

您感受到Internet的发展趋势了吗？
如果您完全没有网页程序的设计经验，
但有心想要学习ASP.NET数据库应用程序的设计，
却不知从何下手，本书将是您最好的选择！
本书百分之百以Visual Studio .NET 为开发工具，
结合了Visual Basic .NET语言来开发ASP.NET应用系统。
让您在最短的时间之内，从网页程序设计的门外汉，
成长为可以快速开发出专业的网页应用程序的设计师。

ASP.NET

会计系统程序设计

- ASP.NET程序设计
- ADO.NET的架构讨论与设计
- Microsoft SQL Server与ASP.NET程序如何天衣无缝地整合
- Crystal Report结合ADO.NET来设计精美报表的技巧
- 会计的基础概念与会计系统的设计

陈惟彬 编著



科学出版社

www.sciencep.com



ASP.NET 会计系统程序设计

陈惟彬 编著

科学出版社

北 京

内 容 简 介

本书第1、2两章介绍了 Visual Studio .NET 的 Web 开发环境以及一些常用的控件;第3、4章讨论了开发 ASP.NET 的数据库应用程序——ADO.NET,并介绍了用 ADO.NET 链接到不同类型的数据库的方法;第5章介绍了 .NET Framework 中的结构化异常处理机制;第6章介绍了与 SQL Server 沟通的桥梁——Transact SQL 语言;第7~10章介绍利用 Visual Basic .NET 来设计数据库应用程序,联机 and 离线两种存取模式,并深入讨论了 SqlCommand、DataSet、SqlDataAdapter 的用法;最后三章用一个“会计系统”的实例,介绍了一套商业用的 Web 数据库应用系统的开发过程。

*本书可作为高等院校计算机、电子工程、信息科学等专业高年级本科生和研究生的参考书,也可供用 ASP.NET 开发 Web 数据库应用程序的有关人员参考。

本书繁体字版名为《ASP.NET 会计系统程式设计》,由文魁信息股份有限公司出版,版权属陈惟彬所有。本书简体字中文版由文魁信息股份有限公司授权科学出版社独家出版。未经本书原版出版者和本书出版者书面许可,任何单位和个人不得以任何形式或任何手段复制或传播本书的部分或全部。

图书在版编目(CIP)数据

ASP.NET 会计系统程序设计/陈惟彬编著. —北京:科学出版社, 2004
ISBN 7-03-013414-1

I. A... II. 陈... III. 会计-管理信息系统-主页制作-程序设计
IV. F232-39

中国版本图书馆 CIP 数据核字 (2004) 第 042376 号

责任编辑: 朱凤成/责任校对: 都 岚

责任印制: 吕春珉/封面制作: 东方人华平面设计部

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

双 青 印 刷 厂 印 刷

科学出版社发行 各地新华书店经销

*

2004年6月第 一 版 开本: 787×1092 1/16
2004年6月第一次印刷 印张: 25 1/2
印数: 1—4 000 字数: 590 000

定价: 49.00 元 (含光盘)

(如有印装质量问题, 我社负责调换 (环伟))

序

我们应该都有通过 Internet 查资料的经验吧？Internet 上可让我们查询到所要的资料实在是多到我们没有足够的时间可以逐一消化，这意味着 Internet 所代表的另一个意义，就是它背后蕴藏着超大数据库，也因此才让我们有取之不尽、用之不竭的信息。

也许我们曾上网聊天、玩网络游戏、网上购物等，再加上前面所提到的资料搜索，这些都是 Internet 的应用。

信息与应用是 Internet 之所以可以在短短的几年时间风靡全球的两大主因，同时使得网页数据库应用程序的需求日益重要。即使最近这几年全球经济陷入低迷，也丝毫没有使网络的发展与应用有任何的停顿。

身为世界软件大厂的 Microsoft 当然也体会到了这种趋势。因此其为了进一步抢占 Internet 平台，发表了新一代的网页应用程序设计语言 ASP.NET，而 ADO.NET 则是专为这个网络上复杂的环境所全新设计的数据存取结构，Visual Studio .NET 则是让我们可以快速开发应用程序的优异工具。ASP.NET+ADO.NET+Visual Studio .NET 组合起来就是一个快速开发网页数据库应用程序的环境。有了这些锐不可挡的工具组合，就可以让我们快速开发出符合应用的网页数据库应用程序，以满足每天各地不同网络用户对网络服务的需求。

ASP.NET 的书籍非常多，但严格讲起来，除了书名不同之外，谈的内容都差不了太多。对于想要编写 ASP.NET 应用程序的初学者而言，学完整本书还有可能不知道从何下手，而这些就是我撰写本书的主要目的。

这本书秉持着过去写作的一贯风格，用最简单的方式，由浅入深、一步一步引导大家利用 ASP.NET 来开发 Web 数据库应用程序。除了告诉大家该怎么使用 ASP.NET 来开发应用程序之外，最后还以可应用在实务上的项目实例来一一验证所讨论的概念与知识。

阅读这本书之前，我假设您已经略懂 Visual Basic 语言，因为本书是以最简单的 Visual Basic 语言来讲解的，而对于 Visual Studio .NET 的操作环境也应稍有概念。假如您缺少这方面的经验，可能您需要摆一本 Visual Basic .NET 的入门书籍在旁边来辅助您的学习。至于有没有撰写过网页程序的经验则不是很重要，因为本书是假设您完全没有 ASP.NET 网页数据库应用程序的撰写经验而写的。

