
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（八）
BOOK
内部资料 非卖品

返回总目录

附录 A　排除故障

在使用 Visual InterDev 的过程中，你可能会遇到意想不到的问题和结果。你可以使用下列常见问题的摘要充分了解 Visual InterDev 特性，以便得到你希望的结果。讨论的主要方面如下：

- Web 项目
- 数据库
- 编辑与脚本编程
- 调试

A.1　Web 项目

这一节讨论的问题包括文件、图、站点导航、构造和“Internet 信息服务器”（IIS）。

使用解决方案和项目

典型情况是，在你第一次创建一个项目的时候，你便以同样的名字创建了

一个解决方案。在你创建解决方案之后，你便可以添加其它项目。如果你不希望这个最初的项目和解决方案有同样的名字，你可以在解决方案创建之后再为它重新命名。

为解决方案重新命名

1.在 Project Explorer 中选择解决方案。

2.在 File 菜单中选择 Rename。

3.在 Project Explorer 中，键入解决方案的新名字。

4.解决方案文件（.sln）有了新名字，Web 项目文件（.vip）继续保留原来的名字。

在创建或复制 Web 应用时出现的验证错误

在你为 Web 应用指定服务器时，如果键入的是 IP 地址号而不是计算机名字，你可能会遇到验证错误。如果你的计算机只对外部地址使用代理服务器，这种错误就会出现，因为系统企图通过代理服务器访问这个 IP 地址，而这个地址实际上可能是本地地址或内部地址。

你可以使用下述办法之一避免这种错误：

- 在你创建一个项目或复制一个 web 应用时使用计算机名字代替 IP 地址号。

- 如果你的 Web 项目不使用系统 Internet 设置，就改变对 Web 项目的代理选项：在主机列表中加上代理 IP 地址号，在 Web Project 选项设置中不要代理。
- 改变系统 Internet 设置：在主机列表中加上这个 IP 地址号，在 Internet 属性中不对该操作系统使用代理。

关于规定主机列表的更多内容，参见第九章“增加安全措施”中的“连接代理服务器”。

向项目添加个人文件

如果你的项目中有个人文件，你可以更新主 Web 应用，使其包含这些文件。如果你企图在 Project Explorer 中选择多个文件来更新主应用，就只选择个人文件。如果你包含主服务器上已经有的其它文件，Add to Master 命令就不能使用。

如果有人使用 Windows Explorer 或不通过 Visual InterDev 项目的某种方式去掉主文件，这个文件就作为个人文件出现在你的项目中。在你刷新项目的时候，便提示你从本地 Web 应用中删除这个文件。如果希望恢复这个文件，你就要选择“No”。再使用 Add to Master 命令把这个文件放回到 Web 应用中。

更详细的内容，参见第七章“本地工作”中的“更新主 Web 应用”。

站点图问题

- 在本地方式中，如果你把一个新文件加到一个站点图中，然后保存这个站点图，Site Designer 就自动把这个新页面加到主 Web 服务器上的主 Web 应用中。
- 在本地方式中，如果你把一个个人文件加到一个站点图中，然后保存这个站点图，Site Designer 就把这个个人文件加到主 Web 服务器上的主 Web 应用中。
- Netscape Navigator 和 Netscape Communicator 4.0 版本到 4.3 版本实现级联样式图表（CSS）不同于 Internet Explorer。Navigator 和 Communicator 把相对 URL 解释成与文档相关，不是解释成到 .css 文件的链接。为解决这个问题要：
  - 复制：把你使用的主样板目录中所有图像都复制到包含引用 .css 文件的文档目录中，但是不移动。

    或者
  - 把你 CSS 文件中的相对 URL 改变成相应的绝对 URL。

导航条、布局和主样板问题

- PageNavbar 设计期间控件（DTC）允许你使用客户 HTML 为你的 Web

页面生产导航条链接。如果你对导航条链接使用客户 HTML，就必须把 Orientation（方向）特性设置成 *horizontal*（横向），为该页面上的导航条规定一个客户方向。

- 如果你没把适当的服务器部件加到本地机器和主 Web 服务器两方面，则在导航条链接的位置上出现文本[FrontPage VINavbar Component]。为使用 PageNavbar 控件，你必须把 VINavbar 部件安装到本地机器上和主 Web 服务器上。VINavbar 部件同 FrontPage 98 服务器扩展版本一起安装。

验证 VINavbar 部件是否安装

- 在本地机器中搜索 vinavbar 文件夹。这个 vinavbar 文件夹按照默认设置安装到下述位置：*驱动器*:\Program Files\Microsoft FrontPage\version3.0\bots\。

如果你在本地机器上没找到 vinavbar 文件夹，就必须安装 FrontPage 98 服务器扩展版本。有关安装说明，参见 Visual InterDev Readme 文件中的 Readmevi.htm。

在安装 FrontPage 98 服务器扩展版本之后，你必须重新推算 Web 项目上的链接。

为你的 Web 项目重新推算链接

1.在 Project Explorer 中右键单击项目的根。

2.从快捷菜单选择 Recalculate Links。

- Layout（布局）设计期间控件（DTC）头和尾文本指示你应当按照布局把页面的内容加到应加的地方。在实际布局样板文件中，Layout.htm、Layout DTC 头和尾文本给出相对位置。
- 在你预览你的页面的时候，在站点图或 PageNavbar 控件中作出的改变可能不是立即出现在导航条上。为解决这个问题：
  - 在 Project 菜单中选择 Web Project，然后选择 Recalculate Links。

  Recalculate Links 命令用最新的导航结构信息更新导航条。

  注意:如果你在 General Tab（项目属性对话框）上不选 Always update navigation structure information on local Web server（总是更新本地 Web 服务器上的导航结构信息）选项的话，在站点图或 PageNavbar 控件中作出的改变还可能不立即出现导航条上。
- 如果你定制一个主样板文件，如一个.css 文件，并且从 FrontPage 采用一个主样板，那么你作出的定制可能被弃掉。为防止这种现象：

  1.在 _Themes 目录中包含定制主样板文件的主样板文件夹要重新命名。

  例如，如果你使用 Nature 主样板，可能把 Nature 文件夹重新命名成

MyTheme。

2.按照定制的主样板重新命名 .inf 文件。例如，把 Nature.inf 改名为 MyTheme.inf。

3.在重新命名的 .inf 文件中，改变 title=文本成新主样板文件夹名字。例如，把 title=Nature 改名为 title=MyTheme。

4.把这个主样板重新加到以前使用定制主样板的所有页面上。

- 如果你在 URL 中为站点图中的一个 HTML 页面指定一个参数或书签，PageNavbar 控件将不为 .htm 文件产生导航条链接。PageNavbar 控件对 .htm 文件不支持 URL 中的参数和书签。为防止这种问题：
  - 如果你已经指定了一个参数，就在 Project Explorer 中创建一个新 .asp 文件。把 .htm 文件从站点图中去掉，使用 Add Existing 命令为 .asp 文件指定参数，添加新 .asp 文件。
    详见第十二章“设计 Web 站点”中的“把页面加到站点图上”。
  - 如果你已经指定了一个书签，创建一个 .asp 文件。该文件有一个参数把浏览器重定向到 .htm 文件中的这个书签中。把这个 .htm 文件从站点图中去掉，加上新 .asp 文件。

链接图问题

- 在本地方式中，“链接视图”显示你的 Web 项目链接信息，就像你用你

的变更更新了主 Web 服务器一样。因而这个链接图显示的链接状态可能同 Web 应用实际状态不同。

注意:在本地方式中，如果你在 General Tab（项目属性对话框）上不选 Always update navigation structure information on local Web server（总是更新本地 Web 服务器上的导航结构信息）选项的话，“链接视图”可能不立即显示你文件中链接信息的变更。

- 如果你企图查看在目录中以下划线符号（_）开头的项所对应的链接，例如 _myfiles 目录中项的链接，有些项对应的链接可能断开，有些项可能不显示正确的输入链接和输出链接，或者有些项的链接不出现在图中。为避免这些现象，不要用下划线符号_作为名字的第一个符号创建目录。

  告诫:按照默认设置，Visual IntrDev 使用下划线符号作为名字的第一个符号创建了四个目录: _Layout、_private、_ScriptLibrary 和_Themes。不要改变这些目录的名字。

- 当你使用 View Links（查看链接）命令创建一个链接图时，你创建的是链接的设计期间视图。动态生成的链接，例如由 ASP 脚本生成的链接或从数据库生成的链接，不出现。要查看动态生成的链接，使用 View Links on WWW 命令,并键入文件的全 URL。这个 View Links on WWW

命令提供一个 Web 应用中的运行期间链接视图。

要了解更多信息，参见第十一章“站点设计”中的“链接检验”一节。

断开链接报告问题

- 要得到最好的结果，你在运行“断开链接报告”（Broken Links Report）之前，应当先对你的 Web 项目重新推算链接。重新推算链接可以保证你在“断开链接报告”中有最新链接信息。有关更多内容，参见第十六章“维护链接”中的“维修链接”。
- 如果你用带下划线_符号开头的目录，例如 _myfiles，包含在这个目录中的文件可能不在“断开链接报告”中，也可能作为不使用的文件列在“断开链接报告”中。为解决这个问题，要对这个目录重新命名，在名字中开头第一个字符不使用下划线符号_。

不能远程连接到 Windows 95 或 Windows 98 Web 服务器

微软 Windows 95 和 Windows 98 支持用 FrontPage 或 Windows NT Option Pack 安装 Web 服务器。这些服务器用于本地开发。如果你希望用一台机器作为远程 Web 服务器，就安装 Windows NT 服务器和 Windows NT Option Pack，它包含在“Internet 信息服务器(IIS)”中。

在 MTS 安装时 IIS 管理操作失败

要试着执行 Web 管理操作，如设置应用根，在下列情况下，若没有用户登录到 Web 服务器上，则会出现失败。

- 如果你正在使用“Microsoft 事务处理服务器”（MTS）。

  并且

- 如果在安装 MTS 期间，你接受默认选项 Interactive User（交互式用户）作为运行 MTS 会话的选项。

没人登录到服务器上时设置“允许 Web 管理操作”

1.向服务器打开 Microsoft“主控制台”（Master Console）。

注意:打开“主控制台”: 从“起始”菜单选择 Windows NT 4.0 Option Pack 选项、Microsoft Internet Information Server，再选择 Internet Server Manager。

2.展开 Microsoft Transaction Server。

3.展开 Computers。

4.展开 My Computer。

5.展开 Packages Installed。

6.右键单击 System，在快捷菜单上选择 Properties。

7.在 Identity 选项卡上，选择一个在服务器上有管理特权的用户。

8.关闭“主控制台”。

要了解更多内容，参见 Internet Information Server（Internet 信息服务器）和 Microsoft Transaction Server（事务处理服务器）有关文档。

Build Order 与 Visual InterDev 项目

如果你把你的 Web 项目放到 Build Order（建立顺序）对话框中，这个项目就自动加到顺序列表的末尾，因而最后建立。Visual InterDev 项目必须最后建立。如果你手动把 Visual InterDev 项目在列表中往前提，建立就可能失败。

Visual SourceSafe Explorer 和 Visual InterDev

在 Visual SourceSafe 和 Visual InterDev 一起使用的时候，Visual SourceSafe Explorer 只用来查看以前版本的历史，或滚回到以前的版本中。如果你用 Visual SourceSafe Explorer 设置检查文件检入检出（check in and out）的选项，你可能得到不可预料的结果。

例如，你可以把 Visual SourceSafe Explorer 中去掉文件本地副本的选项设置为检入。如果你选中这个选项，Visual SourceSafe 便认为主副本是本地副本，会把这个文件从 Web 服务器中去掉。

