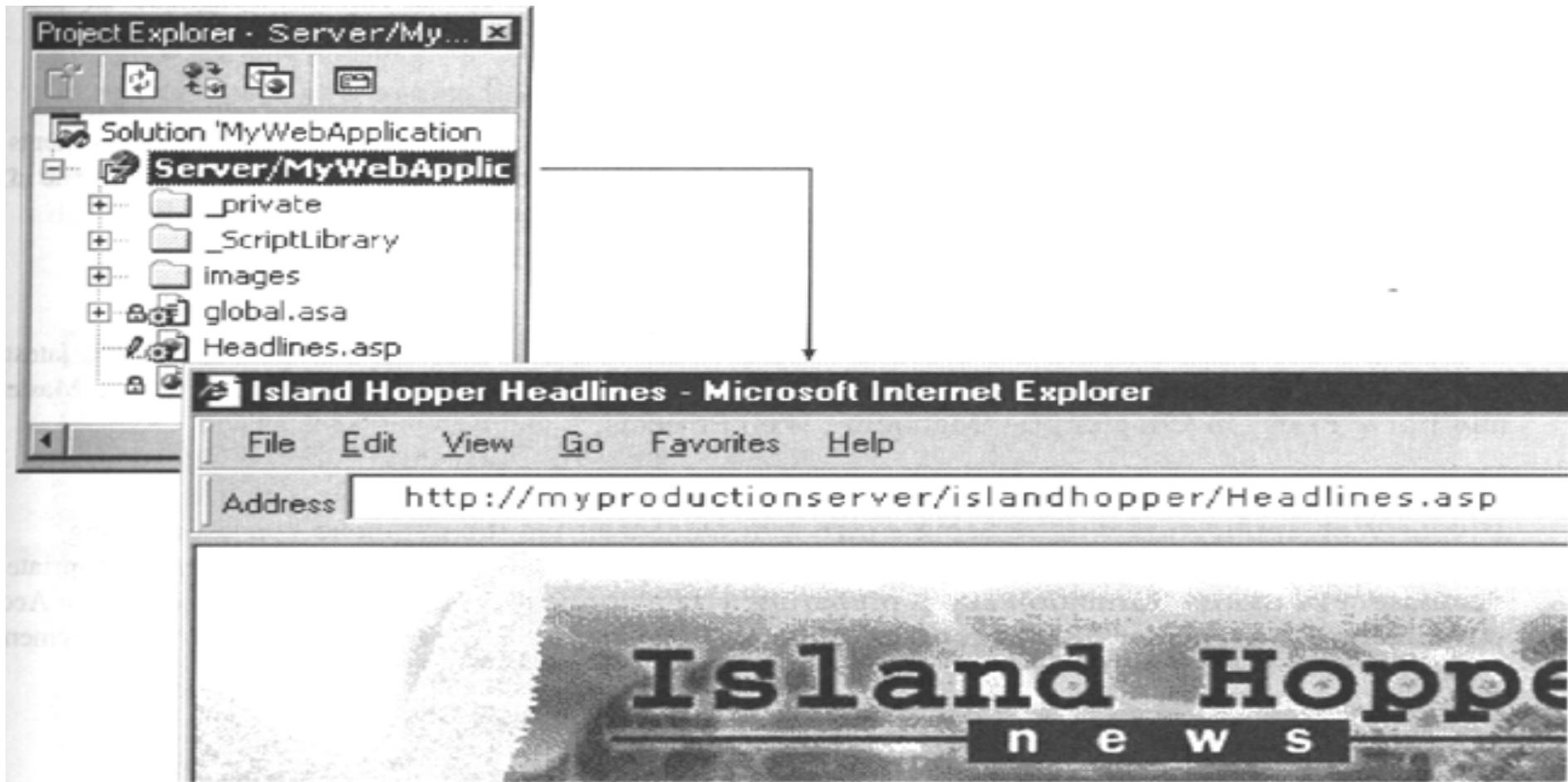
现在，取代“获得”和“刷新”文件的是“check out”（检出）和“check in”（检入）文件。

有关使用 Visual SourceSafe 详细内容，参见第八章“同多个开发者一起工作”和 Visual SourceSafe 文档。

5.4 部署 Web 应用程序

部署一个 Web 应用程序就要在另外一个位置把你的 Web 应用程序建立一个副本。例如你可能希望建立一个应用供产品服务器上的最终用户随时使用，或者把一个副本加到档案中。当你部署一个 Web 应用程序，而且是同你的主 Web 应用程序分开部署的，这时就需要建立它的副本。

一旦应用程序开发完毕，Visual InterDev 使得部署变得很容易。你只需把 URL 指定给该应用程序，Visual InterDev 处理把该应用程序复制到另一个服务器的细节。



从项目到发布 Web 应用程序，部署都用到你的应用程序

在上图中已开部署的 URL 显示出产品 Web 服务器是 islehop1，应用程序的名字是 myproductionserver，而且应用程序的开始页是 Headline.asp。关于标识开始页的更多内容，参见 Web 服务器文档，例如 Internet 信息服务器（IIS）。

在 Visual InterDev 中，部署由下述步骤或部分步骤构成：

- 部署准备

- 部署 Web 服务器软件
- 证实生成服务器内容

部署准备

在进行部署之前证实一些事情可使部署变得容易。关于预览和调试的详细内容，参见第二章“Web 基础”的“预览页面”和第二十六章“调试页面”。

部署核对表



链接工作吗？

有时候链接可能是工作的，但是跳到的页面不对。你可能希望证实你所有链是否都跳到相应的页面上。要了解关于证实链接的详细信息，参见第十六章“维护链接”中的“维修链接”和第十五章“按项查看链接”中的“按项目查看链接”。



应用使用的所有文件都包括在 Web 项目中吗？

你可以包括应用程序所涉及的任何文件。例如，如果你提供文档供下载，如用户指南和产品信息，则可以把这个文档文件包括在你的 Web 项目中。详细内容，参见第一章“Web 项目管理”中的“添加文件”一节。



主 Web 服务器上的文件都更新了吗？

如果你的工作副本还没有全部发出，主 Web 服务器就可能没有你的最新版本。关于更新主 Web 服务器的详细内容，参见第十章“管理 Web 项目”中的“使主文件与本地文件同步”。

续表

☑ 你的数据连接指向产品数据库吗？

如果 Web 应用程序所使用的数据库不是在你开发期间使用的数据库，你就要确保你项目中的数据连接指向相应的数据库。关于为部署准备数据连接的详细内容，参见第十八章“数据库概念”中的“数据访问结构”或第二十八章“Web 应用部署”。