当然，ASP.NET 的内容之多，不是这本书就可以全部讲完的。若要全部都讲，则恐怕这本书又变成参考手册。我希望是从入门与应用的角度来切入这本书的内容。在一步步学习 ASP.NET 的过程中，当具备足够开发应用系统的知识之后，就应该试着做出一套应用系统，而不是等到全部学完所有 ASP.NET 的内容才去开发应用系统。在学习与

实践之间，培养出坚固的基础实力后，再进一步向 ASP.NET 更深一层的内容挺进，不是在学习上更有信心吗？

最后，感谢您购买本书，也希望本书真的能带给您一些实质上的收获。

陈惟彬

2003 年 3 月

目 录

第 1 章	Hello, Internet	1
1.1	晋升为网页程序设计师.....	1
1.2	Visual Studio .NET 与记事本	5
1.3	以 Visual Studio .NET 设计的 Web 项目的组成	7
第 2 章	装饰网页	11
2.1	Label 控件	12
2.2	Literal 控件	15
2.3	Button 控件	17
2.4	Image 控件	21
2.5	HyperLink 控件	23
2.6	LinkButton 控件	24
2.7	ImageButton 控件	26
2.8	TextBox 控件	28
2.9	CheckBox 控件	30
2.10	RadioButton 控件	33
2.11	Panel 控件	35
2.12	CheckBoxList 控件	37
2.13	RadioButtonList 控件	39
2.14	Table 控件	41
2.15	DropDownList 控件	44
第 3 章	初探 ADO.NET	48
3.1	什么是 ADO.NET	48
3.2	ADO.NET 的对象模型	49
3.3	联机存取与离线存取	51
3.4	第一个 Web 数据库应用程序	52
3.5	初探客户数据管理程序	57
3.6	命名空间 (Namespace)	61
第 4 章	使用 Connection 对象来连接到数据源	66
4.1	选择适合的联机方式	66
4.2	如何联机到数据库	67
4.3	联机到 Microsoft Access	69
4.4	联机到 Microsoft Excel	71
4.5	用可视化的方式建立 Connection 对象	75
4.6	连接到一个 XML 文件	77

4.7	探索数据库的信息.....	79
4.8	Connection 对象的事件	81
4.9	数据库的 Pooling	83
第 5 章	结构化异常处理.....	92
5.1	结构化异常处理.....	92
5.2	基本的异常处理.....	94
5.3	SQL Server 的异常处理.....	102
5.4	InfoMessage 事件.....	108
第 6 章	Transact SQL 程序设计	112
6.1	SQL 基本介绍	112
6.2	数据库操作语言——SELECT 查询指令	116
6.3	INSERT 插入记录的 SQL 指令	123
6.4	UPDATE 修改记录的 SQL 指令.....	124
6.5	DELETE 删除记录的 SQL 指令.....	125
第 7 章	探索 Internet 无穷的宝藏	126
7.1	SqlCommand 对象.....	126
7.2	动态查询.....	132
7.3	存储过程.....	138
7.4	在 Visual Studio .NET 中建立存储过程	141
7.5	用 SqlCommand 对象来修改数据.....	144
第 8 章	ADO.NET 的离线存取模型	150
8.1	ADO.NET 的离线存取模型	150
8.2	建立数据库与加入表格.....	152
8.3	修改数据.....	162
8.4	建立 DataTable 间的关联与条件限制	174
8.5	DataView 对象.....	189
第 9 章	维护实际的数据库.....	199
9.1	建立测试数据库.....	199
9.2	DataAdapter 对象	204
9.3	深入数据修改的流程.....	225
9.4	Master-Detail 程序设计	243
第 10 章	数据的交易控制.....	263
10.1	认识数据库的交易管理.....	263
10.2	SQL Server 的交易管理.....	264
10.3	ADO.NET 的交易处理	265
10.4	数据交易的隔离层级.....	268
第 11 章	会计系统：系统分析与建立数据库.....	275
11.1	会计系统的系统分析.....	275
11.2	建立数据库.....	289

11.3	触发器.....	292
11.4	规则.....	312
11.5	用户自定义函数.....	315
第 12 章	会计系统：主菜单与数据输入操作的设计.....	321
12.1	主画面.....	321
12.2	AC110 会计科目类维护.....	322
12.3	AC120 会计科目维护.....	330
12.4	AC130 支票输入.....	353
第 13 章	会计系统：报表设计.....	370
13.1	电子表格.....	371
13.2	损益表.....	381
13.3	资产负债表.....	384
13.4	收入统计表.....	386
13.5	费用统计表.....	388
13.6	支票打印.....	390
13.7	日记账.....	393
13.8	明细分类账.....	395
13.9	总分类账.....	399

第 1 章 Hello, Internet

本章假设大家都还不会设计网页应用程序，将用最简单的网页设计来带领大家熟悉 Web 应用程序设计的世界。除此之外，我们还要对 Visual Studio .NET 的 Web 应用程序项目的组成作概括性的介绍。

1.1 晋升为网页程序设计师

现在就先让我们用两种方式向 Internet 问好。

第一种方法是：

1. 首先，请打开记事本。
2. 接着，键入以下内容：

```
<HTML>
<BODY>
Hello, Internet!
</BODY>
</HTML>
```