如果你希望检入这些文件之后去掉本地副本，就使用 Visual InterDev 中这个

可用的选项。

去掉本地副本

1.在 Tools 菜单中选择 Options。

2.在 Options 对话框中选择 Project，然后选择 Web Projects。

3.在 Project Options 区域，选择 Remove local copies when checking in files（在检入文件时去掉本地副本）

4.选择 OK。

A.2 数 据 库

这一节讨论使用 ODBC、数据绑定和记录集控件连接数据库方面的有关问题。

添加数据连接

你可以使用 Add Data Connection（添加数据连接）命令把一条数据添加到一个项目上。在 Project Explorer 中，在你选择一个项目或数据环境时就可以使用这个命令。

Add Data Connection 命令允许你创建数据源名字（DSN）并使用这个数据

源定义数据连接，如第三章“数据库基础”中的“连接数据库”所述。

你还可以使用 Control Panal（控制面板）中可用的 ODBC Data Source Administrator（ODBC 数据源管理器）创建 DSN。这些 DSN 可在 Select Data Source（选择数据源）对话框中使用，因而你可以使用这些 DSN 在 Visual InterDev 中建立数据连接。

然而，你必须使用 Add Data Connection 命令做这件事；直到你使用 Add Data Connection 命令从数据源创建连接以前使用 ODBC Data Source Administrator 创建的 DNS 都不变成数据连接。

有关用 Visual InterDev 使用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”和第十八章“数据库概念”。

数据绑定

如果你希望 Web 页面显示数据库中的数据，一定要把 Recordset（记录集）设计期间控件放到页面上，并设置属性以显示你已经建立起数据连接的数据库中的一组记录。更多内容，参见第十八章“数据库概念”中的“数据绑定”一节。

使用引号标识符

这是 Create New Data Source ODBC（创建新数据源 ODBC）对话框中的“Use

quoted identifiers”（使用引号标识符）复选框，这个框的选项默认是 on。如果你把这个选项变成 off，并且有包含带引号的数据库对象名字，则会出现问题。你对数据库对象使用的标识符可能包含引号，因而要保证这个选项为 on。

然而，如果你选择了这个选项，在直接处理数据库对象时(例如存储过程或触发代码)，你可能得到不可预料的结果。如果你选择这个选项，要当心你在脚本中如何使用字符串。尽可能使用单引号(‘’)可能是个好主意。

Tooltips(工具提示)显示 Recordset、PageObject 和 FormManager 设计期间控件中的错误

如果在 Recordset、PageObject 和 FormManager 设计期间控件中有一个错误，而你正在设计它们，你可以用鼠标器光标显示一个描述这个错误的 Tooltip（工具提示）。

Tooltip（工具提示）在下列情况中显示出来：当你把鼠标器光标移动到 Design 视图上设计期间控件时上，放到属性页面中的字段上或者放到 Recordset（记录集）控件面上显示的属性字段上。

例如，如果你在 FormManager 控件属性页面上 Actions Performed Before Transaction 表中 Action 选项卡上创建一个行，但是在这个行中的各个段都没填入值，于是你便得到控件错误。如果你把鼠标器光标移到 Design 视图中 FrontManager 控件上，这个错误(“Errors were found in the ‘Action’property page of

this DTC”）将显示在 Tooltip（工具提示）中。

记录集控件上出现感叹号

当你看见在 Recordset（记录集）设计期间控件（DTC）上出现一个包含感叹号的红色圆圈时，表示这里有数据问题。选择记录集 DTC，以便使它有一个焦点，然后把鼠标器光标移到红色圆圈上，于是 Visual InterDev 就显示一个 Tooltip（工具提示）并解释问题。

一些典型原因是：

- 在你改变连接或数据库对象设置时，在 Object Name 列表框中出现红色圆圈。如果你改变连接或数据库对象，要选择一个新对象名字。
- 在数据连接存在问题时，在 DTC 左上部 Recordset 名字的左边出现红色圆圈。要调查的方面包括：
  - 对数据库服务器的连接是否建立。
  - 在 Project Explorer 中 Global.asa 文件下 DataEnvironment 内的数据连接。
  - Recordset Design-Time Control Property Pages（记录集设计期间控件属性页面）对话框的 General 选项卡上的连接。

DTC 属性页面上的 Recordset 名字变成红色

通过把DTC的Recordset属性发送给同一个页面上已经存在的RecordsetDTC之一的办法可把数据绑定在一个数据绑定设计期间控件上。如果你把一个数据绑定 DTC Recordset属性设置为一个已经存在的 Recordset DTC，然后从该页面上删掉同样的 Recordset DTC，这个被删掉 Recordset的名字就在数据绑定 DTC的 General属性页面上变成红色。

要解决这个问题，你必须把一个 Recordset DTC 加到被删除 Recordset DTC的页面上，或者把数据绑定 DTC Recordset 属性设置改变成一个已经存在的RecordsetDTC。

在数据库记录之间快速移动时的错误

如果你正在包含数据绑定窗体的 ASP 页面上工作，企图从一个记录到另一个记录快速导航，你可能看见下述错误：

Microsoft OLE DB Provider for ODBC Drivers error '80040e21'

这个错误是由于前一个数据库请求完成之前企图获得下一个记录造成的结果。你可以通过对包含这个窗体的页面进行缓冲的办法避免这种错误，这样在数据完全取出来并构成页面之前便不显示出来。要设置缓冲，把下边的命令加到该页面的顶部：

```
<% Response.Buffering=true %>
```

A.3 编辑与脚本编程

这一节讨论使用 HTML 编辑器、使用设计期间控件和其它控件所遇到的问题。

开启脚本对象模型

只有在脚本对象模型对所在页面开启的情况下，设计期间控件的功能才是正常的。如果你把设计期间控件拖到尚未使脚本对象模型开启的页面上，消息框便提示你加上这个模型。然而，如果你选择“No”并接着进行，这个控件便不能同其它控件通信，因而在运行期间便不能正确显示。有关细节，参见第二十三章“脚本编程概念”中的“脚本对象模型”。

用设计期间控件和脚本对象编写脚本

在编写脚本过程中同由“设计期间控件”创建的脚本对象进行交互作用时，要记住下列提示：

- 在编写脚本过程中，在任何时刻按 CTRL+J 总能显示完善语句(statement

completion)下拉列表。如果你在一个具体对象描述体中，这个对象的属性、方法和事件都在这个列表中。如果你不在对象描述体中（例如你在一个空行上），下拉列表包含完成的对象列表和它们的编号。

- 对属性、方法和事件的名字要区分大小写，即便你在 VBScript 中编写脚本也要这样。在许多情况中，你使用的名字都是用脚本对象模型框架（在 JavaScript 中多半如此）求值的，因而区分大小写。

  提示:当你用 JavaScript 进行工作时，完善语句却不能辨认对象，除非你使用了正确的大写字母。

- 当你在拥有开启脚本对象模型的 ASP 页面上工作时，你可以用对象的 thisPage 访问当前页面。如果在这个页面上有 PageObject 设计期间控件，这个对象的 thisPage 便出现在 Script Outline 窗口中，并可用于“完善语句”。如果在这个页面上没有 PageObject 控件，你还可以在你的脚本中使用这个 thisPage 对象，但是因为在 Script Outline 窗口中是不可见的，因而“完善语句”也不能使用。

  在客户脚本中，对象 thisPage 只当你在页面上有一个页面对象时才是可用的。但是在客户脚本中，你使用这个 thisPage 对象只用于访问页面对象的属性和方法。对客户脚本中当前页面的属性、方法和事件，你可以使用 DHTML 文档对象。

- 当使用方法和函数名字时，你必须正确使用小括号。如果你使用“完善语句”编写方法调用，就不能对方法调用加小括号。

  如果你在 JavaScript 中编写脚本，在下列情况中必须使用小括号：如果方法作为一个函数执行、如果方法调用返回一个值或者如果方法返回一个对象引用子。下边两个例子说明小括号的使用：

```
str = Textbox1.value()      // returns a value
//   In the following getPagingNavbar() returns an object.
//   and hide() executes a function
Gridl.getPagingNavbar().hide()
```

  如果你不使用小括号，这个方法调用就作为一个函数指针来对待，把函数的名字作为参数进行传递。例如，如果你调用一个对象通知方法，你把这第二个参数（函数名字参数）作为不带小括号的函数指针进行传递，如下例所示：

```
< S CRI P T      L J a v aS
function setAdviseMethods(){
    Btn1.advise(“onmouseover”,changecaption)”;   //no (),function ptr
}
</SCRIPT>
```

如果你用 VBScipt 进行工作，且方法调用返回一个值，则必须使用小括号，否则它们就是可选的。例如，在 VBScript 中，下边两个语句都能工作：

```
Textbox1.hide()
Textbox1.hide
```

在启用一种把一个对象引用子返回的方法时，要小心使用小括号，如下例所示：

```
Grid1.get PagingNavbar().hide
```

如果你在 VBScript 中调用一个对象通知方法，你要把第二个参数的名字（函数名字参数）作为字符串，放在小括号中进行传递：

```
<SCRIPT LANGUAGE="VBScript">
Function setAdviseMethods()
   Btn1.advise("onmouseover", "changecaption()") ' note quotes and ()
End function
</SCRIPT>
```

提示:要记住，因为脚本对象模型使用 JavaScript，你应当对方法名字区分

大小写，即便是在 VBScript 中也是一样。

使用属性窗口

Properties（属性）窗口使用一种栅格显示所选控件或对象的所有属性。下边列出使用属性窗口的一些提示。

- **检查选择** 窗口顶部的下拉列表显示当前元素的类型。如果你没看见你所期望个属性，就检查该选择。个别时候（如当你用 HTML 表元素进行工作时）这个表显示元素的层次关系，你可以用来选择正确的元素。
- **特性分类** 在窗口顶部的按钮允许你按字母顺序或按类别显示属性。把属性按类别分组易于查找属性，例如查找内联样式属性。
- **使用定制选项** 某些元素，例如设计期间控件，提供定制属性页面，易于设置属性值。如果客户属性页面是可用的，则（Custom）属性就首先出现在按字母顺序的列表中，你可用单击“下一个”框中的按钮进入显示窗口。
- **使用帮助** 对大多数单个属性都提供帮助。要看帮助，选择栅格中的属性，然后按 F1。

使用快速视图选项卡进行预览

HTML 编辑器的 Quick（快速）视图选项卡只处理页面的客户部分：HTML

文本和客户脚本。尽管你可以在快速视图中预览 .asp 文件，编辑器却不能完成到服务器的往返传递，因此：

- 没执行服务器脚本，
- 没出现其目的平台是服务器的设计期间控件。

剪切与粘贴 HTML 文本

你可以从一个 HTML 源中剪切或复制文本到 Windows 粘贴板上。HTML 源如 Microsoft Internet Explorer、Visual InterDev 编辑器的 Design 视图、Visual InterDev 帮助系统或样板应用)有两种文本版本是可用的：HTML 版本和文本版本。

HTML 版本对保留符号如<、>和引号使用 HTML 的换码序列。例如，你若把<MARQUEE>复制到粘贴板上，HTML 版本就是“&1t;MARQUEEgt;”。文本版本包含你剪切或复制的原文本的准确副本。

在你粘贴的时候，你可以选择其中一种版本。要粘贴 HTML 版本，选择 View 菜单中的 Paste，或右键单击菜单。要粘贴文本版本，从右键单击菜单中选择 Paste as Text。一般说：

- 在 Design 视图中，如果你希望看见实际文本如“<MARQUEE>”，就选择 Paste。如果你希望创建一个标记，就选择 Paste As Text。标记不出现在 Design 视图中，因而你看见的文本可能同你从其它源中剪切的文本