部署 Web 服务器软件

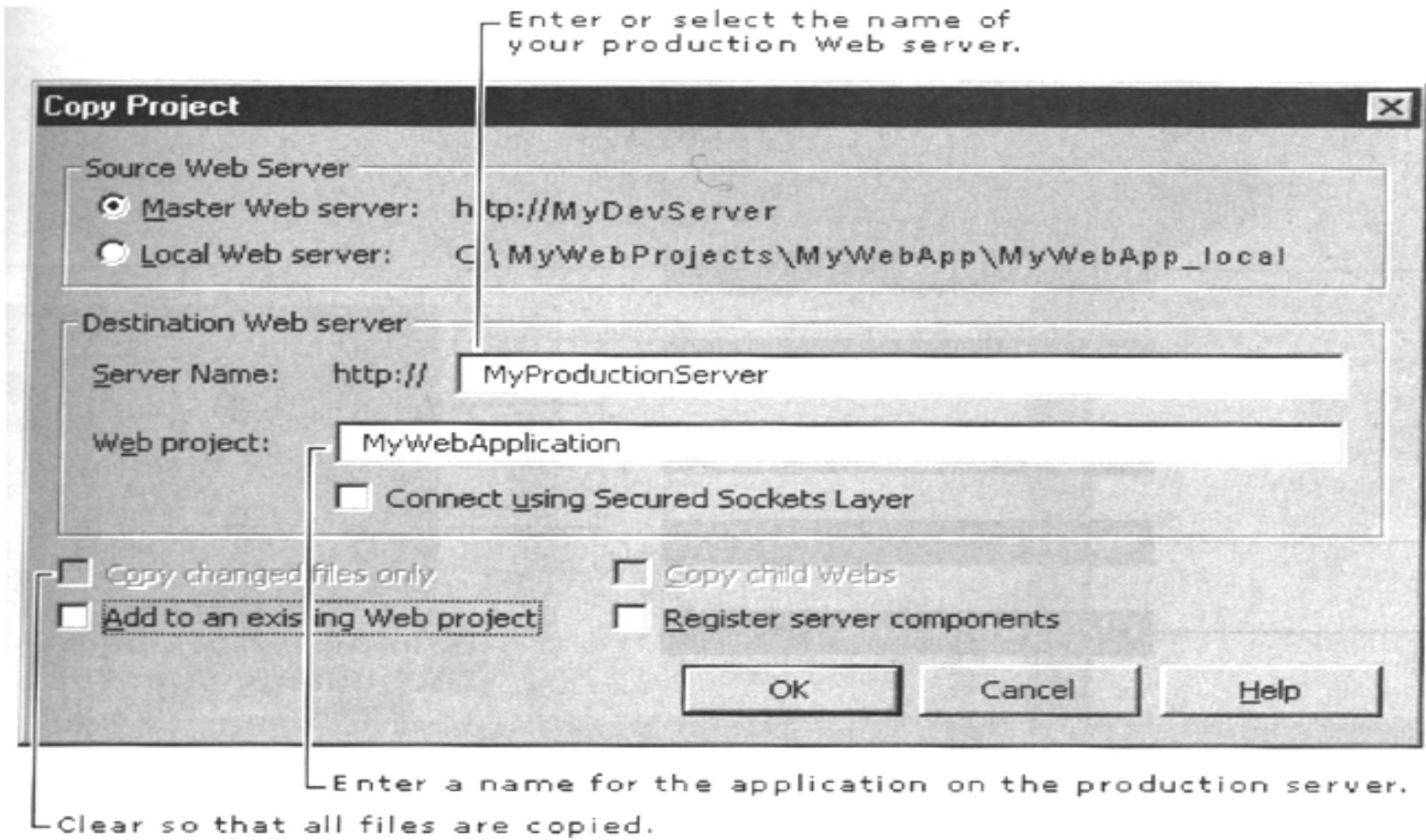
一旦为项目做好了部署准备，你就可以很快地把你的应用程序提供给你的用户（但你的产品服务器必须安装了 FrontPage 服务器扩展版本）。对没有 FrontPage 服务器扩展版本的服务器如何进行部署，参见第二十八章“Web 应用部署”。

部署 Web 应用程序

1. 在 Project Explorer 中，选择你希望部署的 Web 应用程序项目。



2. 在 Project 菜单中选用 Web Project, 然后选用 Copy Web Application, 以便显示 Copy Project 对话框。



Visual InterDev 为指定的产品服务器上建立应用程序的根，并复制你的 Web 应用程序。现在你的 Web 应用程序已准备好在新的位置运行。

检验产品服务器内容

你可以通过在浏览器中查看应用程序的办法进行检验。你还可以让其他人运行该应用程序来确认其运行正确性。为了得到最好的结果，要在没有 Web 项目文件副本的计算机上测试应用程序。

检验你部署的应用

- 在 Web 浏览器的地址框中，为产品服务器上的应用键入全 URL。



你的应用已经准备好供你的用户浏览

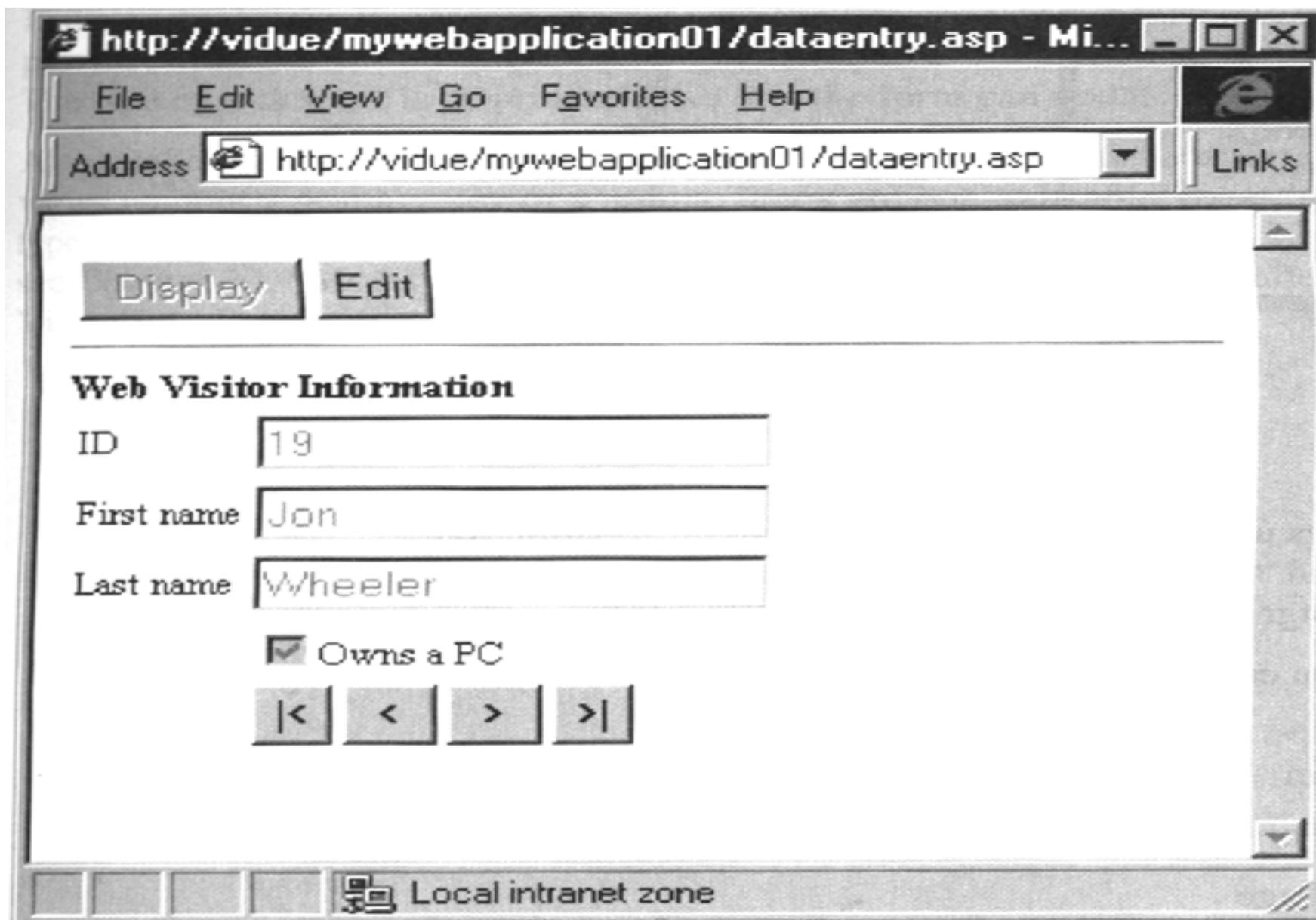
5.5 简化数据录入页面

你是否愿意维护一个单一的页面，上面有所有各方面的数据入口，而不是维护若干页面，而每个页面都有类似的标签组、文本框和按钮？你是否愿意把精力集中到功能上，而把脚本编程的细节和多个对象模型留给 Visual InterDev 处理？

