

MAPGIS7-IMS 行业版 二次开发技术篇

目录

目录.....	- 2 -
1 概述.....	- 3 -
2 .NET开发环境介绍.....	- 3 -
2.1 .NET开发环境.....	- 3 -
2.1.1 行业版的开发环境.....	- 3 -
2.1.2 行业版选择的语言类型.....	- 4 -
2.1.3 行业版所用的文件类型.....	- 4 -
2.2 开发软件---Visual Studio.NET 2005.....	- 4 -
3 JAVA开发环境介绍.....	- 10 -
3.1 JAVA开发环境.....	- 10 -
3.1.1 行业版的开发环境.....	- 10 -
3.1.2 行业版的开发语言.....	- 10 -
3.1.3 行业版所用的文件类型.....	- 11 -
3.2 开发软件---Eclipse.....	- 11 -
4 行业版二次开发技术.....	- 21 -
4.1 HTML语言.....	- 21 -
4.2 CSS样式表.....	- 22 -
4.3 JavaScript脚本语言.....	- 24 -
4.4 DOM模型.....	- 27 -
4.5 AJAX技术.....	- 28 -
4.6 VML语言.....	- 31 -
4.7 XML语言.....	- 32 -
4.8 JSP技术.....	- 33 -
4.9 数据库.....	- 34 -
5 程序调试方法.....	- 35 -
5.1 客户端脚本调试.....	- 35 -
5.2 .NET服务器端程序调试.....	- 40 -
5.2.1 调试的快捷键.....	- 43 -
5.2.2 调试方法.....	- 43 -
5.3 JAVA服务器端调试方法.....	- 48 -
5.3.1 调试的快捷键.....	- 48 -
5.3.2 调试方法.....	- 48 -

1 概述

MAPGIS7-IMS 行业版是以 MAPGIS7.0 平台为基础开发的新一代 WEBGIS 软件，它依托于 MAPGIS7.0，将 MAPGIS7.0 的强大功能应用在网络服务中。

MAPGIS7-IMS 行业版的推出是在 MAPGIS7-IMS 专业版(针对矢量地图)、MAPGIS7-IMS 搜索引擎版(针对栅格地图)的基础上，将两者的功能结合在一起，成为一个结合性的开发平台，更能满足广大用户的需求。

现今 WEB 开发的主流技术就是 .NET 技术和 JSP 技术，根据其不同的优点，把这两个技术用到不同的领域发挥所长。为进一步满足不同用户二次开发的需求，MAPGIS7-IMS 推出了两个方向的二次开发环境，一个是基于 .NET 与 C# 搭配的 .NET 环境开发平台；另一个是 Eclipse 和 JSP 搭配的 JAVA 环境开发平台。在两个开发环境中都提供全面的技术支持，用户可以根据自己的需要和项目的领域需求灵活选择开发环境。

IMS 平台使用了很多技术来帮助用户进行二次开发，CSS 技术、DOM 模型、AJAX 技术等。这些技术更优化系统，使之具有速度快、页面无刷新、处理专业级用户、专业数据的空间分析及查询、基于全国地图搜索等优点。

下面的章节会对两个方向的开发环境、二次开发的主要技术以及如何实现两个开发环境下的程序调试方法等方面入手，讲述 MAPGIS7-IMS 行业版二次开发。

2.NET 开发环境介绍

.NET 代表一个集合，一个环境，一个可以作为平台支持下一代 Internet 的可编程结构。即：.NET=新平台+标准协议+统一的开发工具。在二次开发中，采用 Visual Studio.NET 2005 开发工具，C# 开发语言等组成的一个 .NET 开发环境，以 MAPGIS7-IMS 行业版为开发平台，用户能够更快速、方便的开发一些满足需求的新功能。为使用 .NET 开发环境，需要对环境做一些配置，且不同的操作系统需要配置不同环境。下面将列出在 Windows 2003 Server 操作系统下需要的环境配置。Windows 2003 Server 操作系统提供了强大的组件、类库、接口等，非常适用于软件开发。因此，在下面章节将对 Windows 2003 Server 操作系统下需要的环境配置作全面的介绍。

2.1 .NET 开发环境

MAPGIS7-IMS 行业版中应用 ASP.NET 开发环境的配置如下：

1. 操作系统： Windows 2003 Server（推荐）
2. 浏览器： IE 6.0 及以上版本
3. 环境支持： .Net FrameWork 2.0 和 IIS5.0（或更高版本）
4. 数据库： ORACLE 9i 或 SQL Server 2000