不同。

- 在 Source 视图中选择 Paste as Text。

在 HTML 编辑器的设计视图中剪切或复制

如果你从 Design 视图中剪切或复制一个元素到 Windows 粘贴板上，要包含用于管理这个元素的辅助信息。如果你把这个元素粘贴到 Design 视图以外的任何地方，可能同时粘贴上额外的<DESIGNTIMEP>标记。如果信息源是 Design 视图，就一定要检查粘贴结果。

区分设计期间控件与 HTML 控件

要认真区分 ToolBox 上设计期间控件和 HTML 控件。这两个控件有类似的名字，例如在两种选项卡上都有文本框和按钮。在 Design 视图中工作时，因为 HTML 控件以图形显示，两者更难以区分。

在 Source 视图中，你可以发现区分两者比较容易。在这里，HTML 控件是按照简单 HTML 标记表示的，例如<BUTTON>或<INPUT>。设计期间控件是用图形表示或用<OBJECT>标记加辅助信息表示的。

在 Source 视图中以文本视图工作

你可以在 Source 视图中按照文本查看控件（设计期间控件、Java 小程序和

其它控件），这在你希望看见页面的准确内容或快速改变一系列控件是很有用的。在你使用文本视图进行工作时，下边的提示对你是很有帮助的。

- 只在 Source 视图中查看采用文本形式的控件。Quick 视图和 Design 视图总能提交控件的图形视图（如果可能的话）。
- 你可以在 Options 对话框中使用 HTML 选项卡为控件设置默认视图。
- 你可以为单个控件选择文本视图，办法是右键单击它，然后再选择 Always View As Text。当做这件事的时候，属性 VIEWASTEXT 便加到控件的<OBJECT>标记上。
- 你还可以在 View 菜单中选择 View Controls As Text，临时在文本视图中观察控件。然后你可以使用 View Control Graphically 命令重新显示控件图形（控件的<OBJECT>标记包含 VIEWASTEXT 属性的除外）。
- 如果你在 View 菜单中选择 Refresh，编辑器在当前默认视图（按照 Options 对话框的设置）中显示所有控件，但是控件的对象标记中包含 VIEWASTEXT 属性的除外。

使用文本视图可能影响控件在你页面上的工作。如果 Visual InterDev 设计期间控件按文本显示，功能就不正常，这是因为它们不能同脚本对象模型框架进行通信。在你把一个设计期间控件加到一个页面上以前，要保证你已经把选项设置成查看图形控件。如果在设置了文本视图选项情况下，你不经意地把一个

控件加到页面上，HTML 编辑器不能创建控件实例。你只能看见控件的 HTML<OBJECT>标记，而看不见控件本身。

打印包含设计期间控件的页面

当你在 HTML 编辑器中打印一个页面时，有些设计期间控件不能以你设置的值显示出来。如果你在一个页面上放上多个设计期间控件实例，例如，如果把多个 Textbox 或 Button 控件放到一个页面上，第二个和以后的控件在打印件上将显示它们的默认值。

在脚本块中创建控件

不要在 Source 视图中向<SCRIPT>块中拖放：设计期间控件、ActiveX 控件或任何其它对象控件。<SCRIPT>块总是假设只包含脚本，因而编辑器不分析控件的语法，因此不在块中创建它们的实例。

如果你把一个控件拖放到<SCRIPT>块中，编辑器就创建一个<OBJECT>块，里面包含有关控件的信息，如控件的类别 ID，但是不创建其它信息，如参数或脚本，因而这样的控件是不能用实例说明的，也不能起正确作用。

在 Toolbox 中不起作用的 ActiveX 控件

有些实例，你可能看见在 Toolbox ActiveX Controls 选项卡上列出的一些控

件是没有用处的。例如，你可能看见 Toollbox 上的 WalletAddress 和 WalletPayment，但是当你试着把它们拖到一个页面上的时候，便会出现这样错误“不能示出图形控件”。

如果你在 Windows NT 中工作和在安装 Internet Explorer 4.0 时选择“Browser Only”就会出现这种错误。这种最小的安装不把控件复制到你的计算机中，因而它们是不可用的。然而 Windows 登记了创建这种控件的条目，因而它们出现在 Toolbox 中。

从 Toolbox 使用服务器对象

Visual InterDev Toolbox 的 Server 选项卡列出多个在“Internet 信息服务器”（IIS）中常用的对象。这些服务器对象都是 IIS 的组成部分，但是主要在 ASP 页面中使用，用作在 Visual InterDev 中建设 Web 解决方案的一部分。例如，Visual InterDev 开发者经常创建使用 ActiveX Data Objects(ADO)访问的基于服务器的数据库。

Server Objects 选项卡上的列表是事先定义的，不是用实际安装到特定服务器上的对象为基础构成的。这就允许 Toolbox 显示同一个列表，而不管你当前使用的是什么服务器。然而，这还意味着可以把一个控件从 Toolbox 加到一个 ASP 页面上，在该页面运行的时候服务器不能使用这个控件。你应当经常用产品服务器上的服务器对象测试页面。

要使用服务器对象，就把对象从 Toolbox 拖到你的 Web 页面上（不要把它们拖到<SCRIPT>块中）。当你这样做的时候，编辑器以正确的 RUNAT 和 PROGID 属性以及一个 ID 属性创建一个<OBJECT>块。有些对象创建这种方式还要求其它信息，一般情况下你可使用<OBJECT>块中<PARAMETER>标记提供信息。

当这个页面运行的时候，服务器在<OBJECT>块中创建一个实例。你可以使用这个对象的 ID 在你的脚本中引用这个对象。例如，你把一个 Browser Capabilities 对象拖到你的页面上，编辑器就创建一个对象标记，如下所示：

```
<OBJECT RUNAT=server PROGID=MSWC.BrowserType id=OBJECT1>
</OBJECT>
```

提示:改变对象的 ID 是有一定意义的。例如，在这种情况中，你可能把它改变成 oBType。

在使用<OBJECT>块为这个对象创建实例之后，就可以用它的 ID 在你的脚本中引用它。下边从一个页面摘录出一部分示范如何按照上述说明使用对象：

```
Your browser is <%=oBType.browser%>, version <%=oBType.version%>.
<p>
Your browser supports these programming languages:
<UL>
<%If oBType.javascript Then%>
   <LI>JavaScript</LI>
<% End If %>
<%If oBType.VBScript Then%>
   <LI>VBScript</LI>
<% End If %>
</UL>
```

注意:如果你工作中使用的对象在安装了 Visual InterDev 的计算机上可以使用，在你编写脚本的时候你就为对象的属性、方法和事件得到“完善语句”。例如，如果 Visual InterDev 正在与 IIS 的同一个计算机上运行，Visual InterDev 就在 Server Objects 选项卡上拥有对对象的访问权。

你可以在 Microsoft Developer Network 或其它地方找到有关使用服务器对象和有关单个对象的信息，如下表所示。

有关信息	在 MSDN 或其它地方查找下列主题
使用服务器对象	工具和技术 激活服务器页面 使用脚本语言 使用部件和对象
ADO 命令 ADO 连接 ADO 记录集 注意：服务器 ADO 对象不是 Visual InterDev 数据绑定模型的组成部分。要把 Visual InterDev 设计期间控件绑定到数据库上，就使用“记录集”设计期间控件。	数据库和报文服务 微软数据访问 SDK 微软 ActiveX 数据对象 ADO 程序员参考 用 ADO 启动 用 ADO 2.0 启动

续表

Ad Rotator	平台 SDK
浏览器 Caps	因特网/内部网/外部网业务
内容链接	激活服务器页面
My Info	为 ASP 安装部件
	参见第二十五章“用 HTML 元素编写脚本”中的“编写可移植脚本”。有关使用 AD Rotator 对象的例子，参见 Visual InterDev 在线文档中 Random Ad Sample。
目录文件系统	Visual Studio 文档
	参考
	语言参考
	VBScript 语言参考
	对象
CDONTS 新邮件	平台 SDK
CDONTS 会话	因特网/内部网/外部网业务
	激活服务器页面
	为 ASP 安装部件
	用于 NTS 部件的合作数据对象

续表

索引服务器实用程序	这些对象在 Index Server 文档中描述。
索引服务器查询	要显示这个文档，在安装 IIS 的服务器上从 Windows Start 菜单开始沿着下述途径： Program Windows NT 4.0 Option Pack Product Documentation 在文档中沿着下述途径： Microsoft Index Sever Bulding Search Forms Active Server Pages
MSMQ	平台 SDK
MSMQ 邮件	联网和分配业务 微软报文队列服务器（MSMQ） MSMQ 参考 MSMQ ActiveX 部件 MSMQ 邮件 ActiveX 部件

为页面对象方法传递参数

在你把参数传递给页面对象上的方法的时候，如果在往返于服务器中调用结果，参数就要转换成字符串。为了在页面之间用 http 协议传递参数就要求这

样做。

如果你编写一个方法，用于提取需要具体数据类型的参数，你总应当检查数据类型，并在必要时进行转换。例如，在页面对象中的下述函数提取必须作为数字进行处理的参数。因此在使用参数之前要转换这个参数。

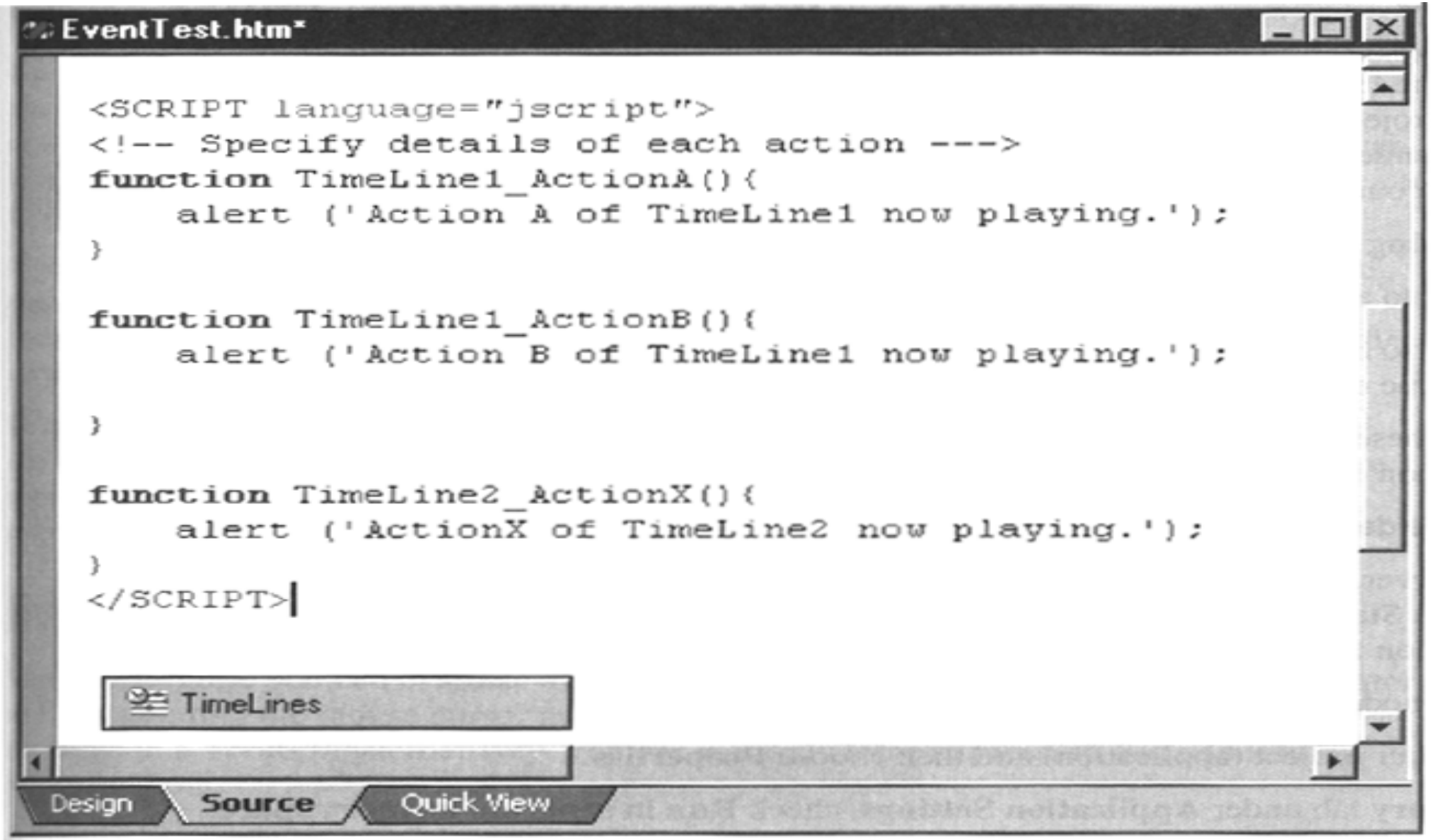
```
Sub SubmitBid( BidAmount )
    errorCode = ""
    iBidAmount = CInt( BidAmount )
    If iBidAmount < 0 then
        errorCode = INVALIDBID
    End if
    ' etc.
End Sub
```