使用新的 FormManager 设计期间控件生成运行期间脚本，你可以建立多用途页面，可以使用多种方式以便进行显示、编辑和把记录加到你的数据库中。

取代为你自己的 HTML 格式编写脚本，你可为页面上的控件与事件设置 FormManager 句柄特性。使用这些方式建立多用途页面，可简化你的 Web 应用设计。

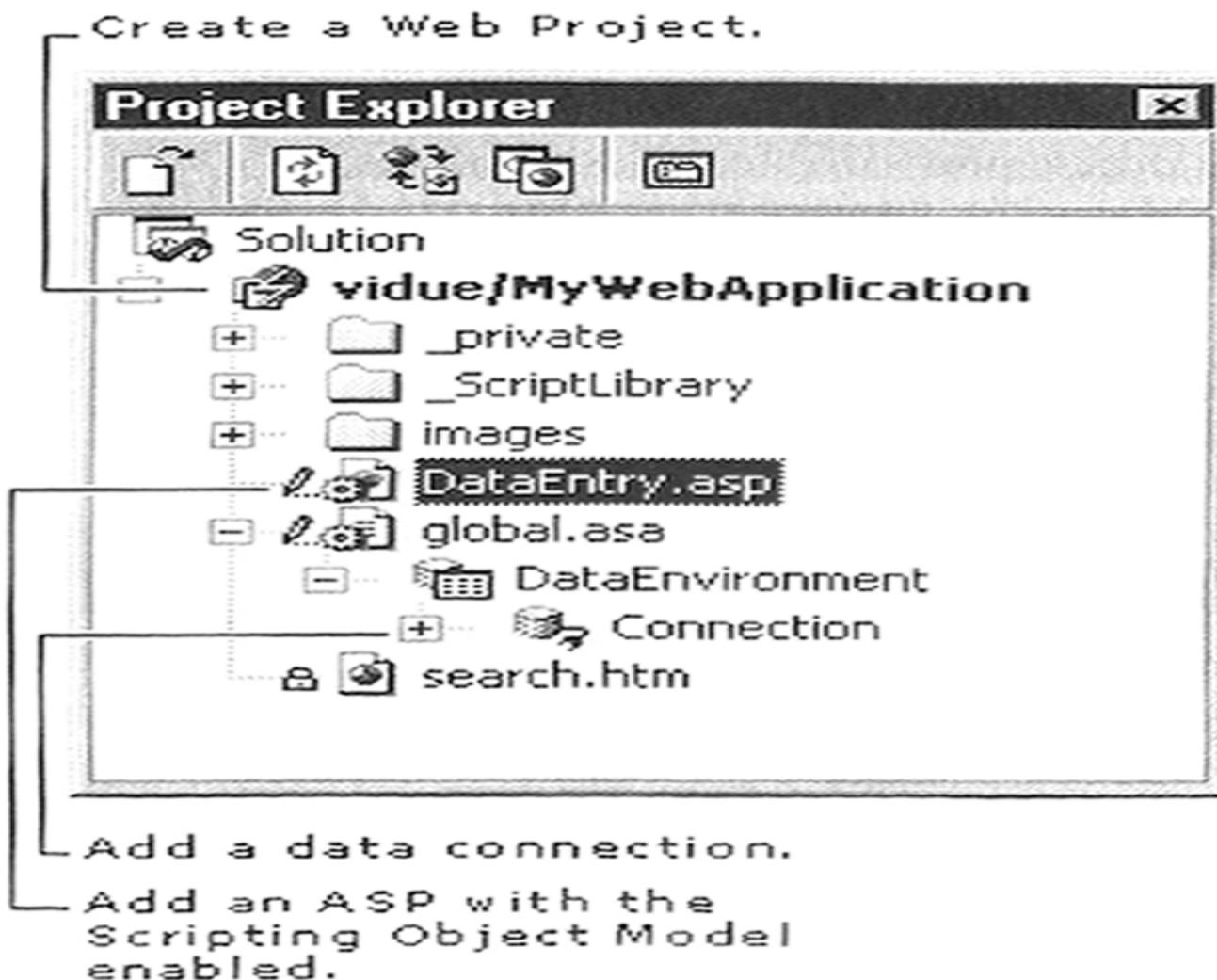
FormManager 控件使得数据录入页面的创建变得快速而简单，从而维护也变得简单，只需打开文件并改变 FormManager 特性。关于这种控件的详细内容，参见第三章“数据库基础”中的“建立事件驱动窗体”。



Web 页面上的数据绑定窗体

为完成数据录入任务，这种页面要有两种模式：

- **显示模式** 提供一种记录的只读视图。这也是打开页面时使用的一种模式。
- **编辑模式** 允许用户进入并保存改变记录、增加记录和删除记录。



为使例子简单，这个页面显示每个记录的很少几段。在真正的应用中，按照你的需要放置尽量多的段，从而实现你的构想。数据绑定概念同样不管段的数目。

在创建数据录入页面之前，要保证已经具备下列项。

你还可以使用这些控件创建 HTML 页面。要了解详细内容，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“改变对象平台”和“用设计期间控件创建窗体”两节。

你准备好之后就可以把数据录入控件和方式加到你的 ASP 页面上，步骤如下：

- 创建数据录入窗体
- 用格式方式简化脚本编程
- 完成数据录入页面

创建数据录入窗体

你可以使用数据绑定控件在 Web 页面上建立一种窗体。在过去，你可能要使用 HTML<FORM>和<INPUT>标记，但是使用 Recordset（记录集）和数据绑定控件使得数据录入格式更容易实现，这是因为在 ASP 和 Internet Explorer 4.0 两种情况中你得到同样的脚本编程对象模型。