3. 输入完毕后，请把它保存，文件名为 Hello.html，然后关闭记事本。

4. 打开 Internet Explorer，在地址处输入刚才我们第一个网页处女作的完整文件名与路径，如图 1.1 所示。

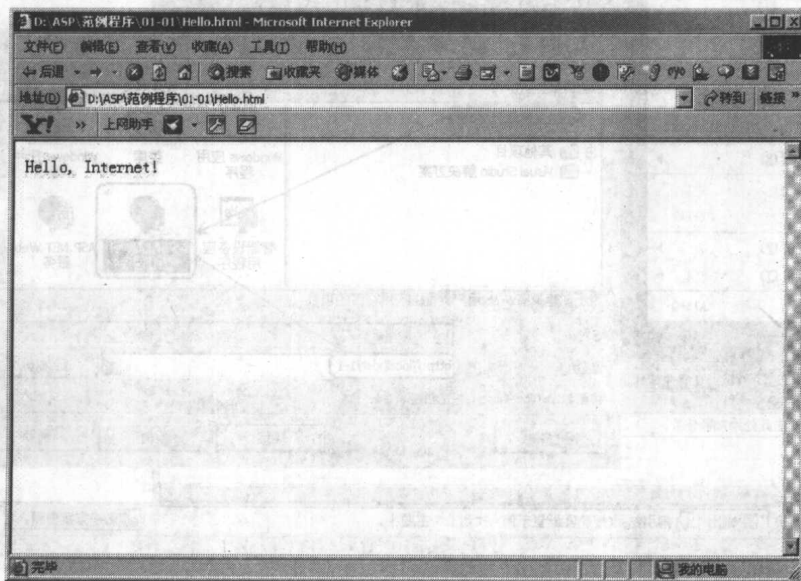


图 1.1 Hello.html 的执行结果

5. 试着单击 Internet Explorer 菜单【查看】→【源文件】，则会用记事本打开我们

刚才的处女作 Hello.html 的内容。HTML 文件内容说明如表 1.1 所示。

表 1.1 HTML 文件的内容

标 记	说 明
<HTML>与</HTML>	用来表示<HTML>与</HTML>之间的内容是一份 HTML 文件
<BODY>与</BODY>	用来表示<BODY>与</BODY>之间的内容, 在经过浏览器解码后, 会显示在浏览窗口中

现在, 让我们用第二种方法来完成“Hello, Internet!”。

1. 运行 Microsoft Visual Studio .NET。

2. 单击菜单【文件】→【新建】→【项目】, 会出现如图 1.2 所示的【新建项目】对话框。在【项目类型】中选择【Visual Basic 项目】, 【模板】选择【ASP.NET Web 应用程序】。在【位置】输入 http://localhost/1-1。位置的输入内容应包括了应用程序的完整 URL 地址, 包括 http://、服务器的名称以及项目的名称, 而且我们所连接的 Web 服务器必须是 IIS 5.0 版本以上或者是更高的版本, 且安装有 .NET Framework。如果我们的计算机已经安装 IIS, 则可以将服务器指定为 http://localhost。如本例, 我的计算机已经装有 IIS, 所以直接用 localhost 来代替 Web 服务器的名称, “1-1”就是项目的名称(本书一律以 Visual Basic 语言作为程序设计的语言, 同时都是以开发 Web 应用程序为主, 所以在后面的章节, 本步骤就不再赘述)。最后单击【确定】按钮来完成新建项目的对话框。