更多细节，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”。

规定时间线事件

在把 Timeline（时间线）控件加到页面上之后，你需要添加一个脚本，这个脚本规定你在控件属性页面上命名的事件。例如，在下图中的控件有两条规定的时间线。TimeLine1 有两个事件：ActiveA 和 ActiveB。TimeLine2 只有 ActiveX。为了简化例子，事件产生提示信息。

更多内容，参见在线帮助中的 Timelines Design-Time Control（时间线设计期间控件）。


EventTest.htm*
<SCRIPT language="jscript">
<!-- Specify details of each action --->
function TimeLine1_ActionA(){
alert ('Action A of TimeLine1 now playing.');
}
function TimeLine1_ActionB(){
alert ('Action B of TimeLine1 now playing.');
}
function TimeLine2_ActionX(){
alert ('ActionX of TimeLine2 now playing.');
}
</SCRIPT>
TimeLines
Design
Source
Quick View

测试页面过渡

因为在你从一个页面到另一个页面移动时出现页面过渡，你只能在 Web 浏览器中查看过渡。编辑器的 Quick 视图不出现过渡。

要查看进入过渡的情景，使用你的 Web 浏览器，从一个页面开始，然后打开要过渡的页面。

要查看离开过渡的情景，打开过渡的页面，然后打开另一页面。

要了解更多内容，参见在线帮助中的 PageTransitions Design-Time Control。

A.4 调　　试

这一节包括有关调试客户脚本和服务器脚本的信息，还包括 SQL 调试注释。

在服务器上开启 ASP 调试

只当对 IIS 服务器上你的应用（项目）开启调试的情况下，你才可以调试 ASP 页面中的脚本。如果通过在你项目中启动页面的办法启动一个调试器会话，Visual InterDev 就可以自动开启服务器上的调试，如同第二十六章“调试你的页面”中的“对 ASP 页面开启服务器脚本调试”。

然而，在下述状态中这些选项不会允许你进行调试：

- 你希望连接到在服务器上已经运行的文档（进程）。
- 你希望启动调试器以便对运行时间检测到的错误作出响应（及时调试）。

在上述情况下，必须为服务器上的应用程序手动开启调试。

为 IIS 应用手动开启调试

1.在服务器上，使用下述路径从 Windows Start 菜单启动“管理控制台”（MMC）：Programs\Windows NT 4.0 Option Pack\Microsoft Internet Information Server\ Internet Service Manager。

2.打开你的服务器节点。

右键单击你的项目（应用），然后选择 Properties。

在 Application Settings 下的 Directory 选项卡上，选中 Run in separate memory space，它使你的应用在进程外（out-of-process）运行。选择 Apply。这可能要占一会儿时间。

3.选择 Configuration，然后在 Application Configuration 对话框中选择 App Debugging 选项卡。

4.在 Debugging Flags（调试旗标）下，选中 Enable ASP server-side debugging(开启 ASP 服务器一侧调试)，然后选择全部对话框。

关闭调试，按步骤 3 到步骤 4 反向进行。

不要永远保持这样的调试选项，因为这会影响性能。有关细节，参见本附

录后边“调试服务器脚本时的性能问题”。

调试中浏览器显示错误页面

如果你启动调试器，浏览器显示错误页面，即不是你希望调试的页面，则返回到 Visual InterDev，确认按照项目起始页面设置你希望调试的页面。

服务器页面的及时调试

如果调试器安装在服务器上，则服务器不会把服务器脚本语法或运行时间错误传递给客户。相反，它将停止页面处理，并在其计算机屏幕上显示一条消息，提示启动调试器。服务器可以启动同 Internet 信息服务器（IIS）一起安装的 Script Debugger（脚本调试器），或者，如果安装上的话，启动 Visual InterDev 调试器。

如果你查看运行服务器软件的计算机屏幕，你可以对消息作出响应，并在服务器本地进行调试。但是，如果你不能查看服务器计算机屏幕，你会愿意让服务器把错误信息传递给客户机，以便你使用远程调试以查找并解决错误。

要这样做，你必须去掉服务器上所有调试器。如果 Script Debuuger 已经安装了，也把它去掉。同样，要确认 Visual InterDev 和其它 Visual Stidio 工具没安装在服务器上。但是你必须确认 Remote Debug Manager（远程调试管理器）还安装在上面，否则你在服务器上根本不能进行调试。

在调试器上计算表达式

在调试期间，你可以动态计算任何表达式，办法是在编辑器中进行选择，然后把鼠标器光标指向那里。表达式计算出来，结果显示在一个“工具提示”（Tooltip）中。另外，你还可以用右键单击表达式，然后把它加到 Watch（观察）窗口中。

然而，你在计算表达式时必须小心，它会影响你的调试会话，甚至会影响你的 Visual InterDev 会话。例如，如果你选择表达式 Session.Abandon，调试器便计算这个表达式，而你的当前会话会被放弃。

在非正常结束之后重新设置调试器选项

如果 Visual InterDev 设置为自动开启服务器调试，在打开项目之后你第一次调试 ASP 页面的时候，读取“Internet 信息服务器”（IIS）上当前调试器的设置。如果要求，Visual InterDev 便开启调试，包括把 IIS 应用移到进程之外。在你完成调试会话的时候，Visual InterDev 又恢复服务器的以前设置。

但是，如果 Visual InterDev 在调试期间退出，例如，计算机突然掉电，原来的设置便丢掉了。当你下一次再启动 Visual InterDev 的时候，就再一次读取服务器调试设置，但是它们反映 Visual InterDev 异常退出时遗留下来的设置。

如果出现这种情况，并且你希望设置同遗留下来的设置有所不同，则必须

手动在 IIS 中重新设置。具体说，如果你希望关闭 ASP 调试，并且把应用复位成在进程内运行，你必须亲自改变这些选项。有关细节，参见本附录前边的“在服务器上开启 ASP 调试”。

调试服务器脚本时的性能问题

在“Internet 信息服务器”（IIS）上开启调试可能在以下几方面影响你项目的性能。

Visual InterDev 提供的选项，可以在你的 IIS 应用（你的 Visual InterDev 项目）上自动开启服务器调试。在打开项目之后，你第一次启动调试器，Visual InterDev 便对你的项目核实 IIS 调试器选项；如果选项没设置，就设置选项。你可以通过下述办法明显减少设置时间：使用 IIS“微软管理控制台”（MMC）应用为你的应用手动开启调试，如本附录前边“在服务器上开启 ASP 调试”所述。

在服务器上开启调试，可能会影响服务器的性能。因此建议你只在需要的时候才开启调试，而且在正用着的产品服务器上你不要开启调试。有关调试服务器脚本的细节，参见第二十六章“调试页面”中的“调试服务器脚本”。

在你进行调试的时候不要使用 Internet Explorer 的 Active Desktop 模式。这样做会有两种影响。第一，调试器将监视在 Desktop（桌面）上运行的所有应用，这些应用可能影响性能。第二，在调试期间可能出现问题，原来只要求引导 Internet Explorer,现在还要求引导 Windows。

不能安装 SQL 调试部件

如果 BackOffice 安装向导报告说不能安装 SQL 调试，则核实你为 SQL 服务器是否安装了 Service Pack 3 或更高版本。

排除 SQL 调试故障

如果在你试图使用 SQL 调试的时候出现错误，试一试下列方法：

- 如果 SQL 服务器配置成按 System Account 进行登录，你就不能试用 SQL 调试。
- 服务器上的错误写到事件日志上。要查看日志，在服务器上 Windows Start 菜单上选择 Programs,然后选择 Administrative Tools，再选择 Event Viewer。打开 Application 日志，然后在 Source 下寻找 SQLDebugging98。有时候核实 System 日志是有用的。
- 如果你不能保证在你的服务器上是否设置正确，就在你的 SQL 服务器安装的 Bin 文件夹中查找文件 MSSD198.DLL。

如果你不能在 Windows 95 工作站上进行 SQL 调试工作，就核实客户 DCOM 设置是否正确。

在 Windows 95 客户计算机上核实 DCOM 配置

- 使用 Regedit.exe，在关键词 HKEY_LOCAL_MACHINE\SOFTARE\

Microsoft\OLE 之下，检查 Windows Registry 中的设置：

- EnableDCOM 应当设置为 Y,
- EnableRemoteConnect 应当设置为 Y。

如果设置不正确，就改变设置，然后重新启动计算机。

在调试器中处理 SQL 变量

如果你正在调试存储的过程或触发器软件，并且如果你敲击一个带有“@”前缀的变量，调试器就把它加到 Watch 窗口中并去掉@前缀。在这个窗口中编辑这个变量的名字时再把@前缀加上。

A.5 从 Visual InterDev 1.0 升级到 Visual InterDev 6.0

你可以很容易地从 Visual InterDev 1.0 升级到 Visual InterDev 6.0。

你在 Visual InterDev 1.0 中创建的 Web 应用可以在 Visual InterDev 6.0 中运行。实际上，如果你的 Web 开发组成员还在继续使用 Visual InterDev 1.0，他们仍然可以查看应用文件，即便你已经升级为新的 Visual InterDev 版本也是一样。

跨越版本的兼容性是 Microsoft Visual Studio 部件的关键特性。然而，Visual InterDev 6.0 提供更强的 HTML 编辑器、一种新的设计期间控件工具框和其它可继续使用的综合调试能力。

对已有的 Web 应用从 Visual InterDev 1.0 升级时的有关重要问题，参见下列各节：

- 服务器从 Visual InterDev 1.0 升级。
- 数据连接在 Visual InterDev 6.0 和 Visual InterDev 1.0 之间的差别。
- Visual InterDev 6.0 和 Visual InterDev 1.0 之间设计期间控件的差别。

服务器从 Visual InterDev 1.0 升级

Visual InterDev 6.0 包含增强的服务器软件能力。使你服务器升级所需要的所有软件都包含在 Visual InterDev 6.0 安装压缩盘中。因为升级服务器软件是为结合 Visual InterDev 1.0 进行工作而设计的，因而在安装新的服务器扩展版本之后仍然可以打开 Visual InterDev 1.0 的项目。

注意:这个服务器扩展版本升级要求运行 Visual InterDev 6.0。

增强的服务器版本

为了发挥 Visual InterDev 6.0 的许多新特点的长处，你需要运行 Visual InterDev 6.0 Server Setup，升级扩展你的服务器。6.0 版本中许多新特性和改善的特性都与最新服务器扩展软件有关。例如 Visual InterDev 6.0 在 Web 应用中的脚本、数据环境、设计期间控件和设计命令都使用升级的服务器能力以简化 Web 应用的开发。这些特性只有在运行 Visual InterDev 6.0 Server Setup 程序之