把 Recordset 控件加到你的页面上

- 从工具框把 Recordset（记录集）控件拖到你的页面上。详细内容，参见第三章“数据库基础”中的“显示记录”一节。

注意：为对象模型编写脚本必须使用设计期间控件。

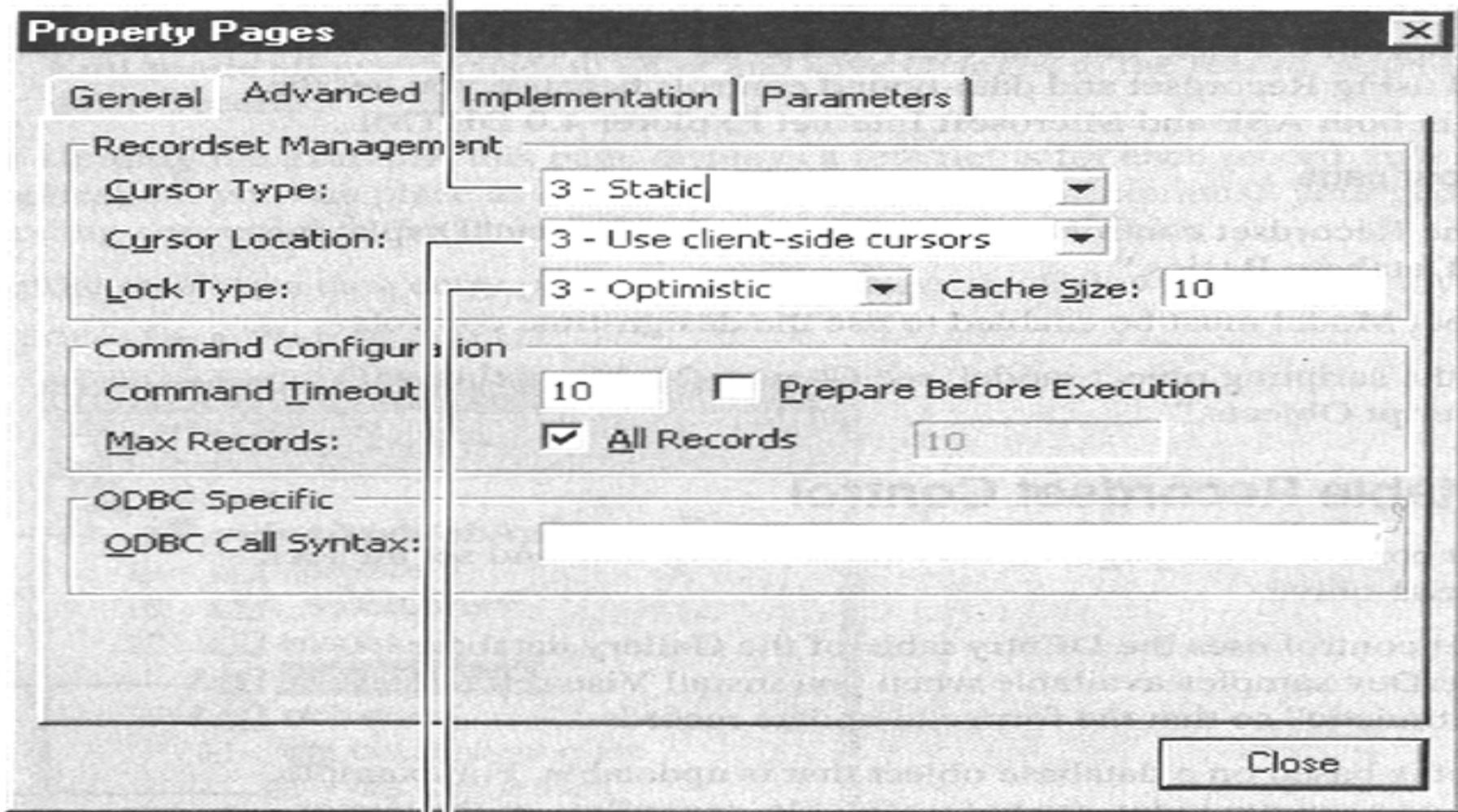
关于为对象模型编写脚本的详细内容，参见第二十四章“用设计期间控件和脚本对象编写脚本”。

规定可更新的记录集控件

把 Recordset（记录集）控件拖到页面上之后，就选择数据源，把只读以外的任何事情都设置为锁定类型。

在例子中，Recordset 控件使用 Gallery 数据库的 DEntry 表，而这种数据库同 Visual InterDev 实例一起提供，安装了 Visual InterDev 就可随时使用。所使用的锁定类型是“优化”的，因而这种格式可以更新记录。

还有，要确保记录集是以可更新的数据库对象为基础。例如，没有主关键字的视图或没有专门索引的视图都是不可更新的，不管选择的光标类型如何。关于更新记录和指定记录集方面的详细内容，参见第二十章“修改数据”。关于 Gallery 的更多内容，参见 Visual InterDev 在线文档。



Set the cursor type.

Choose an updateable lock type (any but "read-only").

Choose where the cursor is processed.

设置可更新记录集控件

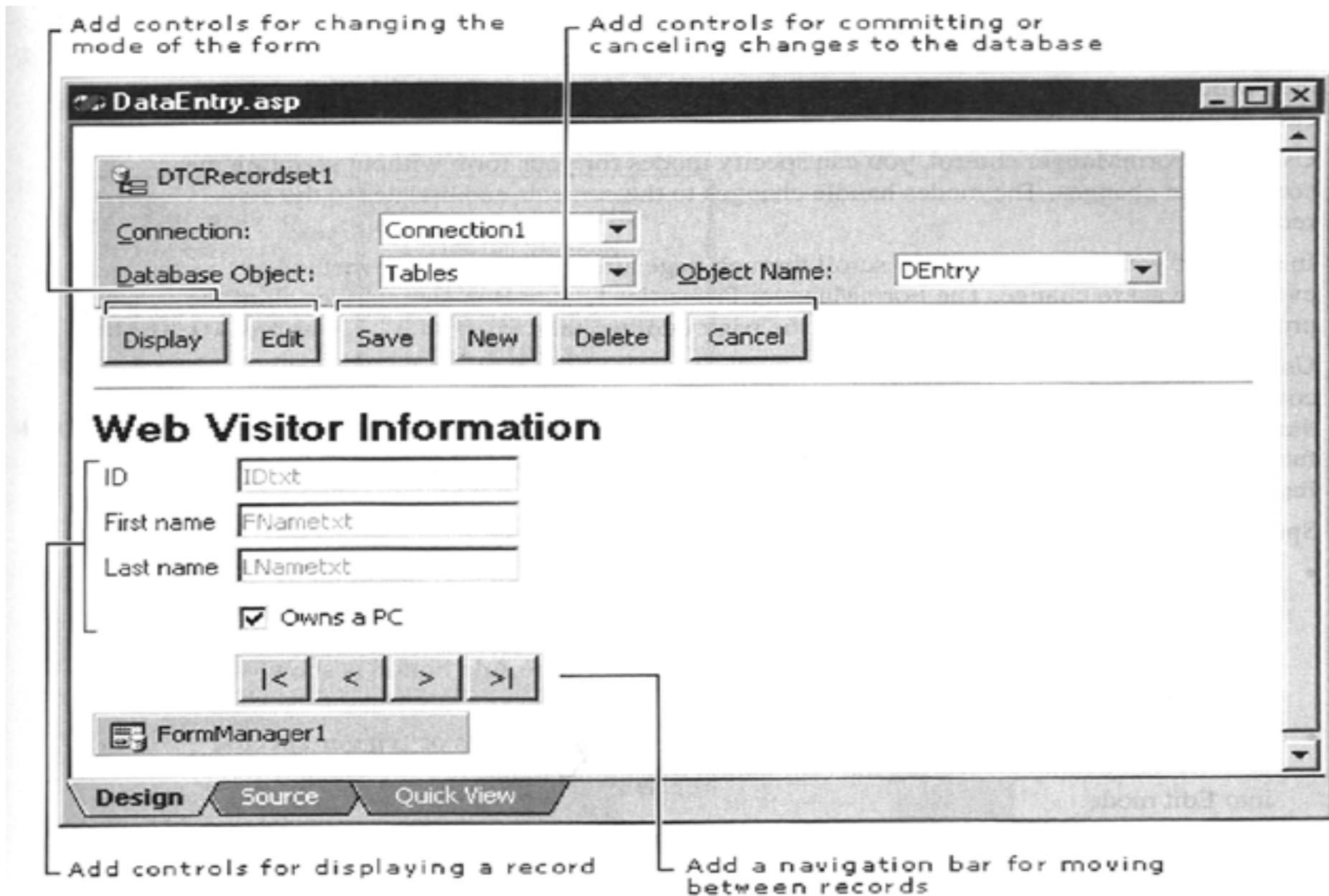
为记录显示和模式转变添加控件

把记录集控件加到你的页面之后，你就可以创建一种窗体，用于显示你希望向用户显示的字段和改变记录。

使用工具箱中的设计期间控件，你可以迅速规定哪些记录集段出现在页面上。这种段是控件的一种特性，使得添加数据的显示变得容易，而且只要记录集中的段不改变，维护也十分简单。关于使用数据绑定控件来显示记录的详细内容，参见第三章“数据库基础”中的“显示记录”一节。