2.1.1 行业版的开发环境

行业版的 .NET 开发环境配置如下：

1. 按上述配置好 ASP.NET 开发环境

2. 开发语言：C#语言和 JavaScript 脚本语言
3. 开发工具：Visual Studio.NET 2005
4. 准备好地图数据和相关的点位信息数据，并参照行业版的配置文档配置好 GIS 服务器和相关数据。示例中自带地图数据和点位信息数据，点位信息数据以 SQL Server 数据库文件形式存放。如果使用自带的点位信息数据时，要将 CityPosInfo_Data.MDF 附加到本地 SQL Server 2000 数据库里。
5. 站点开发过程中必须保持 GIS 服务器 (GridMapSvr 和 MapSpecialSvr) 呈启动状态。

2.1.2 行业版选择的语言类型

在.NET 开发环境中，行业版开发所选择的语言类型为 C#语言。

C#是在 C 和 C++语言基础上开发的在.NET 平台上运行的为适应 internet、intranet 和各类网络应用而设计的编程语言，该语言综合了 C、C++和 Java 以及其它高级语言的优点，是一种语言优雅、类型安全、完全面向对象的编程语言。由于 C#语言是专门为.NET 平台而设计的开发语言，因此它比任何在.NET 平台下提供的其它语言都有无可比拟的优越性。这也是行业版选择 C#语言做开发语言的原因。

2.1.3 行业版所用的文件类型

MAPGIS7-IMS 行业版中用到的文件类型如下所示：

表 2-1 .NET 开发环境的文件类型

文件扩展名	用途及说明
Global.asax	ASP.NET 系统环境设置文件。
.aspx	内含 ASP 程序代码的文件，通常所说的网站后台文件。
.htm	静态网页文件。
.css	样式文件,存放网站样式的样式表。
.cs	ASP.NET 代码分离后的后台文件。
.js	Javascript 编写的脚本文件。
.ashx	用来处理生成动态图片、生成动态文本等不需要回传处理的任务。
.config	配置文件。存储一些全局处理信息，比如：数据库连接串等全局信息。

2.2 开发软件---Visual Studio.NET 2005

Visual Studio.NET 2005（以下简称 VS05）是 Microsoft 公司最新推出的先进的可视化开发工具，可引导用户快速掌握用 C#语言进行各类网络应用程序编程的方法和技巧。其主要特点如下：

- 兼容 .NET 公共语言运行库所支持的任何语言。
- 基于 Microsoft .NET Framework 生成。它提供了该框架的所有优点，包括托管环境、类型安全性和继承。
- Visual Studio 支持快速应用程序开发 (RAD) 工具对窗体进行设计和编程。

- 可使用为 Web 开发提供 RAD 功能的控件进行扩展，从而使您能够快速创建多样式的用户界面。
- 具有灵活性，因为您可以向它们添加用户创建的控件和第三方控件。

下面就来介绍在 VS05 中如何新建一个 WEB 站点，流程如下：

1. 首先打开 VS05 软件，点击“开始”→“程序”→“Microsoft Visual studio 2005”。
2. 开始新建网站：选择文件→新建→网站（如图 2-1）；也可在 VS05 “起始页”左边的“最近的项目”区域里创建，即点击“创建→网站”（如图 2-2）。