后得到支持。

注意:只有服务器要求这种新服务器扩展版本。你不需要在客户机上安装服务器软件。

Internet 信息服务器 4.0

除了 Visual InterDev 6.0 Server Setup 之外，要运行 Visual InterDev 调试器，你还需要安装 Internet 信息服务器 4.0。Internet 信息服务器 4.0 同你的 Visual InterDev 软件和微软 Windows NT 5.0 包装在一起。

你可以使用 Visual InterDev 调试器测试在 Visual Basic Script Edition（VBScript）和 JScript 中编写的脚本，以及测试在 Sun Microsystem Java 中编写并使用 Java Virtual Machine(VM)运行的应用。有关调试器的更多内容，参见第二十六章“调试页面”。

如果你安装了另外一种支持微软调试协议的脚本语言，如 REXX 或 Perl，你还可以使用那种语言调试脚本。

数据连接在 Visual InterDev 6.0 和 Visual InterDev 1.0 之间差别

数据连接为你的 Visual InterDev 项目提供访问具体数据库的能力。一旦你连接到一个数据库上，你就可以在 Visual InterDev 中在你的 Web 页面上显示或编

辑数据。

在 Visual InterDev 中，要同数据库连接，你首先要为数据库创建一个数据源名字（DSN）或选用一个已经有的名字。然后你使用这个 DSN 创建一个数据连接，再把它加到你的项目中。但是，使用这种新版本的 Visual InterDev 项目中存储的数据连接方式发生了一些基本变化。

在 Visual InterDev 1.0 中，建立的数据连接是由你项目中的会话变量标识的。你可以打开项目的 Global.asa 文件看见这个会话变量。

Visual InterDev 6.0 中，数据连接是由你项目中的应用变量标识的。你可以在你项目的 Global.asa 文件中看见应用变量。你还可以使用这种新“数据环境”为每个数据连接编辑应用变量。有关使用新数据环境的更多内容，参见第十八章“数据库概念”中的“数据环境”。

当你第一次在 Visual InterDev 6.0 中打开 Visual InterDev 1.0 项目的时候，数据连接自动从会话变量转换成应用变量。在 Visual InterDev 1.0 中定义的会话变量仍然保留在 Global.asa 文件中，但是这些变量只用于在 Visual InterDev 1.0 中装配的 DTC 部件。

由应用变量定义数据连接比较好，比较快，需要的资源也比 1.0 版本方法要少。当使用 6.0 版本应用变量的时候，每个数据连接可以在访问你应用的所有用户之间共享。这是对 1.0 版本会话变量的改善，在 1.0 版本中每个用户都必须

重新创建数据连接变量。

但是，在不同版本的 Visual InterDev 项目中工作还有两个问题需要了解：

- 用 Visual InterDev 6.0 中的数据连接打开 Visual InterDev 1.0 项目。
- 用 Visual InterDev 1.0 中的数据连接打开 Visual InterDev 6.0 项目。

用 Visual InterDev 6.0 中的数据连接打开 Visual InterDev 1.0 项目

当你在 Visual InterDev 6.0 中打开 Visual InterDev 1.0 项目的时候，项目的所有数据连接自动从会话变量转换成应用变量。

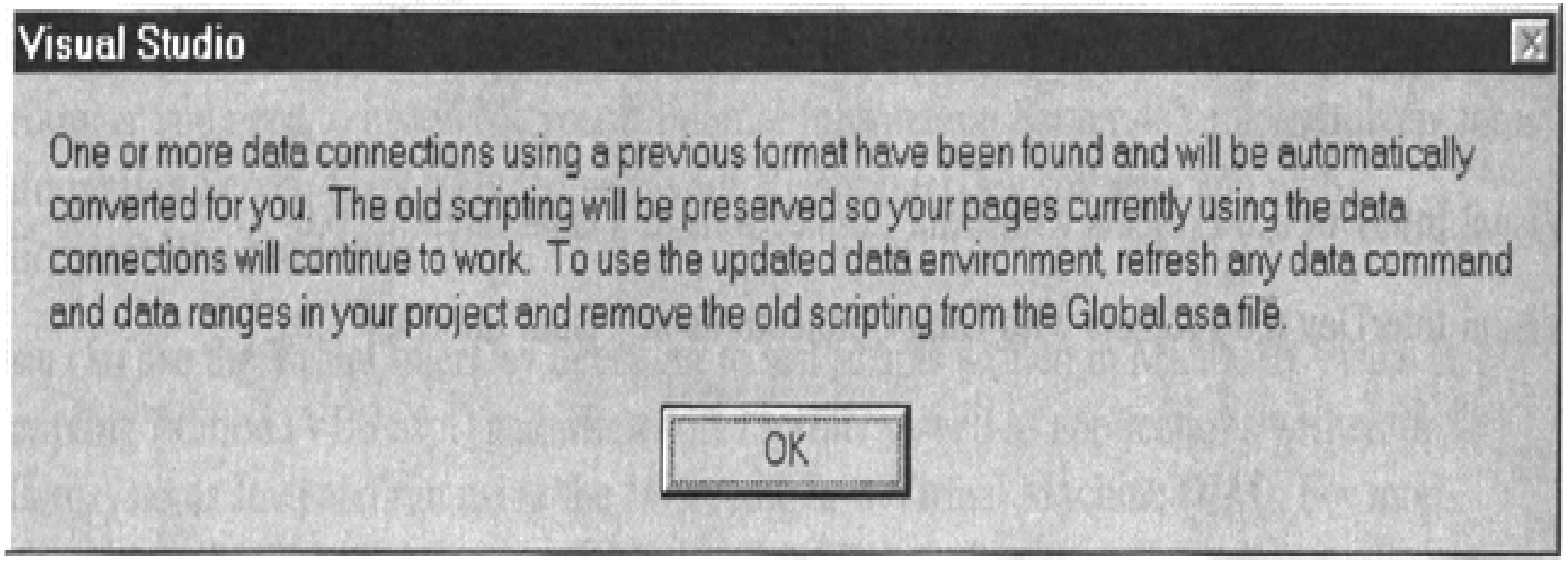
在 Visual InterDev 1.0 创建的会话变量仍然在你的 Global.asa 文件中保持可见，但是不被 Visual InterDev 6.0 所使用。此外，定义你数据连接的新应用变量被加到你的项目 Global.asa 文件中。

因此，数据变量信息实际上在 Global.asa 文件中出现两次，因而两个版本都可以继续正常工作。

在 Visual InterDev 1.0 中，数据连接是用 Global.asa 文件的会话变量定义的，如下边脚本所示。

```
Sub Session_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'-Project Data Connection
Session ("NorthWind_ConectionString")="DBQ=E:\NorthWind.mdb;
   DefaultDir=E:\;Driver={Microsoft Access Driver (*.mdb)};
   DriverId=25;FIL=MSAccess;ImplicitCommitSync=Yes;MaxBufferSize=512;
MaxScanRows=8;PageTimeout=5;SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Session("NorthWind_ConnectionTimeout") = 15
Session("NorthWind_CommandTimeout") = 30
Session("NorthWind_RuntimeUserName") = "admin"
Session("NorthWind_RuntimePassword") = ""
==Visual InterDev Generated - DataConnection endspan==
End Sub
```

当你在 Visual InterDev 6.0 中打开你的 Visual InterDev 1.0 项目的时候，下边的消息通知你：数据连接正向 Global.asa 文件中的应用变量转换。



当你在 Visual InterDev 6.0 中查看项目的 Global.asa 文件的时候，你现在的数据连接是由应用变量定义的，也有用原来会话变量定义的，如下边脚本所示：

```
Sub Application_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'--Project Data Connection
Application ("NorthWind_ConectionString")="DBQ=E:\NorthWind.mdb;
   ➥ DefaultDir=E:\;Driver={Microsoft Access Driver (*.mdb)};DriverId=25;
   ➥ FIL=MSAccess;ImplicitCommitSync=Yes;MaxBufferSize=512;MaxScanRows=8;
   ➥ PageTimeout=5;SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Application ("NorthWind_ConnectionTimeout") = 15
Application ("NorthWind_CommandTimeout") = 30
Application ("NorthWind_RuntimeUserName") = "admin"
Application ("NorthWind_RuntimePassword") = ""
==Visual InterDev Generated - DataConnection endspan==
End Sub

Sub Session_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'--Project Data Connection
Session ("NorthWind_ConectionString")="DBQ=E:\NorthWind.mdb;
   ➥ DefaultDir=E:\;Driver={Microsoft Access Driver (*.mdb)};DriverId=25;
   ➥ FIL=MSAccess;ImplicitCommitSync=Yes;MaxBufferSize=512;MaxScanRows=8;
   ➥ PageTimeout=5;SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Session("NorthWind_ConnectionTimeout") = 15
Session("NorthWind_CommandTimeout") = 30
Session("NorthWind_RuntimeUserName") = "admin"
Session("NorthWind_RuntimePassword") = ""
==Visual InterDev Generated - DataConnection endspan==
End Sub
```

如果你再一次在 Visual InterDev 1.0 中打开 Web 应用，则 1.0 版本忽略新应用变量，并使用原来的会话变量，而不管数据连接是什么时候建立的。

Visual InterDev 6.0 忽略已经存在的会话变量，并使用新连接，而不管数据连接是什么时候建立的。

用 Visual InterDev 1.0 中的数据连接打开 Visual InterDev 6.0 项目

当你在 Visual InterDev 6.0 项目中创建数据连接的时候，数据连接是用项目 Global.asa 文件中的应用变量定义的。这些应用变量只当你在 Visual InterDev 6.0 中打开项目时才放在需要定义的数据连接上。

Visual InterDev 1.0 不自动解释在 Visual InterDev 6.0 项目中定义的数据连接。

如果你希望在 Visual InterDev 6.0 中创建项目，又希望在 Visual InterDev 1.0 中打开这个项目，你就必须手工重新定义项目的数据连接，以便与 Visual InterDev 1.0 一致。要与 Visual InterDev 1.0 一致，你的数据连接必须用会话变量定义，也用应用变量定义。

重新定义你的项目数据连接，以便与 Visual InterDev 1.0 一致，很容易。有两种方式可以实现：

- 在 Visual InterDev 1.0 中添加数据连接。

- 把按应用定义的数据连接复制成按会话定义的数据连接。

在 Visual InterDev 1.0 中添加数据连接

重新定义 Visual InterDev 6.0 数据连接，以便与 Visual InterDev 1.0 一致的最简单方式是在 Visual InterDev 1.0 中打开项目，加上数据连接，就像新建立连接一样。

这个过程将自动生成 Visual InterDev 1.0 所需要的按会话定义的数据连接。这将不修改 Visual InterDev 6.0 所要求的应用定义的数据连接。

在 Visual InterDev 1.0 中把数据连接加到你的 Web 应用上

1.在 Visual InterDev 1.0 中打开你 Visual InterDev 6.0 项目。

2.在 FileView 中选择项目。

3.在 Project 菜单中单击 Add to Project，然后单击 Data Connection。

4.在 Select Data Source 对话框中，选择你在 Visual InterDev 6.0 中连接的已有的 DSN。

5. 如果需要的话，登录到数据服务器上。

现在你的数据连接已经准备好了，供你在 Visual InterDev 1.0 中使用。

Visual InterDev 1.0 中验证你的数据连接属性

1.关闭 Global.asa 文件，如果在编辑器中打开了的话。

2.在 FileView 中展开 Global.asa 文件。

3.右键单击数据连接的名字，再单击 Properties。

4.验证数据连接属性与在 Visual InterDev 6.0 你为数据连接设置的属性相同。

你的 Visual InterDev 6.0 项目现在有了数据连接，而且在 Visual InterDev 1.0 和 Visual InterDev 6.0 中都能正常工作。