为了完成页面的可视设计，要添加文本并赋予格式，还要为窗体(form)提供标题。此外，还要创建按钮行，为此要在页面上添加一个表格，把按钮控件拖到表格的各个表元中。

为显示记录，要把标签、文本框和 RecordSetNavBar 控件拖到页面上。在例子中，通过使用特性页面把文本框控件和 Checkbox 控件都绑定到记录集上。上述特性页面规定哪个段的控件要显示出来。对于这个页面上各个控件的所有初始特性设置列表，参见本章稍后的“完成数据录入页面”部分。



由数据绑定控件确定的格式易于创建和维护

用窗体模式简化脚本编程

与用同一控件组的各种变型创建并维护多个页面不同，FormManager 把一切都放到一个页面上，这个页面可以处理多个任务，并且容易维护。

使用 FormManager 控件，你可以为你的窗体指定方式，而不需要为控件编写脚本和作变更。这些模式处理在控件记录和更新记录两方面的各种改变。

此外，你不再需要一页接着一页的滚动去寻找你希望改变的方法或事件。FormManager 特性对话框把所有特性变化和页面事件处理都汇集到两个选项卡中。

使用 FormManager，你可以为你的窗体规定模式，而不用为控件和变更编写脚本。当然，如果有面向用户的功能，你还可以在 FormManager 控件中引用这些功能。例如，你可能已经有了希望在上述窗体中使用的有效功能。你可以使用 FormManager 调用这种功能，而且只要更新记录集就可使数据生效。

规定模式涉及到三个主要概念：

- **标识每种模式，在模式激活时为控件规定特性设置和方法。**在数据录入格式中，这些模式规定什么时候保存、增加和隐蔽删除按钮。在编辑模式中，保存和删除按钮被显示出来，而增加按钮被隐蔽起来。在插入模式中，增加按钮将显示出来，而保存按钮将隐蔽起来。
- **规定转变事件，这些事件引起模式变化或触发特殊动作。**例如，单击 Edit（编辑）按钮，将引起窗体从显示模式转变成编辑模式。
- **添加动作，这种动作在转变事件被触发之后和转变完成之前出现。**例如，单击 Save（保存）按钮不仅从编辑模式转换到显示模式，而且按用户在上述窗体中完成的改变更新记录集。

为把显示模式和编辑模式添加到数据录入格式中，要完成下列步骤。

规定模式

1. 打开或创建一个 ASP 页面，这个页面上有显示记录的控件，也有转换模式的按钮。

提示：添加模式之前要确保你对窗体拥有所需要的全部控件。例如，你可能需要添加按钮，例如变成编辑方式的 Edit 按钮和在该方式中触发转变事件的 Save（保存）按钮。

如果你决定添加更多控件，过一会儿你可以这样做，但是添加控件要在添加模式之前，以便使得这些控件为 FormManager 控件做好准备。这种情况也适应于你希望在页面上为之编写脚本的功能和方法。

2. 从工具框添加 FormManager 控件，在编辑器中用右键单击 FormManager，从而显示其特性页面。

3. 标识你希望的模式，并为每种模式标识控件设置。

显示模式

显示控件是只读的；
编辑按钮已经开启；
保存和删除按钮是隐蔽的。

编辑模式

显示控件是可写的；
显示按钮已经开启；
保存和删除按钮可见而且有效能。

对两种模式及其控件设置的项目

4. 在 Form Mode 选项卡中，为你已经标识的每种模式添加新模式。

Display mode

Display controls
are read-only

Edit button
is enabled

Save and Cancel
buttons are hidden

Edit mode

Display controls
are write-enabled

Display button
is enabled

Save and Cancel
buttons are visible
and enabled

5. 对每种模式规定特性设置，也为在 Form Mode 选项卡中的显示控件和窗体模式规定调用方法。

注意：对于你希望为控件设置的每个特性，都要在 Property/Value 格中添加一行。例如，如果你希望 btnNew 按钮可见但是对模式是非开启的，就在一行中把非开启特性设置为真，在另一行中规定 Show() 方法。如果希望调用你希望创建的功能，可以把 Object (对象) 栏保持空着，在 Member 栏中键入功能名字，并在 Value 栏中随便键入参数。一定要保证参数放到圆括号中。

提示：为保证窗体合适，对每种模式都要设置特性和方法。Web 环境是缺乏稳定性的，因而对控件的设置可能意外地遗留给下一个模式。例如，对那种在指定模式期间应当是不可见的控件规定 Hide() 方法，对应当是可见的控件规定 Show() 方法。如果在每种模式中对每种可见元素没

规定 Hide()方法或 Show()方法，则按钮或文本框可能是可见的，也可能是隐蔽的，这取决于在前一个模式中对这些控件规定的设置。

对于这种窗体模式设置完整列表，参见本章后边的“完成数据录入页面”一节。

Choose a mode to specify settings for.

Property Pages

Form Mode | **Action**

Name: FormManager1

States:

New Mode: [] >

Form Mode: Display, Edit

Default Mode: Display

Delete Mode

Actions Performed For Mode:

Object	Member	Value
Displaybtn	disabled	true
Editbtn	disabled	false
Savebtn	disabled	true
Deletebtn	hide	()

Close Help

Select an object. Specify a property or method. Enter a value or parameters.

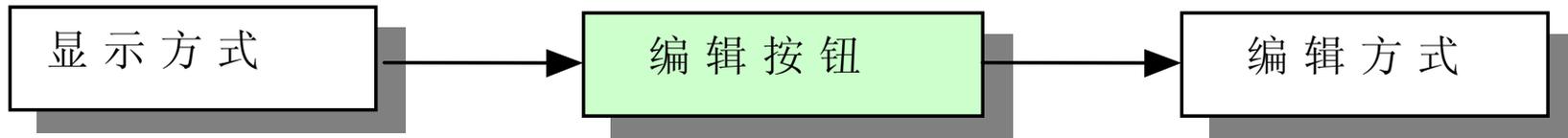
对模式规定特性设置和方法调用

注意:要保证关闭的特性是固有的设置。如果希望使控件开启,其关闭特性就需要是“假”。这些值同情况密切相关。设置稍不留心就会搞反。

此外，要使一个控件是可见的或隐蔽的，就要使用 Show()和 Hide()方法。在服务器把 HTML 发送给 Web 浏览器的时候，将把它们转变成相应的标记和属性。