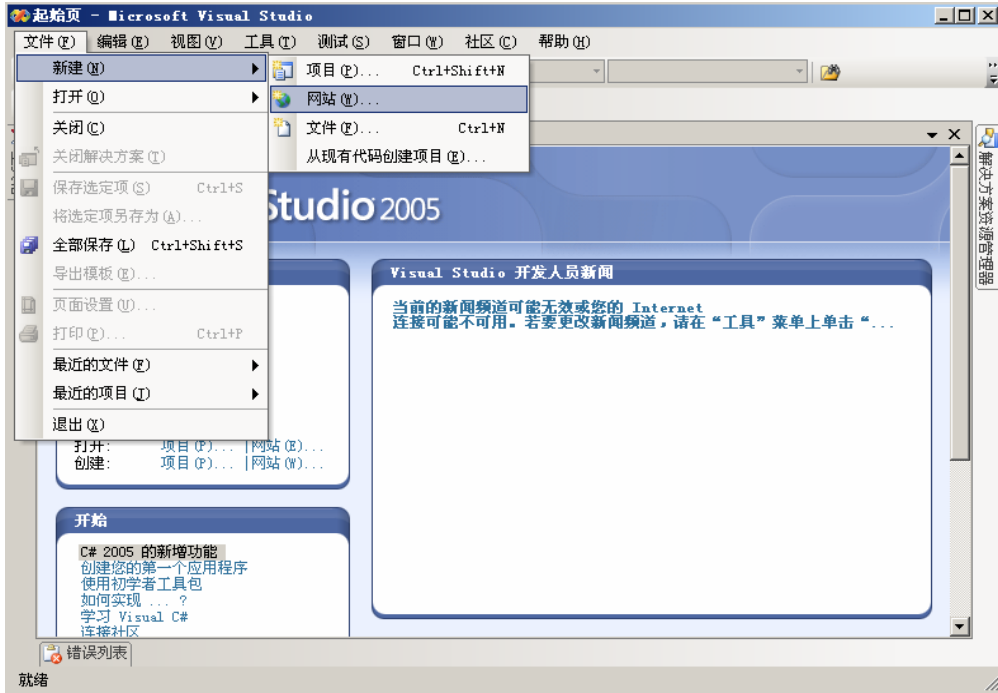


图 2-1 新建网站

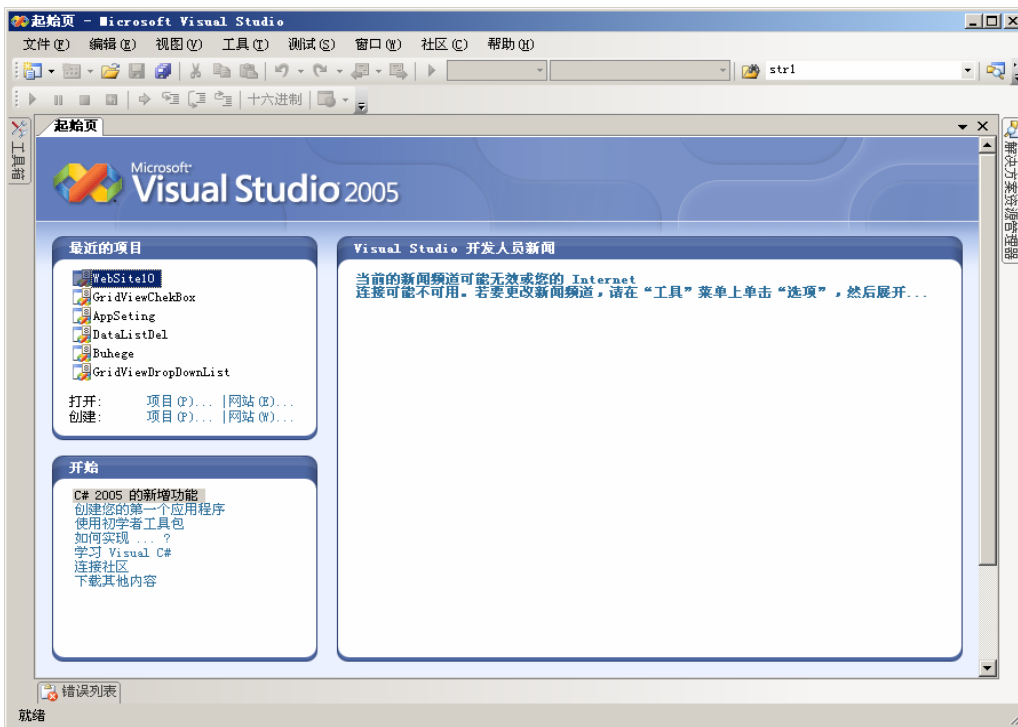


图 2-2 在最近项目里创建网站

3. 在跳出的“新建网站”对话框中“Visual studio 已安装的模板”中选择“ASP.NET 网站”，在位置栏里，修改网站的名称和保存位置（如图 2-3）。