把按应用定义的数据连接复制成按会话定义的数据连接

如果你希望使得你的 Visual InterDev 6.0 项目能够在 Visual InterDev 1.0 中正常工作，又不要先在 Visual InterDev 1.0 中打开项目，你可以手动在 Visual InterDev 6.0 的 Global.asa 文件编辑器中把按应用定义的数据连接复制成按会话定义的数据连接。

把按应用定义的数据连接复制成按会话定义的数据连接

1.在 Visual InterDev 6.0 编辑器中打开你项目的 Global.asa 文件。

2.找到定义数据连接的 Application_OnStart 脚本命令。

代码程序将与下例类似。

```
Sub Application_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'--Project Data Connection
Application ("NorthWind_ConectionString")="DBQ=E:\NorthWind.mdb;
DefaultDir=E:\;
    ➥ Driver={Microsoft Access Driver (*.mdb)};DriverId=25;FIL=MSAccess;
    ➥ ImplicitCommitSync=Yes;MaxBufferSize=512;MaxScanRows=8;PageTimeout=5;
    ➥ SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Application ("NorthWind_ConnectionTimeout") = 15
Application ("NorthWind_CommandTimeout") = 30
Application ("NorthWind_RuntimeUserName") = "admin"
Application ("NorthWind_RuntimePassword") = ""
==Visual InterDev Generated - DataConnection endspan==
End Sub
```

3.复制按应用定义的全部数据连接代码，再贴回到 Global.asa 文件中，这样定义数据连接的代码就出现两次。

4.手动改变第二个数据连接定义实例中的亮显行，在 Global.asa 文件中创建按会话定义的数据连接，如下例所示。

```
Sub Application_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'--Project Data Connection
Application("NorthWind_ConectionString")="DBQ=E:\NorthWind.mdb;DefaultDir=E:\;
   ➥ Driver={Microsoft Access Driver (*.mdb)};DriverId=25;FIL=MSAccess;
   ➥ ImplicitCommitSync=Yes;MaxBufferSize=512;MaxScanRows=8;PageTimeout=5;
   ➥ SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Application("NorthWind_ConnectionTimeout") = 15
Application("NorthWind_CommandTimeout") = 30
Application("NorthWind_RuntimeUserName") = "admin"
Application("NorthWind_RuntimePassword") = ""
==Visual InterDev Generated - DataConnection endspan==
End Sub
Sub Session_OnStart
'==Visual InterDev Generated - DataConnection startspan==
'--Project Data Connection
Session ("NorthWind_ConectionString")="DBQ=E:\NorthWind.mdb;DefaultDir=E:\;
   ➥ Driver={Microsoft Access Driver (*.mdb)};DriverId=25;FIL=MSAccess;
   ➥ ImplicitCommitSync=Yes;MaxBufferSize=512;MaxScanRows=8;PageTimeout=5;
   ➥ SafeTransactions=0;Threads=3;UID=admin;UserCommitSync=Yes;"
Session("NorthWind_ConnectionTimeout") = 15
Session("NorthWind_CommandTimeout") = 30
Session("NorthWind_RuntimeUserName") = "admin"
Session("NorthWind_RuntimePassword") = ""
==Visual InterDev Generated - DataConnection endspan==
End Sub
```

现在你已经成功地修改了你的 Visual InterDev 6.0 项目，数据连接与 Visual InterDev 1.0 一致。

Visual InterDev 6.0 和 Visual InterDev 1.0 之间设计期间控件的差别

在 Visual InterDev 6.0 中有多种增强产品结合起来，创建丰富、快速又较坚固的使用设计期间控件（DTC）的环境。在 Visual InterDev 6.0 中，你会发现一组新的 DTC，它包括一些在运行期间有效的设计期间用户界面。此外，DTC 背后的基本思想是进行扩充，以便允许你按照编程模型进行编码，这与脚本编程正好相反。

当 Visual InterDev 1.0 项目在 Visual InterDev 6.0 中打开的时候，你的 Visual InterDev 1.0 Legacy DTC 将继续起作用，但是使用 DTC 的方法却同 Visual InterDev 的以前版本发生了变化。要了解使用 DTC 的更多内容，参见第二十三章“脚本编程概念”中的“Web 应用中的脚本”，第二十四章“用设计期间控件和脚本对象编写脚本”，以及本附录后边的“安装设计期间控件脚本库”。

总的说来，Visual InterDev 6.0 设计期间控件为你提供丰富而直观的编辑界面，供创建数据富集的页面使用。数据绑定控件使 ASP 页面或 HTML 页面中的数据很容易结合起来以便与数据库相互作用。

这个 Visual InterDev 版本中包括的控件允许你把范围广泛的浏览器作为目

标，从而把焦点缩小到 Internet Explorer 4.0 中可用的丰富的动态 HTML 上。有关数据绑定控件更多内容，参见第二十四章“用设计期间控件和脚本对象编写脚本”。

数据环境

在 Visual InterDev 6.0 中除了增加 DTC 变型和功能之外，创建和修改数据相关的对象现在都集中到一个地方：图形数据环境。这种数据环境将影响你使用和思考 DTC 的方式。

在这种数据环境中，你可以把对象拖到 ASP 页面上，自动创建“数据绑定设计期间控件”。数据环境为创建可重复使用的数据相关对象，并把这些对象放到 Web 页面上提供标准界面。

在 Visual InterDev 6.0 中创建新项目之前，为了充分发挥 Visual InterDev 6.0 DTC 的长处，你应当熟悉数据环境，探索数据环境将如何影响你的项目计划。

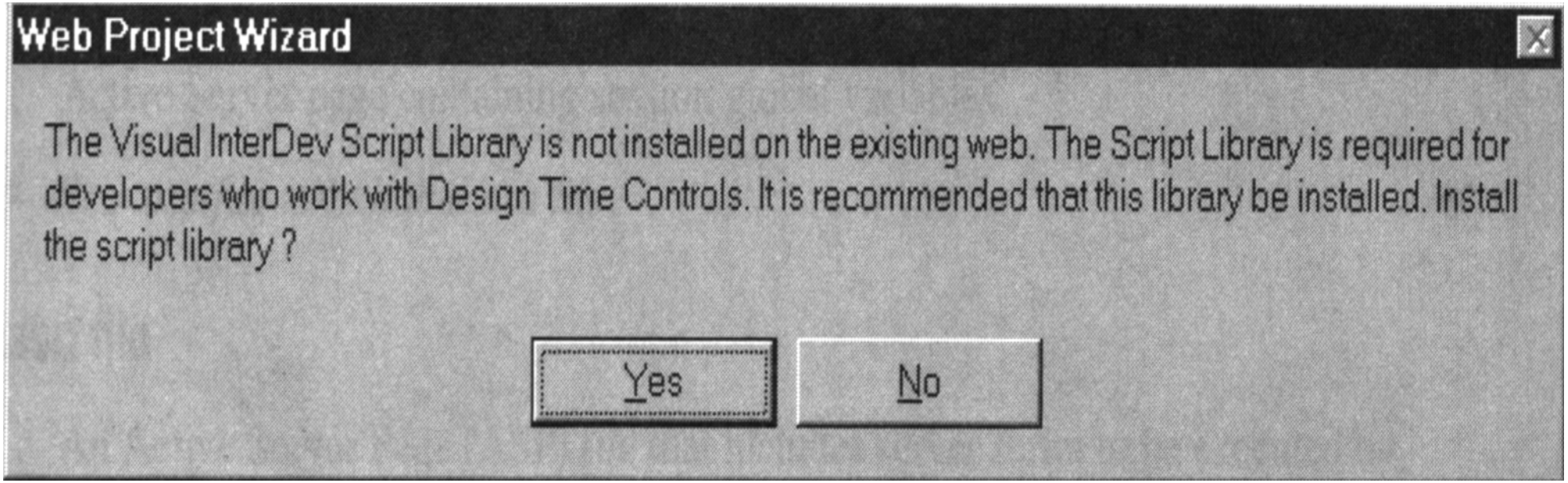
安装设计期间控件脚本库

脚本库是 Visual InterDev 6.0 设计期间控件产生的代码库房。尽管这个库不是 Visual InterDev 1.0 版本的组成部分，但它是 Visual InterDev 6.0 编程模型所要求的。

在 Visual InterDev 6.0 项目中使用设计期间控件的时候，有这个脚本库是很

重要的，因为它保证 Visual InterDev 6.0 设计期间控件功能正常。

当你第一次在 Visual InterDev 6.0 中打开一个 Visual InterDev 1.0 的 Web 应用时，下列信息框提示你安装脚本库。



单击 Yes，便安装这个脚本库。

如果你选择不安装脚本库，你的项目设计期间控件便找不到它们准备生成的代码，因而出现错误。

如果对提示安装脚本库你选择 No，你以后会改变主意的

1.在你的 Web 项目中创建一个文件夹，取名为 _ScriptLibrary。

2.把 Microsoft Visual Studio\VintDev98\ScriptLibrary 文件夹中的内容复制到

这个_ScriptLibrary 文件夹中。

注意:新文件夹有名字_ScriptLibrary 是很重要的。

一旦创建这个新文件夹并把脚本库的内容复制到里面，在 Visual InterDev 6.0 中打开这些设计期间控件的时候，你的 Visual InterDev 1.0 项目便可以使用这些控件。

附录 B　Visual InterDev 术语

Absolute positioning　绝对定位

在当前窗口中使用 x 和 y 坐标规定元素在 HTML 页面上位置的功能。如果一个元素不是绝对定位，则它在页面上的纵向位置就由在 HTML 文本中出现的位置所决定，而它的横向位置由诸如<CENTER>标签 ALIGNMENT 之类的属性所决定。

ADO

ActiveX Data Object（ActiveX 数据对象）的缩写。数据访问的一种跨语言技术，它按照这种技术提出的对象模型把数据连接对象、数据命令对象、记录集对象以及这些对象的集合结合到一起。这个 ADO 对象模型为创建访问数据库中数据的脚本提供容易使用的一组对象、特性和方法。

Alternate text　替换文本

在某些 HTML 元素不能使用的时候用于替换的文本。例如，如果浏览器不能显示图形，它便显示 .gif 文件指定的替换文本。

.asa file　.asa 文件

包含会话全局变量的激活服务器页面。
参见 .asp file 和 Global.asa file。

.asp file　.asp 文件

一种激活服务器页面（ASP）文件，包含由 Microsoft Internet 信息服务器（IIS）执行的服务器脚本。文件上的.asp 扩展名提醒服务器：要先处理文件后发送给浏览器。在 .asp 文件中服务器脚本被服务器处理之后，该文件以与 .htm 文件的同样方式发送给浏览器。

broken link　已断链接

指向一不能定位项的引用子。项不能定位的原因可能是 URL 无效、项指向一个不存在的点或者包含该项的服务器正忙碌或有其它技术困难。

browser　浏览器

对万维网上寄送的 HTML 文件置标进行解释、赋予 Web 页面窗体并显示给用户的软件。如果你有相应设备，有些浏览器还能打开专门程序以演奏声音或在 Web 页面上演示视频文件。

child page　子页

在站点图中父页面所拥有的页面。

client script　客户脚本

由浏览器在用户计算机上执行的脚本。客户脚本是页面的组成部分，在用户请求页面的时候发送给浏览器。客户脚本典型运行是对事件作出响应，例如页面装载或用户敲击按钮等事件；或用于改变页面的外观或验证用户键入的信息。

collapsed item　叠合项

在链接图中其输出链接和输入链接不出现的项。
比较 expanded item(展开项)。

compile-time error　编译期间错误

参见 syntax error(语法错误)。

control　控件，控制

在 HTML 页面上可以被用户操作完成一个动作的项，例如按钮。

cookie　串存单元

在用户计算机中用来存储永久状态信息的小分组。

.css file　.css 文件

包含级联样式图表信息的文本文件。
参见 style sheet(样式图表)。

data command 数据命令

在数据环境中包含有关访问具体数据库对象的对象。例如，一个命令对象可以指向一个表，另一个指向一个存储的过程，第三个指向一个 SQL 命令。

命令对象可在脚本中访问，允许你打开或执行脚本中的基础数据对象。数据命令对象还可重复使用，所以，如果基础数据库变更，你可以使命令对象变更，而对象所涉及的所有页面都依然正常工作。