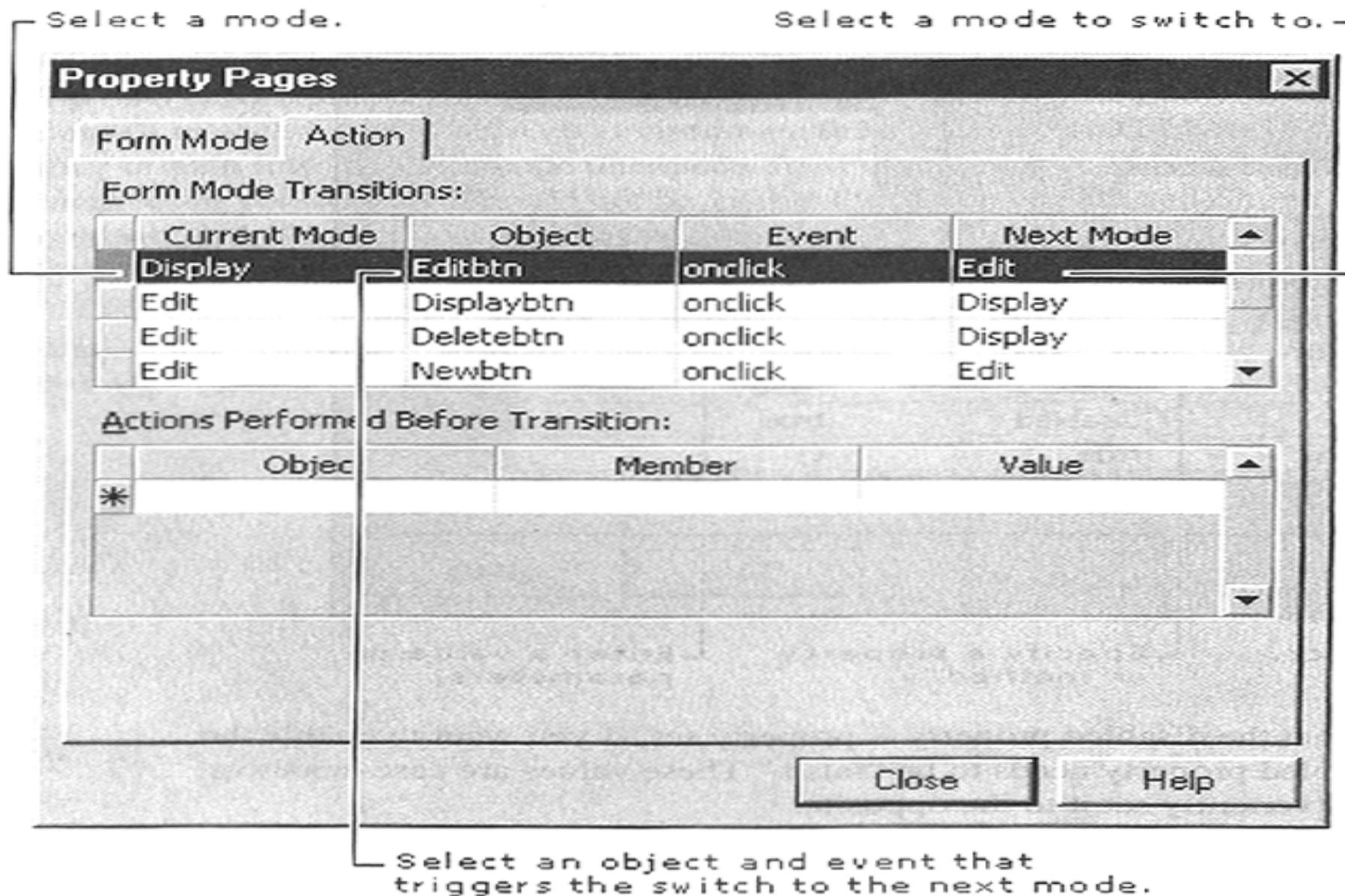
6. 在 Action 选项卡中对窗体上的每个模式按钮设置转变事件。要了解对这种窗体动作设置的完整列表，参见本章后边的“完成数据录入页面”。

例如，在单击 Edit 按钮时，该窗体就从显示模式转换成编辑模式。不用其它脚本编程，该模式就按照在 Form Mode 选项卡中的规定决定改变控件的特性。



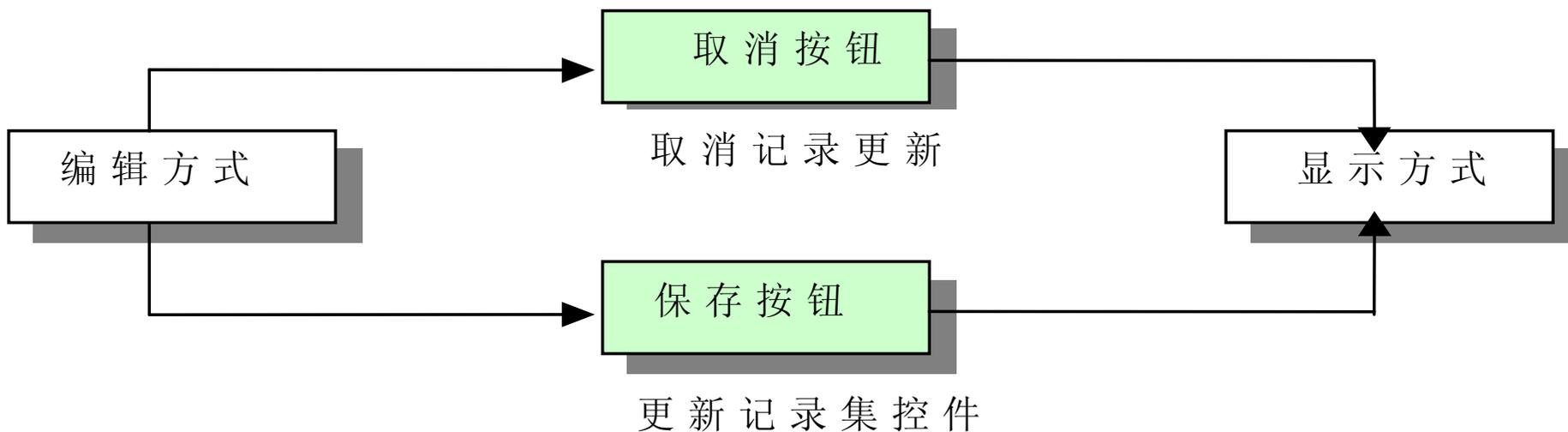
Edit (编辑) 按钮触发转换到下个模式

在显示模式中，编辑按钮只规定一种新模式是激活的，不要求其它动作。



某些转变事件要求一些动作，以便完成由模式企图完成的整个任务。
7. 如果需要，就标识并规定一些在该事件和下个模式之间要完成的动作。

例如在编辑模式中，在单击 Cancel（取消）按钮时，在转换到显示模式之前应当忽略出现的改变。在单击 Save（保存）按钮时，在转换到显示模式之前应当更新记录集。



在两种模式之间进行转变和动作的流程图

在 Action 选项卡中，按下图所示规定这些动作。

Select an object.

Enter values or parameters.

Property Pages

Form Mode | Action

Form Mode Transitions:

Current Mode	Object	Event	Next Mode
Display	Editbtn	onclick	Edit
Edit	Displaybtn	onclick	Display
Edit	Deletebtn	onclick	Display
Edit	Newbtn	onclick	Edit

Actions Performed Before Transition:

Object	Member	Value
DTCRecordset1	deleteRecord	()
DTCRecordset1	moveFirst	()
*		

Close Help

Specify a property that changes or a method applied before the transition.