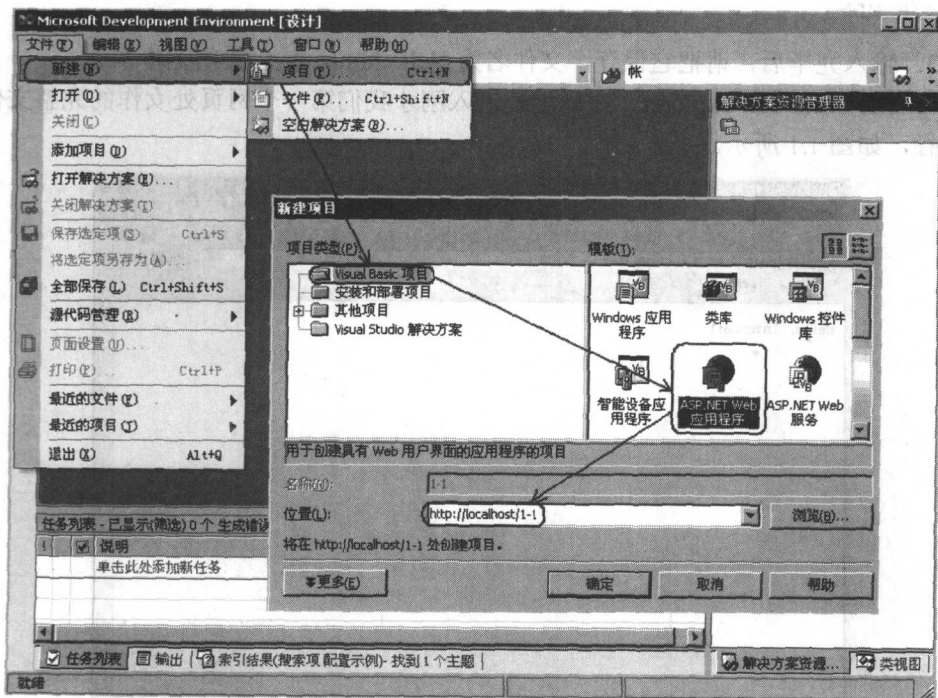


图 1.2 新建 ASP.NET Web 项目

3. 这时, 我们就会看到如图 1.3 所示的画面, Visual Studio .NET 将试着与 Web Server 联系。

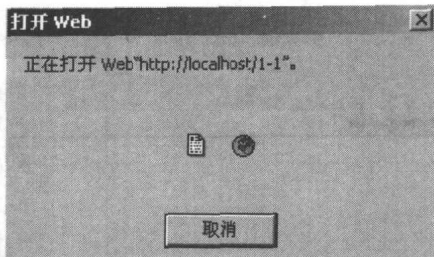


图 1.3 Visual Studio .NET 正试着创建新的 Web 项目文件夹

4. 假如发现已经有“1-1”的项目文件夹，则会出现如图 1.4 所示的画面要求我们换一个项目文件夹的名称。我们可以选择接受 Visual Studio .NET 所建议的新项目文件夹名称，或者是更改一个自己喜欢的项目文件夹名称，当然我们也可以更改已经存在的项目文件夹名称，甚至是删除都行。不过更名的方法可能不太妥当，因为这种方式极有可能造成我们原先的 Web 应用程序项目再次执行时会出现问题，所以不管如何，千万不能覆盖它。

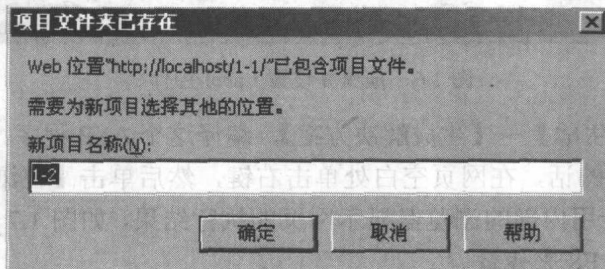


图 1.4 项目文件夹已经存在

5. 比较惨的情况是像出现类似“Web 访问失败”的问题，如图 1.5 所示。当我们看到这个画面时，就要去 Windows 2000 里的“Internet 服务管理器”，看看设置有没有出现问题。还有，看看是否装了 Internet Information Server。

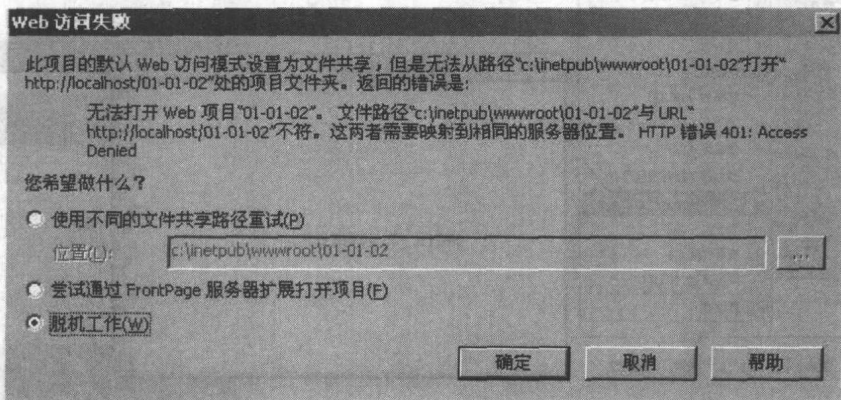


图 1.5 Web 访问失败

6. 参考图 1.6，从工具箱的 Web 窗体控件中，用鼠标拖曳一个 Label 控件到网页设计窗体中。在属性窗口找到 Text 属性，改为“Hello, Internet!”。

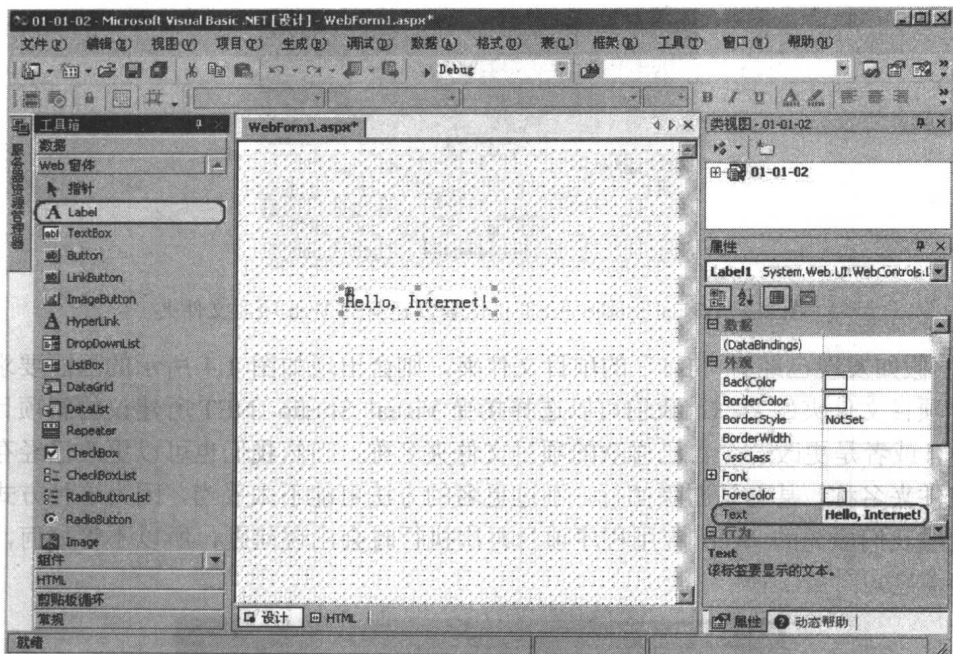


图 1.6 拖曳并设置 Label 控件

7. 单击菜单【生成】→【生成解决方案】，编译这个 ASP 程序。

8. 假如没有错的话，在网页空白处单击右键，然后单击【在浏览器中查看】，则 Visual Studio.NET 会用内部的浏览器显示网页的执行结果，如图 1.7 所示。这个步骤也可以按快捷键 Ctrl+F8 来代替。