更详细说明，参见第十八章“数据库概念”。

data connection 数据连接

要求访问指定数据库的信息集合。这个集合包含数据源名字（DNS）和登录信息。数据连接存储在项目的数据环境中，在用户完成一个请求访问数据库的动作时激活。

例如，Microsoft SQL 服务器数据库的一条连接由数据库名字、数据库常驻服务器的位置、访问服务器的网络信息、用户 ID 和用户口令组成。

更详细内容，参见第三章“数据库基础”中的“连接数据库”和第十三章“数据库概念”。

data environment 数据环境

Visual InterDev Web 项目中保存访问数据库中数据所需信息的储存库。数据环境包含一个或多个数据连接。每个数据连接都可以引用一个或多个数据命令，而每个数据命令代表查询或修改数据库的一种方法。

因为数据连接和数据命令信息都被项目中的多个 Web 页面所共享，因而数

据环境是作为项目的 Global.asa 文件组成部分而被管理的。

详细信息，参见第十八章“数据库概念”中的“数据环境”。

database diagram 数据库图

数据库模式各个部分的图形表示。一个模式是数据库管理系统（DBMS）对数据库的描述，而数据库管理系统是用由 DBMS 提供的数据定义语言生成的。一个数据库图可以是数据库结构的整体图，也可以是局部图；它包括表格对象、表格包含的栏和表格与栏之间的关系。

database object 数据库对象

数据库的表格、栏、存储的过程或其它元素。这些对象都可以作为分立项使用。

database project 数据库项目

可以加到 Microsoft Visual InterDev 解决方案中，提供到数据库活化连接，还包括管理和查询数据工具的项目。你可以使用数据库项目创建与修改表格、栏、视图、索引、查询和其它数据库对象。

数据库项目可以和 Web 项目存在于同一个解决方案中，你可以同时在两类项目中工作。但是，两类项目不互相连接，而把数据库功能加到 Web 应用中。为此目的必须用 Web 应用项目分别建立连接。

Dependent project 相关项目

Microsoft Visual Studio 项目，其输出指定为 Web 项目的组成部分。例如，Microsoft Visual J++项目可以加到一个 Web 项目上作为相关项目，同时建在 Microsoft Visual InterDev 中。

deployment 部署

开发者把 Web 页面和相关文件复制到公共服务器上的过程，以及出版、传播和分派 Web 内容的过程。这个过程可以是简单的复制过程，也可以包括建立相关项目。

详细内容，参见第二十八章“Web 应用部署”。

design-time control 设计期间控件

在 Visual InterDev 中设计期间提供图形界面（包括可在 Properties 窗口中设置的特性）的控件。这种控件在运行期间生成 HTML 文本、脚本对象和其它实现所设计功能的部件。

设计期间控件包含标准用户界面元素：文本框、标签、复选框、列表框、命令按钮等，还包括无可视外观但生成运行期间对象（如 Recordset 对象）的控件。设计期间控件还提供数据绑定能力。

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”。

DHTML

动态 HTML。Microsoft Internet Explorer 4.0 中得到支持的 HTML 扩展版本，

提供 Web 页面和作为可编写脚本对象的所有元素。DHTML 允许你不用运行服务器脚本就可在客户脚本中动态改变 Web 页面的外观、内容和表现。

DHTML 在 Visual InterDev 中创建 Web 时非常有用，可以把多媒体效果和客户访问权结合到数据库中。

document　文档

与应用相关的一种文件，例如 Microsoft Word 或 Microsoft PowerPoint。

executable file　可执行文件

提交程序、应用程序、批量文件、脚本和 DLL 的文件。

expanded item　展开项

其输入输出链接出现在链接图中的项。

比较 collapsed item　(叠合项)。

external icon　外部图标

Link View（链接视图）中不属于当前项目组成部分的项的条目图形表示形式。外部图标给出项的一种图像表示，叠加在表示万维网的全局图像上。

external item　外部项

不属于当前项目组成部份的一个项。

external page　外部页面

不属于当前 Web 项目的页面。外部页面在站点图中不能有子页面。

extranet　外部网

只允许一组注册访问者访问的 Web 站点的一个区域。

filter　过滤器

排除与事先规定的一组规定或规范不匹配的信息手段。

global.asa file　Global.asa 文件

Microsoft Internet 信息服务器（IIS）对每个应用维持的文件。在下列情况中，这个文件由服务器自动处理：

- IIS 应用启动和停止。
- 单个用户启动和停止访问应用 Web 页面的浏览器会话。

典型的 Global.asa 文件包括如下用途的脚本：初始化应用变量或会话变量、连接数据库、发送串存单元和完成其它有关整体应用操作。

global navigation bar page　全局导航条页面

出现在全局导航条上作为链接的页面。要想充分利用这个选项，必须在加到站点的文件中或布局中加上 PageNavbar 设计期间控件（DTC）。

global script　全局脚本

不属于事件处理过程或调用功能的组成部分但在页面处理时立即执行的客户脚本。

horizontal layout　横向布局

在展开项左边显示所有输入链接，在展开项右边显示所有输出链接的链接图。

.htm file　.htm 文件

包含 HTML 文本也可能包含客户脚本的文件。Visual InterDev 还用 .htm 扩展名识别这种文件。

参见.asp 文件。

HTML page　HTML 页面

受权使用 HTML（超级文本标记语言），例如 .htm 文件或 .asp 文件的 Web 页面。

in link　输入链接

指向一个展开项的链接。所指的项在链接线的一端，展开项在另一端。

independent page　独立页面

在站点图中没有父页面也没有子页面的页面。

inline script　内联脚本

插入 HTML 文本或客户脚本以便把信息从服务器送给一个文档的服务器脚本。

当指客户脚本时，“内联”是全局脚本的同义语。

internal icon　内部图标

在 Link View（链接视图）中作为当前项目组成部分的项的图形表示。一个内部图标给出一个图像，表示例如 HTML 页面、多媒体文件等项类。

internal item　内部项

作为当前项目组成部分的项。

item　项，条目

构成 Web 站点的一种资源。Link View（链接视图）以图标表示下列项类型：

- HTML 页面
- 样式图表
- 激活服务器页面（*.asp）
- Global.asa 文件
- 激活的布局
- 图像
- 图像映像
- 声音文件
- 视频文件
- 虚拟现实文件
- 可执行文件
- 数据连接
- 数据命令
- 数据范围头部
- 条件范围头部
- 一般设计者控件
- Microsoft Word 文档
- Microsoft Excel 电子表格
- Microsoft PowerPoint 文件
- 其它应用程序
- 其它文本文件
- 邮件
- 新闻
- Telnet
- 其它

just-in-time debugging 及时调试

在脚本中遇到错误时，Microsoft Visual Studio 自动启用调试器的能力。详细内容，参见第二十三章“脚本编程概念”中的“脚本调试过程”。

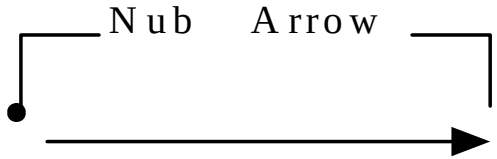
layout 布局

信息、导航条和图形在页面上定位的框图。

link　链接

Web 站点中项之间的关系。链接可以是页面之间的超级链接，或引用一个页面上所包含的文件如图形文件之间的超级链接。Link View（链接视图）根据页面上 HTML 标记和属性确定项的链接。

链接用链接图中的一条线表示。链接的原点用圆点表示，链接的方向用箭头表示，如下图所示。



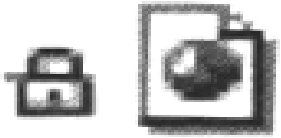
link diagram　链接图

Web 结构的图形表示。链接图用图标表示 Web 站点中的项，例如 HTML 页面；用链接线表示项之间的链接。

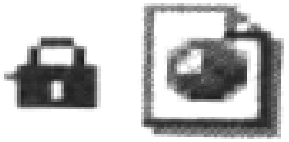
详细内容，参见第十七章“页面样式定制”中的“页面布局”。

local file　本地文件

在本地 Web 服务器上的主文件只读副本。本地文件在 Project Explorer 中是用浅蓝色小锁图标表示的，在 .htm 文件内这种文件符号如下图所示：



如果这种文件在源控件下，则用下列图标表示：



详细内容，参见第六章“Web 项目概念”中的“项目结构”。

local mode 本地模式

项目的一种状态，在这种状态中项目文件的变更保存在开发者工作站的文件副本中，而这种文件副本常驻于本地 Web 应用中，不是常驻于主 Web 应用中。主 Web 应用必须明确用推出的工作副本更新，即做到项目同步。在 Project Explorer 中，本地模式用下述图标表示。

如果项目在源控件之下，你可以用检入文件的办法更新主 Web 应用。该项目用下述图标表示：

参见 master mode 主模式。

详细内容，参见第六章“Web 项目概念” 中的“项目结构”。

local Web application 本地 Web 应用

常驻于开发者工作站上的 Web 页面集合。这些页面用于创建、修改和测试页面，然后把页面发送给 Web 服务器上的主 Web 应用，供内部网或因特网使用。

参见 master Web application 主 Web 应用。

详细内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

local Web server 本地 Web 服务器

处理 Web 页面的服务器，例如 Microsoft Internet 信息服务器。一般这种服务器一般位于开发者工作站上，提供测试所有 Web 元素类型所需要的功能，经测试后元素才发送给主 Web 应用。

详细内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

logic error 逻辑错误

脚本中出现的一种错误，这种错误是指在脚本执行时没有语法错误和运行期间错误，但是得不期望结果的一类错误。例如脚本可能提示用户键入口令，但是即便键入错误口令也允许访问应用。

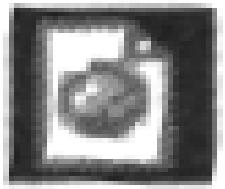
参见 run-time error 运行期间错误，syntax error 语法错误。

marquee 选取框

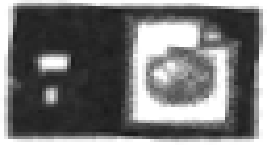
包含文本和其它信息，在定义区域内滚动的 HTML 块。

master file　主文件

项目文件的主要副本。这种副本在主 Web 服务器上，供开发组所有成员使用。在 Project Explorer 中用灰色文件图形表示。在 .htm 文件中这种文件如下列图标所示：



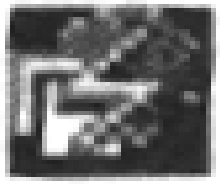
在源控件下，这种文件用下列图标表示：



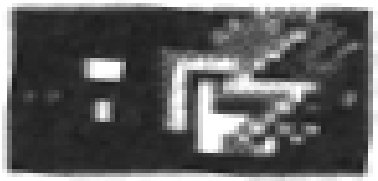
详细内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

master mode　主模式

项目的一种状态，在这种状态中，项目文件变更要保存在开发者工作站 Web 应用的本地文件中和主 Web 服务器上的主文件中。在 Project Explorer 中，主模式是用下面的图标表示的：