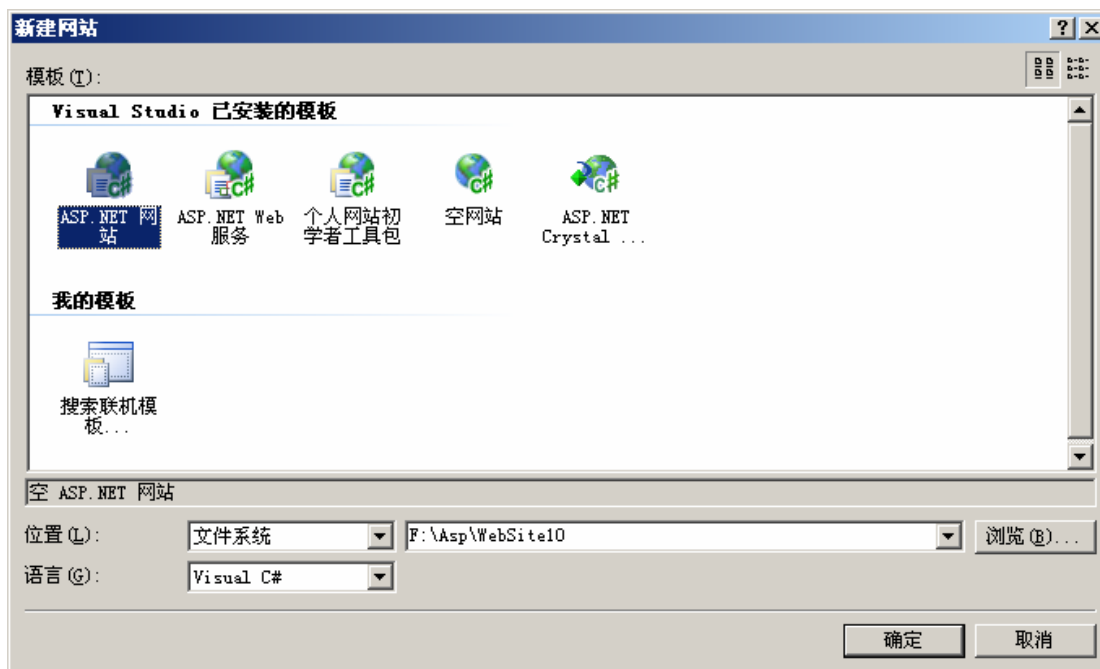


图 2-3 选择新建网站模板页面

4. 单击“确定”，站点创建成功，VS05 编辑域显示 Default.aspx 的源码（如图 2-4）。

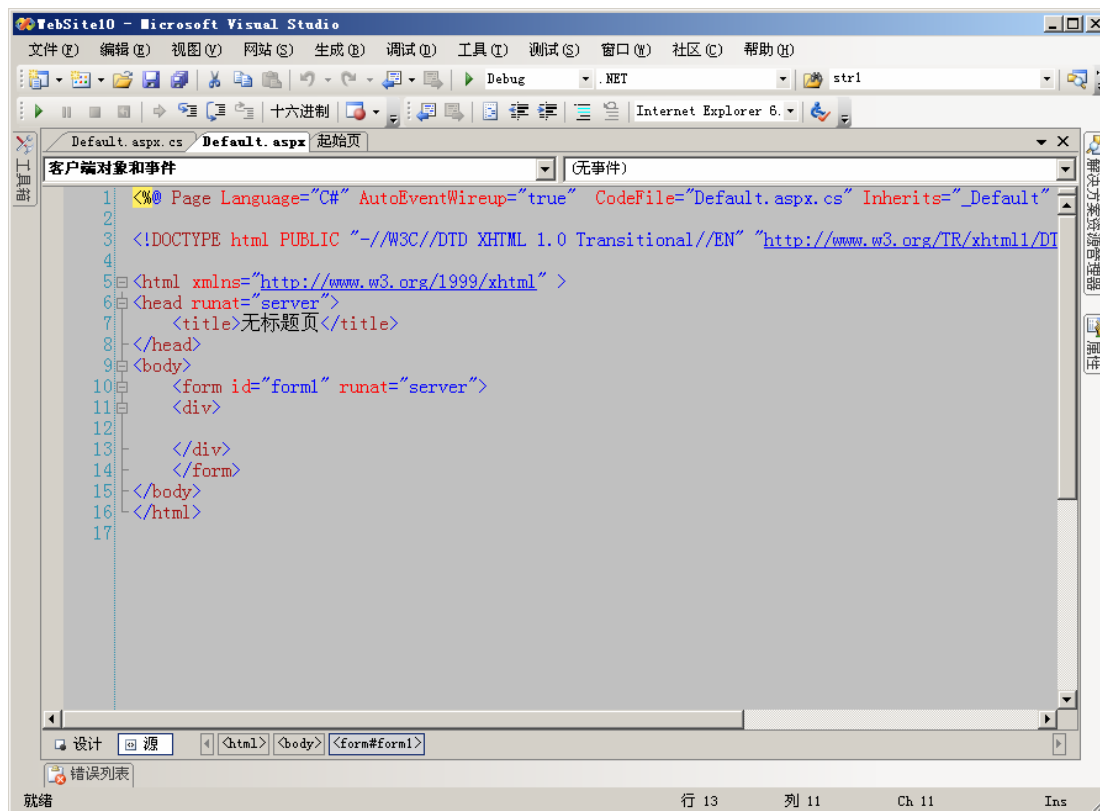


图 2-4 Default.aspx 首页的源码

5. 点击右边的“解决方案资源管理器”→网站目录（F:\Asp\WebSite10）→Default.aspx→Default.aspx.cs，Default.aspx.cs 就是我们所说的“后台代码”（如图 2-5）。