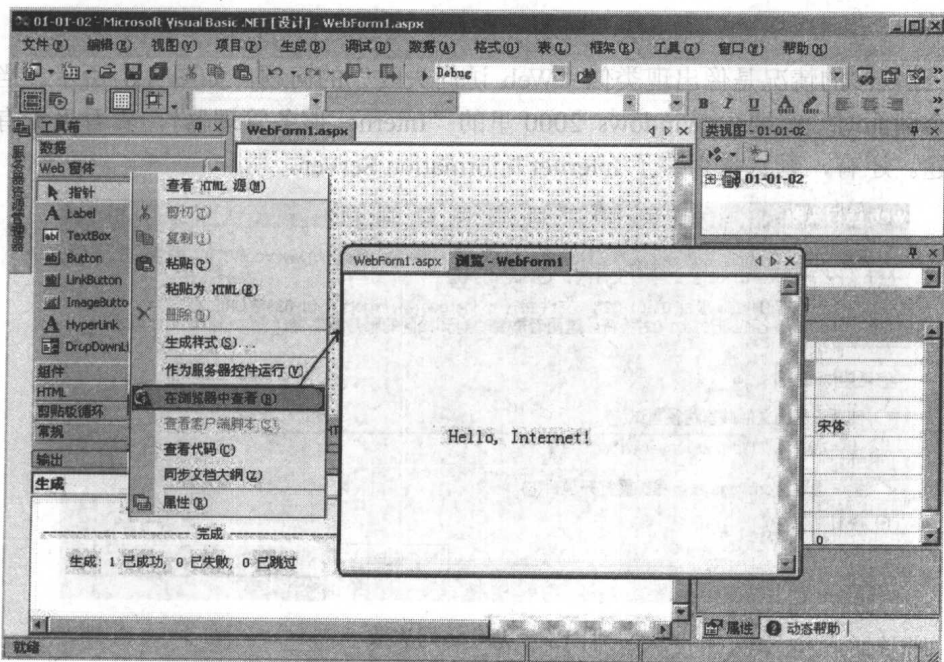


图 1.7 使用 Visual Studio .NET 所设计的网页

对照图 1.1 的结果，我们发现两者结果似乎是相同的。至于“Hello, Internet!”位

置会有所不同,则是因为我们在 Visual Studio.NET 中是以绝对位置来设计的,所以只要调整一下就行了。

另外,单击工具栏上的【启动】按钮,按快捷键【F5】也可以实现编译网页并执行的功能,不同在于是以 Internet Explorer 来浏览网页的执行结果,而不是用 Visual Studio.NET 内部的网页浏览器。若要一律用 Internet Explorer 来浏览网页,则我们可以在解决方案资源管理器中单击右键,单击【浏览方式】,然后在【浏览器列表】的对话框中选择 Microsoft Internet Explorer,如图 1.8 所示,再单击【设为默认值】,最后单击【关闭】来结束对话框。以后每次在浏览网页时,都会用我们所指定的浏览方式来浏览。