如果项目在源控件之下，则用下面的图标表示：



详细内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

master Web application 主 Web 应用

Web 文件的一种集合，这个集合的文件存储在 Web 服务器中，可被多个开发者和作者访问；或者根据你的系统，可被内部网或因特网的用户访问。

参见 local Web application(本地 Web 应用)。

更多内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

master Web server 主 Web 服务器

Web 开发系统中存储与处理 Web 页面主要副本的部件。这个服务器常驻在开发者工作站上或一个单独的机器上。

如果你正使用 Microsoft Visual SourceSafe，你就可以验证主 Web 服务器的名字和路径。

在 Visual SourceSafe Explorer 上，用右键单击检查的文件，选择 Properties，然后选择 Check Out Status 选项卡。该 Computer 名字就是主 Web 服务器名字，而 Folder 名字就是那个机器的绝对路径。

参见 local Web server(本地 Web 服务器)。

multimedia file　多媒体文件

图像、图像映像、声音、视频或虚拟现实文件。

详细内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

navigation bars　导航条

一种图形或文本元素，这种元素把导航链接同作为站点导航结构组成部分的页面结合在一起。可以在 Site Designer（站点设计器）中设计和修改站点导航结构。

navigation structure　导航结构

Web 站点中页面之间的层次关系，这种结构用于确定导航条中到某个站点的链接。

ODBC　开发数据库连接性

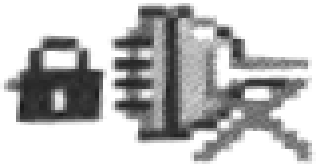
Open Database Connectivity 的缩写。访问基于 SQL 相关数据库的标准协议。

把 ODBC 驱动器软件安装到不同数据库上，就可以连接这些数据库并读写这些数据库的数据。

offline　离线

断开同网络和主 Web 服务器的连接。在 Project Explorer 中，离线方式用下面的图标表示：

如果项目在源控件之下，离线用下面的图标表示：



详细内容，参见第六章“Web 项目概念”中的“项目结构”和第七章“本地工作”。

OLE DB

在企业网络中，从大型机到桌面机(不管数据类型)，提供通用数据继承的数据库结构。在数据访问上，OLE DB 比 ODBC 是更具通用性和更有效的策略，因为它允许访问多种类型的数据，而且是以 COM（部件对象模型，Component Object Model 的缩写）为基础的。

out link 输出链接

指向输出展开项的链接，表示展开项链接到另一个所指的项。

PageNavbar Design-Time Control PageNavbar 设计期间控件

根据存储在站点结构文件中的信息为 Web 项目自动生成导航条的设计期间控件。

page transition　页面过渡

Web 页面上的一种 DHTML 特殊效果，这种效果是在 Web 浏览器中，控制一个页面如何直观替换另一个页面。当页面进入或退出的时候就出现这种过渡效果。

parent page　父页面

在图中有一个或多个子页面的页面。

pending item　悬而未决项

正处于验证处理中的 Link View（链接视图）的一个项。链接视图尚未结束在万维网上按链接所指项的搜寻。

personal file　个人文件

只存在于本地 Web 应用中的文件。这种典型文件尚未加到主 Web 应用中。在 Project Explorer 中，个人文件是用下面的图标表示的：

production server　产品服务器

使用 Web 服务安装的一种 Web 处理单元，例如 Internet 信息服务器，用于处理可被内部网或因特网用户访问的活动 Web 页面。

详细内容，参见第六章“Web 项目概念”中的“项目结构”。

protocol　协议

计算机用于彼此通信的一套正式规则和窗体。FTP 和 HTTP 是两种协议例子，用于在因特网上的计算机之间传输文件。

proxy server　代理服务器

在内部网络和因特网之间起安全传递信息作用的 Web 服务器。这种服务器防止因特网上的用户访问内部网上的信息。

详细内容，参见第六章“Web 项目概念”中的“安全”和第九章“增加安全措施”中的“链接代理服务器”。

radial layout　放射状布局

在一个展开项周围一圈内显示与这个展开项相关的输入输出链接图。

relationship　关系

在站点导航结构中两个或更多个页面所彼此拥有的链接类型。通常有父子关系或同级关系。还可以在一个站点图中定义页面之间的关系。

run-time error　运行期间错误

在脚本中当一个命令企图完成一个无效动作时出现的错误。例如，如果用一个变量完成一项计算，而这个变量尚未初始化，于是出现运行期间错误。如果出现运行期间错误，脚本要么停止要么执行一个特殊例行程序。

参见 logic error(逻辑错误)，和 syntax error(语法错误)。

script 脚本

用户显示 Web 页面时运行的程序。脚本可以由浏览器执行（客户脚本）或服务器执行（服务器脚本）。脚本源代码本身出现在 Web 页面上。

详细内容，参见第二十三章“脚本编程概念”中的“Web 应用中的脚本”。

script object 脚本对象

由设计期间控件生成的脚本所创建的运行期间对象。设置设计期间控件特性所确定的值，在后来的运行期间用于创建对应的动态脚本对象。脚本对象所支持的属性、方法和事件可以用于按照脚本对象模型编写脚本。

详细内容，参见第二十三章“脚本编程概念”中的“脚本对象模型”。

scripting object model 脚本对象模型

对象及其事件、属性和方法的集合，这种集合提供标准面向对象的用于使用 Web 页面的编程模型。脚本对象模型由文档框架和单个设计期间控件组成，它们一起生成运行期间的脚本对象。

脚本对象模型提供一种一致的基于事件的模型，可以跨越客户机和服务器透明工作；脚本对象模型还提供独立于数据库的数据绑定。详细内容，参见第二十三章“脚本编程概念”中的“脚本对象模型”。

server script　服务器脚本

在页面送给请求页面的浏览器之前由服务器执行的 Web 页面中的脚本。在页面发送给浏览器的时候，服务器已运行了这个服务器脚本，并且从页面中把它去掉。典型的服务器脚本完成数据库查询，导航到另一个 Web 页面，或者处理由用户按 HTML 窗体键入的信息。

参见 client script (客户脚本)。

session　会话

用户建立观察 Web 站点的过程。

详细内容，参见 Active Server Pages（激活服务器页面）文档中“Session Object（会话对象）”。

sibling page　同级页面

同一树中与另一个页面共享一个父页面的页面。

site diaggram　站点图

Web 站点中导航结构的图形表示。站点图由一个或多个关系页面树构成。

site strusture file　站点结构文件

Web 服务器上存储 Web 项目页面导航相关信息的文件。使用站点图可以修改 Web 项目的站点结构文件。

site transition　站点过渡

控制浏览器中一个 Web 页面替换另一个 Web 页面的一种 HTML 特殊效果。一个页面从外部页面进入或退出时出现这种站点过渡。

solusion　解决方案

Web 项目和根据这些 Web 项目所组织的 Web 应用程序的集合体。

source page　源页面

在站点图中利用拖动办法同另一个页面建立子页面或父页面关系的页面。

staging server　登台服务器

在 Web 开发系统中在 Web 应用传送给活动产品服务器之前对 Web 应用和 Web 元素进行全面功能测试的 Web 处理单元。详细内容，参见第六章“Web 项目概念”中的“项目结构”。

style sheet　样式图表

描述单个 HTML 标记外观的一组附加标记。样式图表标记描述公共 HTML 标记，例如抬头、段落和列表的字体、段落对齐和其它属性。这些标记可以在文档之中，也可以作为一个单独的文本文件。

当样式图表加到一个 Web 页面中的时候，Web 页面中的所有标记都在样式图表中有对应的条目，因而都自动按照这个样式图表中的描述产生窗体。如果一组样式图表标记在一个单独文本文件中，它们就施加于一个或多个 HTML 文

档中。

级联样式图表（CSS）信息在浏览器中被忽略，因而不支持这种特性。

详细内容，参见第十一章“站点设计”中的“站点一致性”。

syntax error　语法错误

脚本中的一种错误，如打错了键、忘记关闭多行命令（如 DO...LOOP）或类似的错误。如果脚本包含语法错误，脚本便停止执行，并且有一条错误消息立即显示在浏览器上或服务器处理的页面上。

参见 logic error(逻辑错误)和 run-time error(运行期间错误)。

target page　目标页面

站点图中的一种页面，把源页面拖到站点图上某个地方所产生的页面。这个页面可以是源页面的父页面或子页面。

template　样板

起主副本作用的 HTML 页面，或 ASP 页面，或它们的集合。样板可以用来生成新页面，免去手动设计页面外观的麻烦，并且给一系列页面类似的视觉和感觉。还可以为已经存在的页面改换样板，迅速改变页面的外观。

在根据一个样板创建新页面的时候，样板中的文本放置到新文件中。样板还可以包括参数，用来在创建新文件的时候查询用户的定制信息。有关创建自己样板的内容，参见第十七章“页面样式定制”中的“创建定制样板”。

在这里，Web 页面样式，诸如抬头位置、背景图像和颜色等元素都已经设

置好了。

theme　主样板

一组图形、字体和其它页面元素结合起来创建一个一致的可视页面设计。详细内容，参见第十一章“站点设计”中的“站点一致性”。

transaction　事务处理

不管成功或失败，作为整体的服务器操作。这个操作可能有许多步骤，例如订货、清点库存和发出付款帐单等。

tree　树

在站点图中的一组相关页面。

unknown item　未知项

Link View（链接视图）中尚未验证的项。这样的项可能存在，也可能不存在。使用 Verify（验证）命令让 Link View 确定该项的链接有效还是已经断开。

URL　统一资源地址

Uniform Resource Locator 的缩写，用于标识万维网上的资源。

valid item　有效条目

Link View（链接视图）已经确认存在的条目。

virtual root 虚根

作为服务器根下一个子文件夹出现的一个目录，实际上它可能在根下若干级别的一个子文件夹中，也可能在网络上。

虚根的名字可以与这个实际子文件夹名字相同，也可以是服务器上给出的一个别名。例如，在 URL http:/MyWebServer /MyWebApplication 中，“MyWebApplication”便在 Web 服务器上指定一个虚根“MyWebServer”。MyWebApplication 实际文件夹可能是文件：\\MyWebServer\Default Web Site\MyWebApp。

Web

存储在万维网服务器上或某个计算机硬盘上首页及其相关的页、图像、文档、多媒体和其它项。

Web application Web 应用

构成 Web 站点或在一个虚根下构成 Web 站点个别部分的元素集合。在 Visual InterDev 中 Web 应用是由 Web 项目建成的。

Web 项目可以引用单个 Web 应用。例如，可以让一个 Web 站点拥有两个 Web 应用：一个定义在线目录；另一个定义页面，用于按目录管理数据库。

参见 master Web application(主 Web 应用)和 local Web application (本地 Web 应用)。

详细信息，参见第六章“Web 项目概念”中的“项目结构”。

Web element　Web 元素

在 Web 应用中用于显示和控制 Web 内容的任何单元。Web 元素包括页面、图像、应用、窗体、窗体元素、控件和脚本。

Web project　Web 项目

规定 Web 应用中元素的文件集合。可以把一个 Microsoft Visual InterDev Web 应用建在一个 Web 应用中。

Web 项目文件存储在两个地方：服务器和本地计算机。项目还可以是一个大型容器，即解决方案的一部分。

详细内容，参见第二篇“创建 Web 项目”。

Web server　Web 服务器

通常在因特网上起 http 或相关 Web 服务器软件主机作用的计算机。

Web site　Web 站点

组织在一个域下一个或多个 Web 应用的集合。一个 Web 项目可以同 Web 站点中的一个 Web 应用相关联。

working file　工作文件

可在本地 Web 服务器上使用的主文件的读写副本。在 Project Explorer 中，工作文件用如下图所示的笔图标表示：

这个文件通常在源控件之下，而且已经检出。在这种情况下，用一个红色勾号图标表示。

详细内容，参见第六章“Web 项目概念”中的“Web 文件处理”和第七章“本地工作”。