动作在事件发生之后，在模式转换即将发生之前出现

8. 选择 Close (关闭)。

注意:如果你把控件/用户定义的功能添加到页面上, 要保证把它们结合到使用 FormManager 控件的模式中。

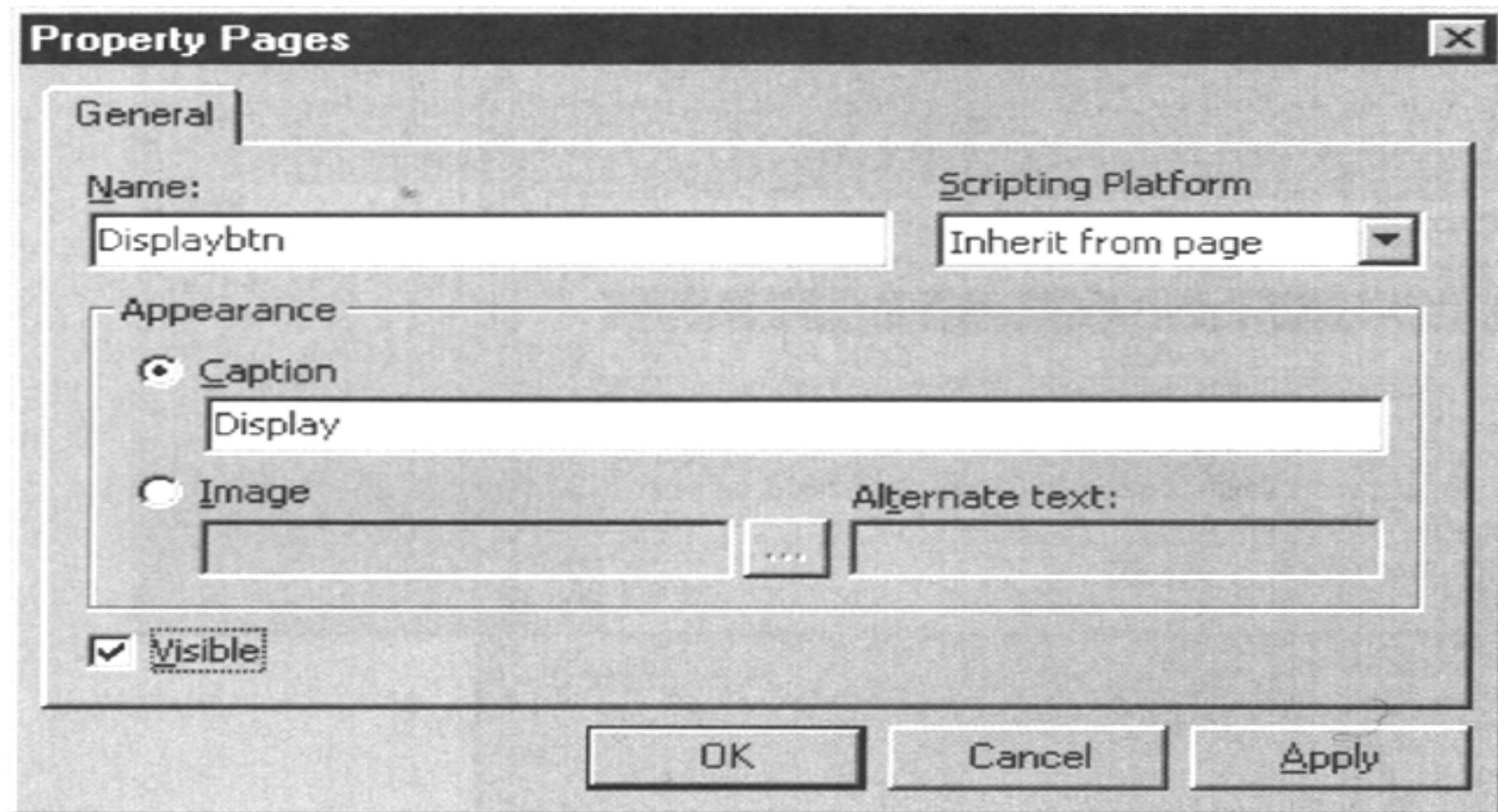
完成数据录入页面

现在你已经了解了 FormManager 控件的基本用法, 因而你可以为数据录入页面规定方法和动作。你可以使用下边的步骤进行其它设置, 完成数据录入页面。

- 对窗体上的控件进行初始特性设置
- 对控件完成模式设置
- 完成模式转变和动作。

对控件进行初始特性设置

你可以在窗体的控件特性页面上 General 选项卡中对控件进行初始特性设置。



对一个按钮进行特性设置的示例

下边表格中给出在添加模式之前在窗体上用于控件的特性设置。

记录控件

初始设置

Recordset	由 Visual InterDev 实例所使用的到 Gallery 数据库的连接 Database object=DEntry table Lock type=Optimistic 或 Batch optimistic
Label1 In1b1	可见
Text box1 Idtxt	非开启 (只读), 可见
Label2 Fname1b1	可见
Text box 2 Fname1txt	非开启 (只读), 可见
Label 3 Lname1b1	可见
Text box 3 Lname1txt	非开启 (只读), 可见
Label 4 PCchk	可见
RecordSetNavBar	可见, 设置为 recordsetdtcl

窗体模式的控件

初始设置

Display Button

Name=Displaybtn
Caption=Display
Visible=true
Disabled=true

Edit Button

Name>Editbtn
Caption=Edit
Visible=true
Disabled=false

Save Button

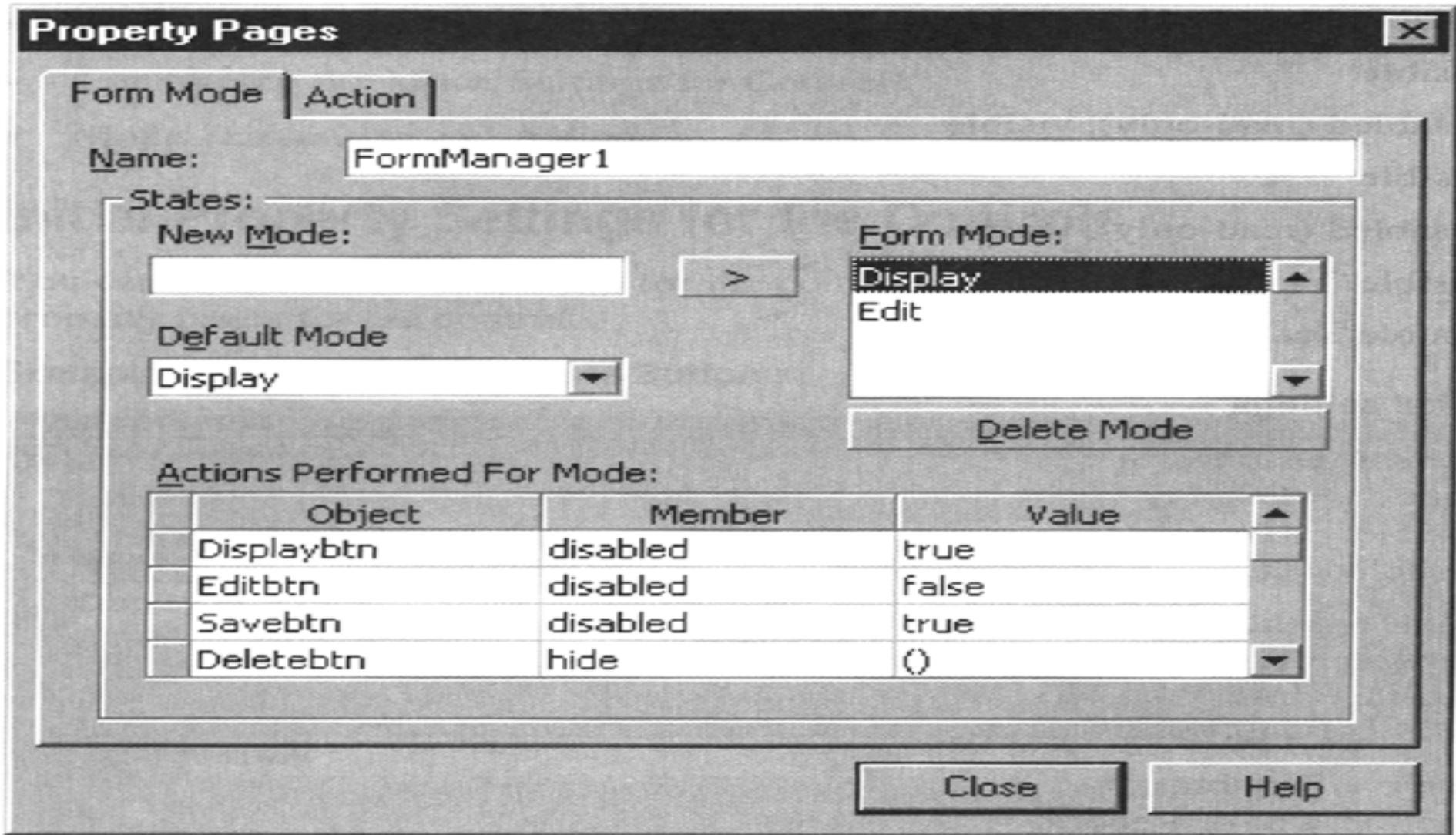
Name=Savebtn
Caption=Save Record
Visible=false
Disabled=false