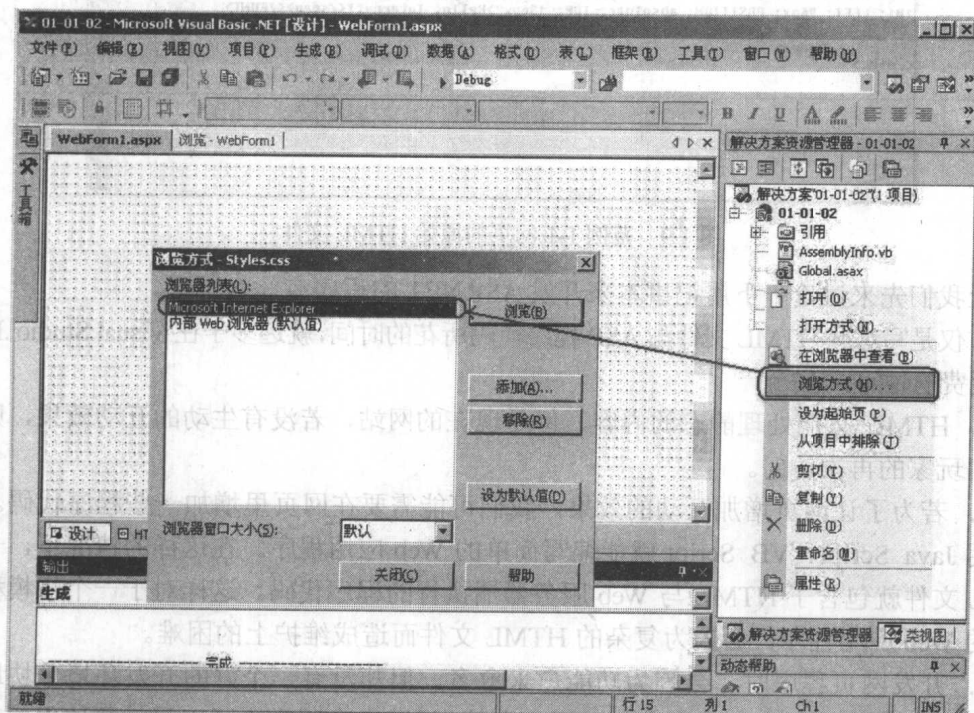


图 1.8 自定义专属的浏览方式

1.2 Visual Studio .NET 与记事本

用 Visual Studio .NET 来开发网页,或者是直接用记事本来写 HTML,哪一种方法比较好呢?在回答这个问题之前,让我们先打开 Internet Explorer,然后在地址输入范例 1-1 的完整路径。接着单击菜单【查看】→【源文件】,这时会用记事本显示该网页的原始 HTML 文件的内容,如图 1.9 所示。

若要说明用记事本来开发 ASP.NET 网页的程序,当然可以,因为 HTML 是组成网页的核心,而它又只不过是文字文件的格式,只要我们写的内容能让浏览器“看”懂就可以了。但是,整个开发过程将会变得很辛苦,除非我们无法购买到 Visual Studio .NET 这套工具,否则真的不建议这么做。想想看,当网页的内容变得非常复杂的时候怎么办呢?

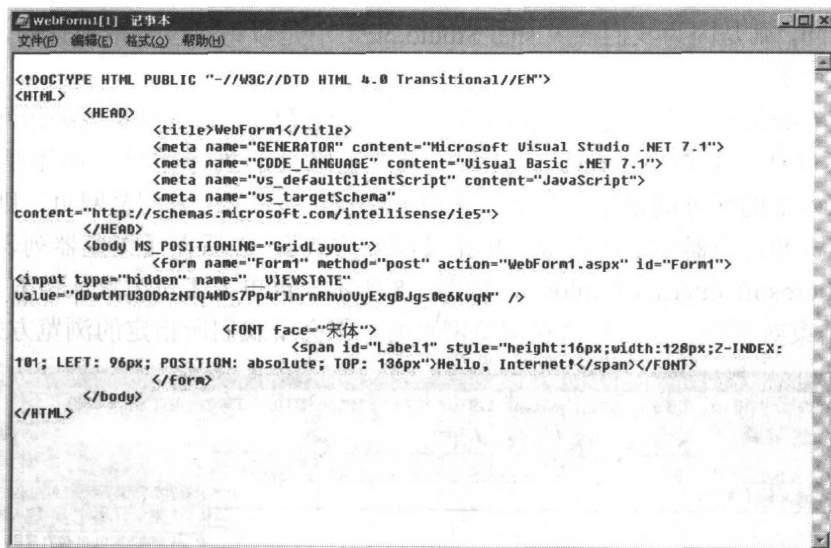


图 1.9 范例 1-1 网页的原始 HTML 文件

让我们先来讨论一下用记事本来开发 ASP.NET 的程序有什么不妥。

1. 仅是写这些 HTML 文件与 ASP 程序代码所花的时间,就远多于在 Visual Studio .NET 上所耗费的时间。

2. HTML 只能处理静态的内容,但是现在的网站,若没有生动的互动效果,则很难吸引玩家的再次访问。

3. 若为了让网页增加互动的效果,我们可能需要在网页里增加一些程序代码。例如,用 Java Script、VB Script 就能编写简单的 Web 应用程序。在这样的情况下,一个 HTML 文件就包含了 HTML 与 Web 服务器端执行的程序代码。这样对于一个规模稍微庞大的 Web 应用程序,会因为复杂的 HTML 文件而造成维护上的困难。

4. 开发网页程序时,会因为功能愈来愈多,但却没有一个好的开发环境来协助而导致除了开发的程序 Bug 发生率相对提高之外,调试也会因为少了工具的帮忙而困难许多。

其实,用开发 Windows 应用程序的开发方式,可以更浅显地来说明用记事本与 Visual Studio .NET 的优劣。我们用记事本写一个 C 语言的“Hello! World.”,还是用 Visual C++、或者是 C++ Builder 来开发呢?前者我们要写一堆的 SDK,但后者我们却无须担心这些问题,因为开发工具都已经帮我们包装好了所有复杂的底层。相同的,我们也无须在和过去一样去处理 HTML 接口与 Script 整合的问题,只要专心于程序的编写即可。换句话说,Visual Studio .NET 会让我们体验到设计 Web 应用程序就像设计过去 Windows 应用程序般的快乐与容易。

若要问用记事本来开发 Web 应用程序有什么优点呢?针对这个问题,我曾经和几位从 Windows NT 4.0 就开始写网页程序的资深 Web 应用程序设计师聊过,他们认为到现在还是用记事本来开发 Web 程序的原因:

1. 自己写的程序代码比较简单,少了 Visual Studio .NET 那么多奇奇怪怪的东西,这样客户端在下载网页时,所花的时间会比较短。

2. 已经习惯于用记事本来开发应用程序。对于 Visual Studio .NET, 通常是用来调试程序。

3. 有一些 Visual Studio .NET 做不到的功能。

对于他们所讲的, 我们可以参考一下。

既然用 Visual Studio .NET 或者是记事本皆可以开发 ASP 的应用程序, 那么, 这两种方式有什么差异呢?