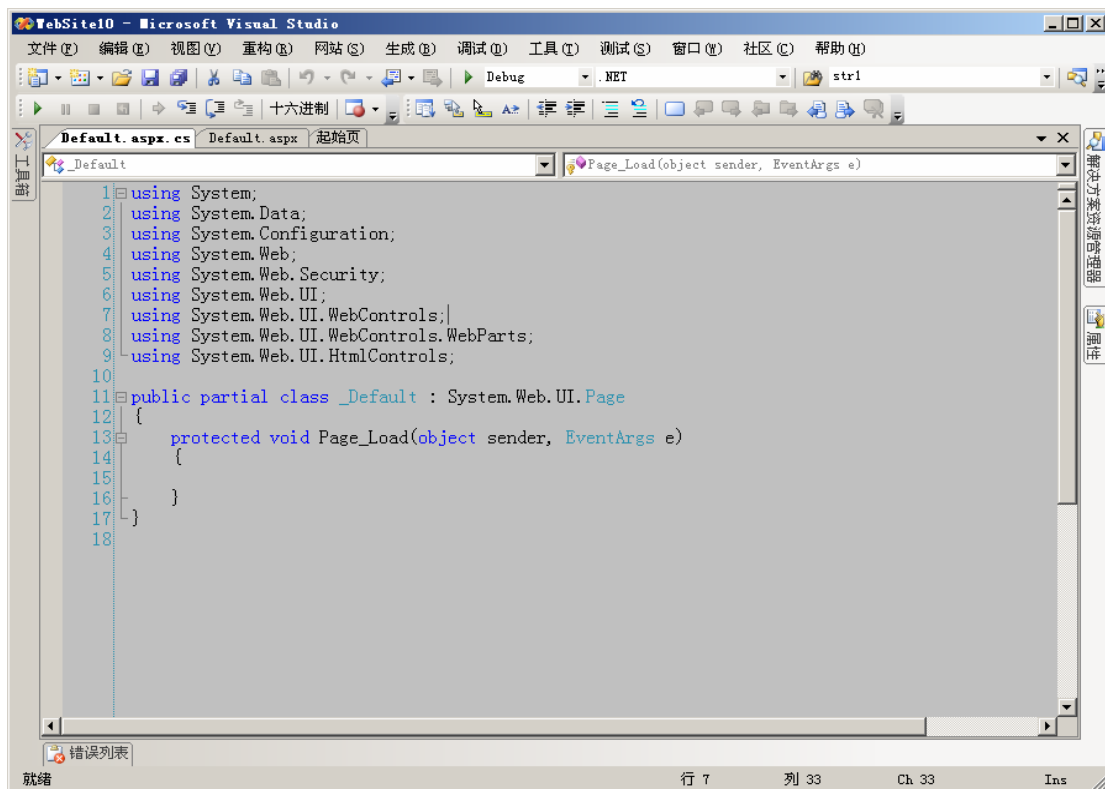


图 2-5 Default.aspx.cs 代码页面

6. 在 VS05 中，可以很方便的新建不同类型的文件，新选中工程→右键→添加新项，流程如图 2-6、图 2-7。

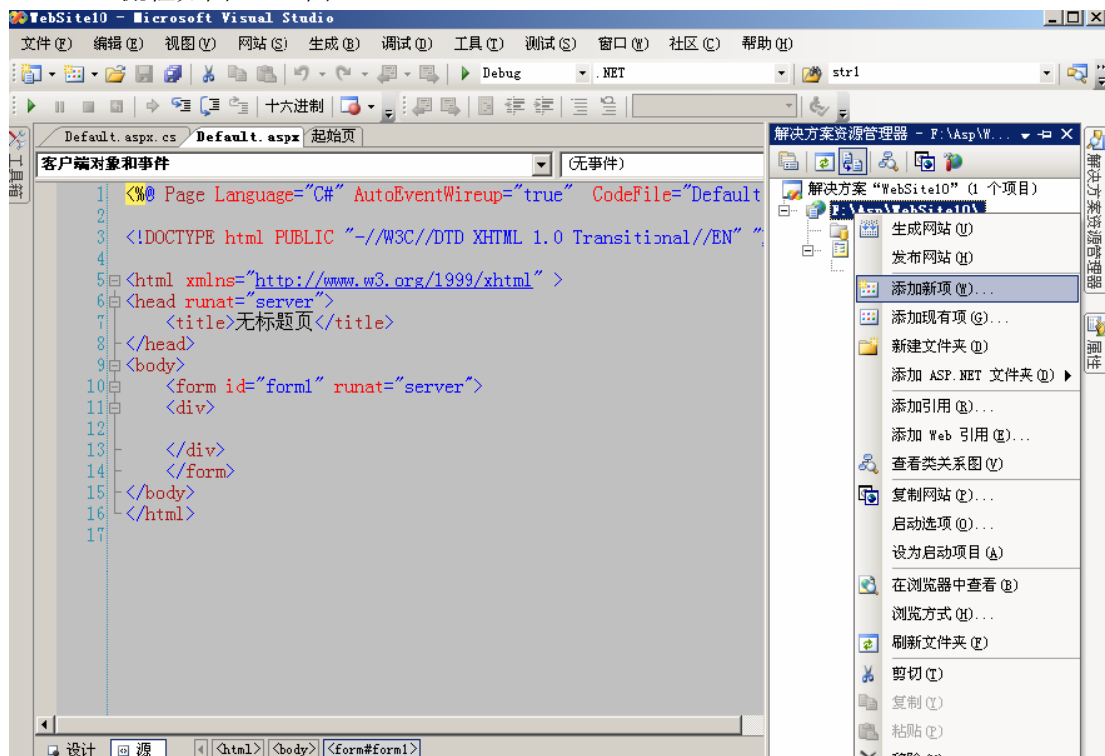


图 2-6 添加新项

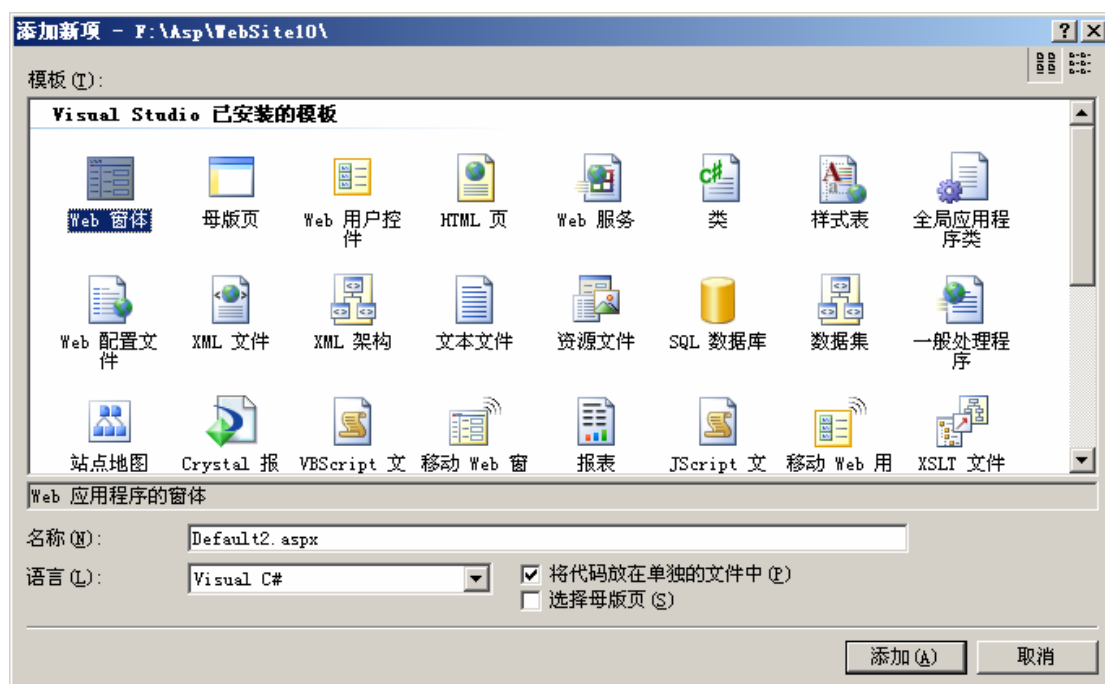


图 2-7 添加新项目模板选择页面

可以根据需要选择文件类型，主要的文件是 Web 窗体(.aspx 文件)、HTML 页(.htm 文件)、Web 配置文件(.config 文件)。点击文件类型后，在名称栏修改文件名，再点击“添加”按钮，文件建立完成。

7. 创建文件后就可以在编辑区打开该文件进行编程。例如，在 Default.aspx 中加入了如下代码，完成网站首页上的简单编码（如图 2-8）。

```

<head runat="server">
    <title>.Net Website</title>
</head>
<body>
    <form id="form1" runat="server">
        <div align=center>
            You are welcome!
        </div>
    </form>
</body>
</html>


```

图 2-8 Default.aspx 中添加代码

所有文件的编程完成后，站点就搭建好了。如图，一个简单的 ASP.NET 网站建好了。现在打开管理工具→Internet 信息服务 (IIS) 管理器→本地计算机→网站→默认网站，就可以看到所建网站在 IIS 下的虚拟目录。

8. 站点搭建完成后，要设置站点的起始页。如果没有设置站点的起始页面，网站运行时将使用系统默认的 default.aspx 页面作为起始页。如果站点没有设置起始页，又没有 default.aspx 页面，网站就会出现“找不到资源”的错误。