续表

New Button	Name=Newbtn Caption=New Visible=true Disabled=false
Delete Button	Name=Deletebtn Caption>Delete Visible=true Disabled=false
Cancel Button	Name=Cancelbtn Caption=Cancel Visible=false Disabled=false

完成控件的模式设置

你可以通过把 Property/Value(特性/值)对添加到 Form Mode 选项卡中的办法对数据录入窗体完成模式设置。



使用 FormManager 控件的 Form Mode 选项卡来指定模式
 下表列出为完成数据录入页面所需的“特性/值”对的全部设置。

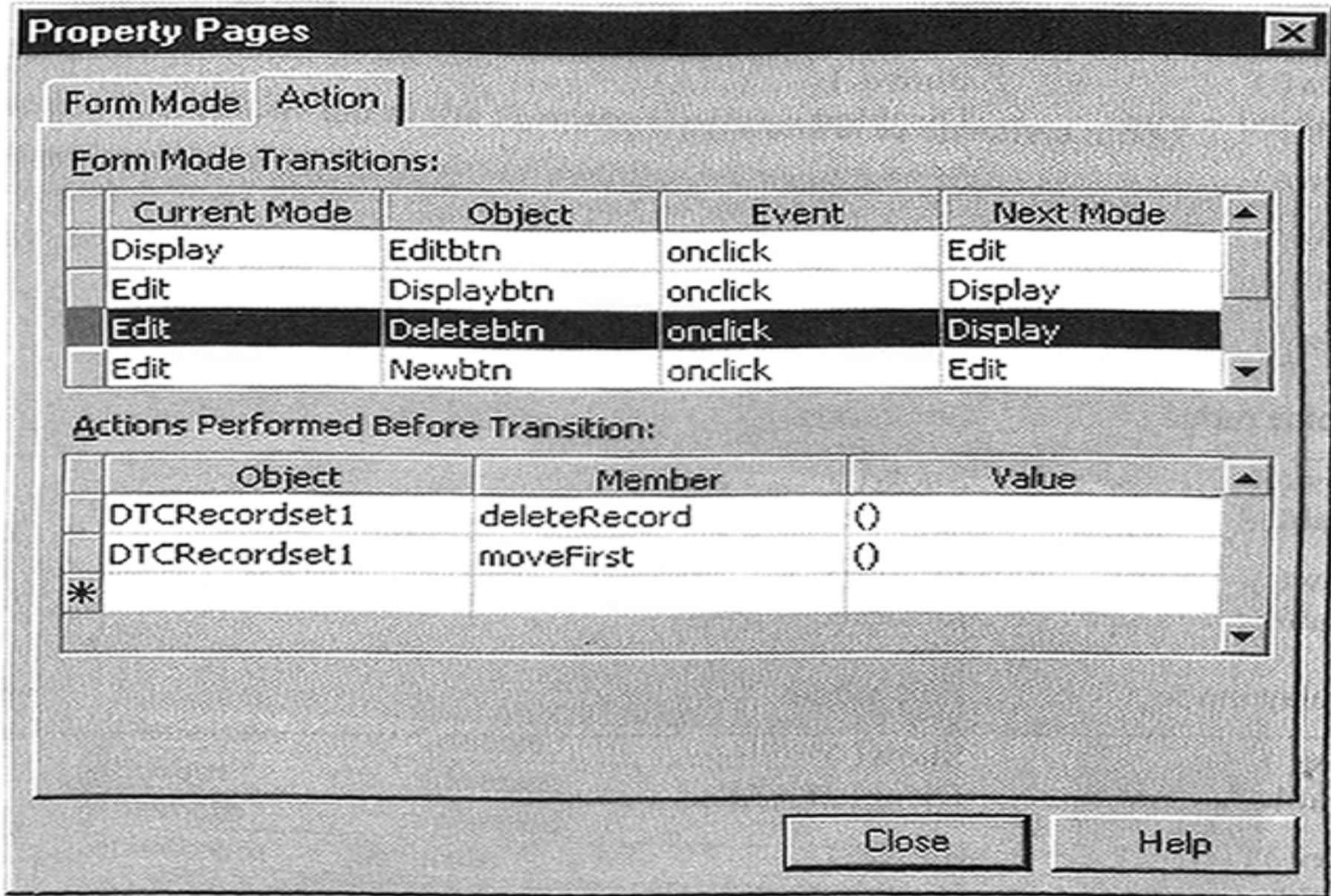
记录显示控件	显示状态	编辑状态
Text box3	Show() Disabled=true	Show() Disabled=false
Label 4	Show()	Show()
Text box 4	Show() Disabled=true	Show() Disabled=false
RecordSetNavBar	Show()	Hide()

记录显示控件	显示状态	编辑状态
Display Button	Show() Disabled=true	Show() Disabled=false
Edit Button	Show() Disabled=false	Show() Disabled=true

记录显示控件	显示状态	编辑状态
Save Button	Hide() Disabled=true	Hide() Disabled=false
Insert Button	Hide() Disabled=true	Hide() Disabled=false
Add Button	Hide() Disabled=true	Hide() Disabled=false
Cancel Button	Hide() Disabled=true	Hide() Disabled=false
Delete Button	Hide() Disabled=true	Hide() Disabled=false

模式转变和动作

你可以对数据录入窗体完成转变和动作设置，办法是标识在不同方式之间你希望用于触发转换的事件，还要标识在转换完成之前出现的动作。



使用 FormManager 控.件的 Action (动作) 选项卡规定转变和动作

下表列出在 FormManager 控件的“Action”选项卡中两个面板的设置。切记，模式本身规定了显示控件的特性和方法。下列设置规定了用于触发进入下个模式之前必须完成的模式和动作。

当前模式	对象与事件	下个模式	按钮动作
Display	btnEdit	Edit	None
Edit	btnDisplay	Display	None
Edit	btnSave	Display	Recordset.Update()
Edit	btnCancel	Display	Recordset.UpdateCancel() Recordset.MoveNext()
Recordset.AddRecord() Recordset.MoveLast()	btnNew	Display	
Recordset.DeleteRecord() Recordset.MoveFirst()	btnDelete	Display	