若用记事本来开发, 则所有与网页有关的 HTML 及 ASP 程序都是放在同一个文件中。若用 Visual Studio .NET 来开发, 则 HTML 文件会是一个后缀名为.aspx 的文件, 如果接着用 Visual Basic 或 Visual C# 来开发 ASP 的程序, 则会伴随着一个.vb 或.vs 的文件。

总之, 我们还是强烈建议用 Visual Studio .NET 来全程开发 Web 应用程序。

1.3 以 Visual Studio .NET 设计的 Web 项目的组成

现在让我们来进一步了解 Visual Studio .NET 项目的组成。再回头看看范例 1-1 的解决方案资源管理器, 会看到一个 Web 项目是由许多文件组成的, 如图 1.10 所示。.aspx 文件是系统的重点, 也就是说, 每个网页, 都会有一个.aspx 文件。就像本例, WebForm1.aspx 是类似 HTML 文件的网页文件。

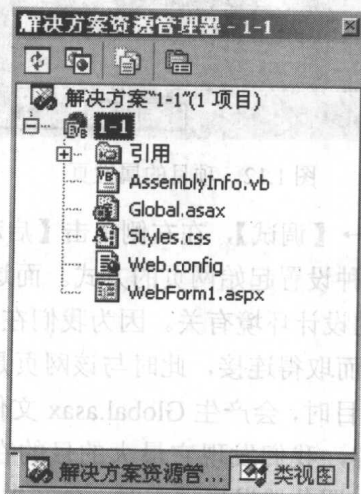


图 1.10 解决方案资源管理器

当然, 像 WebForm1.aspx 这种文件名不太好辨识, 所以我们可以直接在解决方案资源管理器中单击 WebForm1.aspx, 然后单击右键选择【重命名】。此时, 若重新再编译并执行本网页, 则会出现错误, 如图 1.11 所示。这是因为原来默认 WebForm1 是我们的起始网页, 但是在我们更名后, Visual Studio .NET 并没有把起始网页一并变更, 所以才会有这样的问题。

我们可以就简单的依照信息的指示, 在解决方案资源管理器中单击改动后的网页, 然后单击右键, 再单击【设为起始页】, 或者在解决方案资源管理器中单击项目名称,

然后单击右键，再单击【属性】，会出现如图 1.12 所示的属性页。

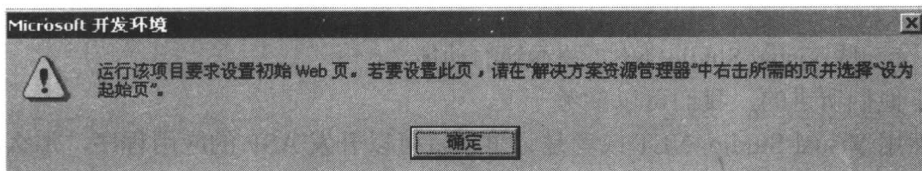


图 1.11 起始网页名称改动后的错误

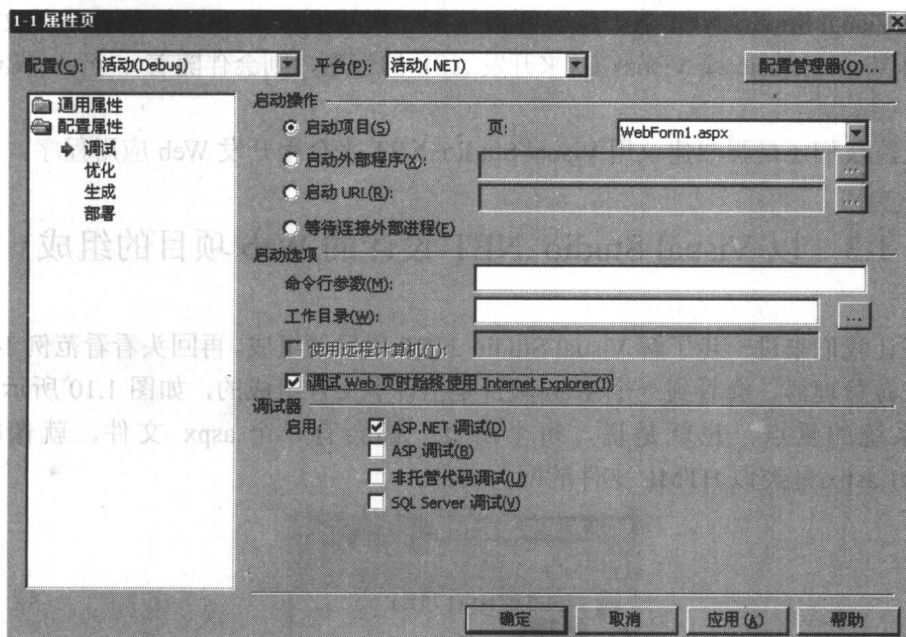


图 1.12 项目的属性页

在左边单击【配置属性】→【调试】，在右侧单击【启动项目】，在网页的下拉列表选择要启动的网页名称也是一种设置起始网页的方式。而这两种设置起始网页的方式，似乎只与 Visual Studio .NET 的设计环境有关。因为我们在用 Internet Explorer 浏览网页时，可以直接输入网页的名称而取得连接，此时与该网页是不是起始网页没有关系。