如果网站出现“找不到资源”的错误，则可参照以上步骤建立一个新的页面，比如：index.htm 页面。建好后在“解决方案资源管理器”中找到 index.htm 页面，右键点击设为起始页选项。这样站点就设置好了起始页，页面一加载时首先访问此页面。

9. 在 VS05 中运行新建的网站，即点击菜单条上的绿色小三角  或者按快捷键 F5，VS05 开始运行。

如果是第一次运行项目就会跳出一个“未启动调试”的对话框，生成一个新的 Web.config 文件，这是一个配置文件。每个项目都有一个配置文件，点击确定后创建网站的 web.config 文件，然后在“解决方案资源管理器”中可看到新生成的配置文件（如图 2-9）。

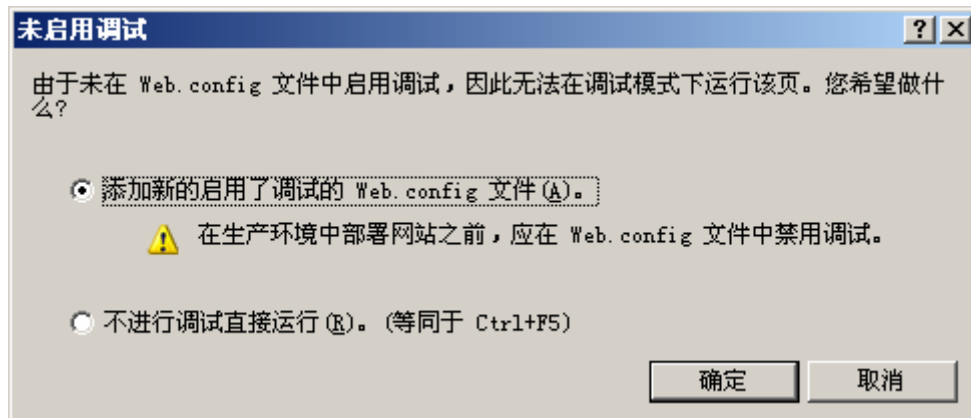


图 2-9 “未启动调试”调试页面

点击“确定”后，跳出运行界面（如图 2-10）。

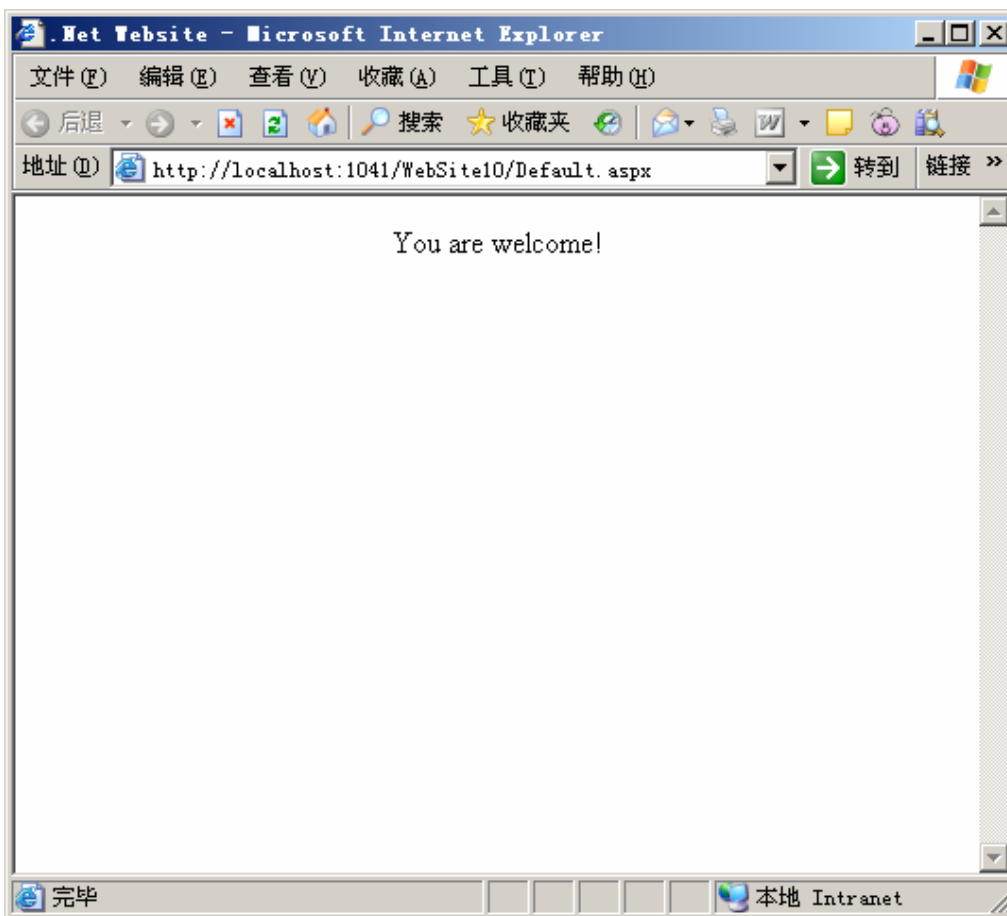


图 2-10 运行页面

网页上的地址栏就是站点的网址。网站在 WEB 服务器上部署后，其他人就可以通过网络在浏览器中输入网址，访问该站点。如果网站中有错，运行就会中止，开始调试。如何调试将在下一节程序调试方法中讲述。

在 VS05 中根据上述流程，一个简单的网站搭建完毕。如果需要详细了解，请参看 VS05 的 msdn 帮助。

3 JAVA 开发环境介绍

IMS 二次开发的 JAVA 开发环境是基于开发工具 Eclipse 3.2、开发语言 JAVA 和 JSP 技术的。这些工具和技术是当下 J2EE 开发的主流技术，具有各自的优点，有助于用户更好地进行 MAPGIS7-IMS 行业版的二次开发。在开发之前需要配置 JAVA 开发环境，设置好环境变量。JAVA 环境的配置相对于 .NET 开发环境要复杂些，但行业版提供了一个 JAVA 环境配置工具 (RegApp.exe)，通过该工具可将复杂的手动配置简单化。使用 JAVA 环境配置工具对 JAVA 环境的安装配置过程将在其它章节中详细讲述。配置好环境后，就能利用功能强大的 JAVA 开发环境开发 WEB 项目。

3.1 JAVA 开发环境

JAVA 开发环境配置如下：

1. 操作系统： Windows2003 Server（推荐）
2. 浏览器： IE 6.0 及以上版本
3. 环境支持：jdk1.5 及以上、tomcat5.0 及以上（JDK 版本要支持 Tomcat 版本）
4. 数据库：ORACLE 9i 或 SQL Server 2000（数据驱动程序 JDBC 2000）