接着，当我们建立 Web 项目时，会产生 Global.asax 文件。这个文件的内容如图 1.13 所示。由这个程序代码的内容，我们发现它最大的目的在于处理 Application 对象与 Session 对象。每当用户向 Web 服务器发出请求时，会产生一个 Session 对象，而第一个 Session 对象发生时，会先建立 Application 对象。换句话说，每个 Web 项目在执行时，会生成一个 Application 对象，以及一些 Session 事件。

Global.asax 主要的目的是可以让我们处理应用系统相关的共享文件与设置。例如，假设我们有一个统计目前连接的用户变量 TotalUsers，则我们可以在 Application 的 OnStart 事件，将这个变量设为 0。因为每个用户连接都会发出请求，故第一个请求发生前会触发该事件，所以我们可以先将变量归 0，然后在每个 User 连接进来时再分别累加。至于 Application 对象与 Session 对象的意义与用途，现在可以不必了解，我们在稍后的章节会适时的进一步做深入的说明。

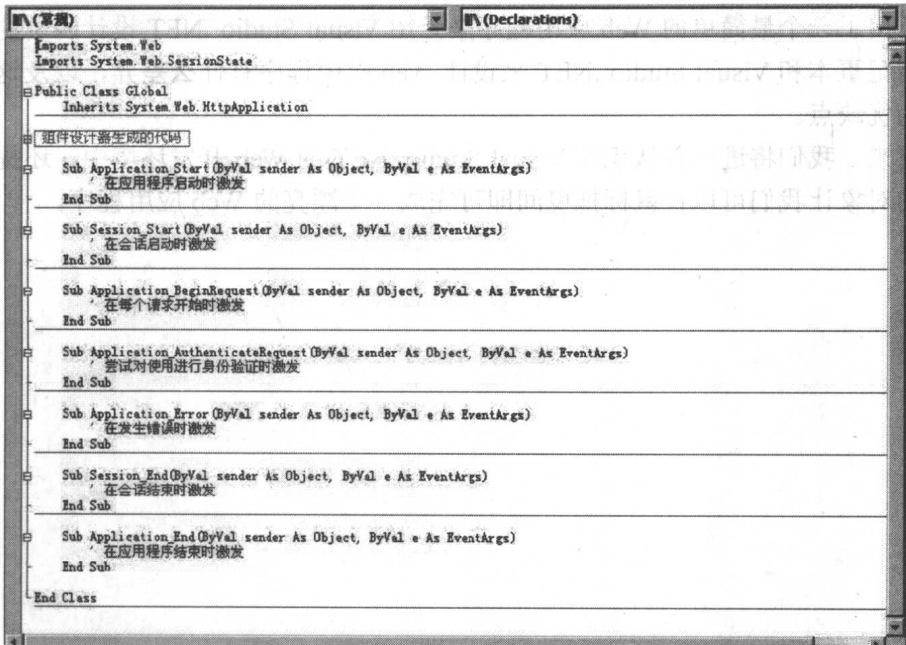


图 1.13 Global.asax 的内容

Web.config 用来定义我们 Web 应用系统的组态。这个文件不可以更改它的名称，也没有辅助工具协助我们去维护，因为在大部分的情形下，我们很少需要去手动改变它的内容。图 1.14 就是 Web.config 文件的部分内容，我们可以发现，它的组成与 HTML 是相似的，也是由一个个的标记所组成。

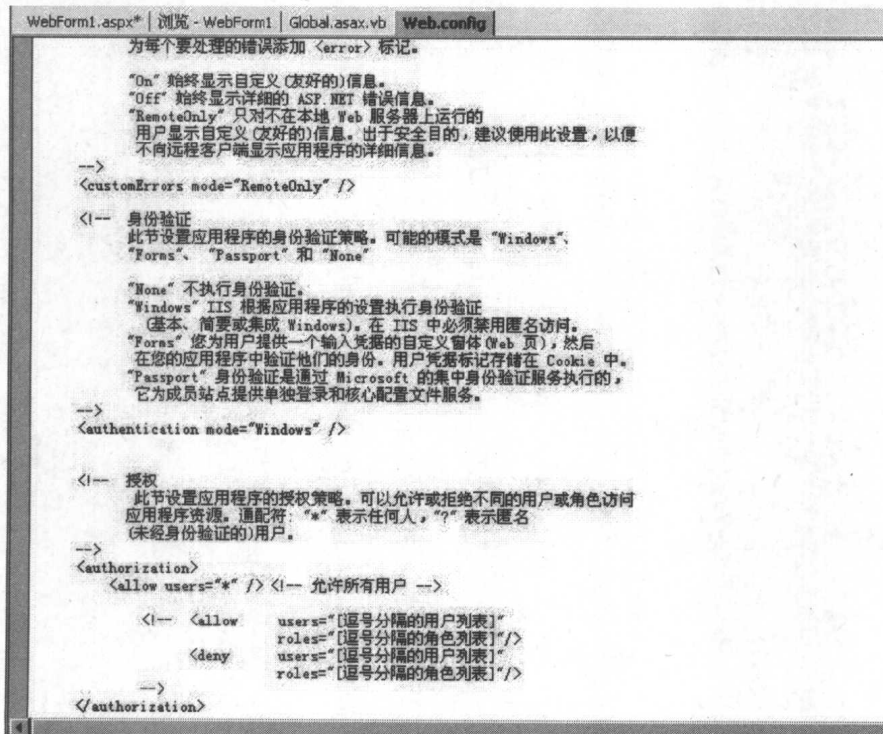


图 1.14 Web.config 文件的部分内容