3.1.1 行业版的开发环境

行业版的 JAVA 开发环境配置如下：

1. 按上述配置 JAVA 开发环境，安装好 JDK1.5 和 tomcat5.0 后，正确配置好环境变量。
2. 开发语言：JAVA 和 JavaScript。
3. 二次开发工具：Eclipse 3.2、MyEclipse5.0、MyEclipse6.0、其他插件。如果不习惯英文版的 Eclipse，可以用相应版本的汉化包汉化，便于开发。
4. 与 .NET 开发环境一样，准备好地图数据和相关的点位信息数据，并参照行业版的配置文档配置好 GIS 服务器和相关数据。示例中自带地图数据和点位信息数据，点位信息数据以 SQL Server 数据库文件形式存放。如果使用自带的点位信息数据时，要将 CityPosInfo_Data.MDF 附加到本地 SQL Server 2000 数据库里。
5. 与 .NET 开发环境一样，站点开发过程中须保持 GIS 服务器（GridMapSvr 和 MapSpecialSvr）呈启动状态。

3.1.2 行业版的开发语言

在 JAVA 的开发环境中，行业版开发的语言为 JAVA 语言。

JAVA 语言由 Sun 公司推出，是一个支持网络计算的面向对象程序设计语言。JAVA 语言吸收了 Smalltalk 语言和 C++语言的优点，并增加了其它特性，如支持并发程序设计、网络通信、和多媒体数据控制等。

3.1.3 行业版所用的文件类型

MAPGIS7-IMS 行业版中用到的文件类型和重要文件如下表所示：

表 3-1 JAVA 开发环境文件类型

文件扩展名	用途及说明
.java	Java 文件。
.class	Java 文件编译后的文件。
.jar	Java 归档文件，存放各种 java 文件的压缩包。
.htm	静态网页文件。
.css	样式文件,存放网站样式的样式表。
.js	Javascript 编写的脚本文件。
.jsp	内含 java 代码的文件，网站后台处理文件。
web.xml	网站配置文件。
.hbm.java 与 .hbm.xml	使用 Hibernate 的数据库表映射文件。
hibernate.cfg.xml	使用 Hibernate 操纵数据库的文件，即数据库连接。

3.2 开发软件---Eclipse

Eclipse 是一种通用工具平台——普遍适用的开放式扩展 IDE。它提供了功能丰富的开发环境，众多插件能够无缝集成到 Eclipse 平台中的工具。Eclipse 的体系结构如下图：

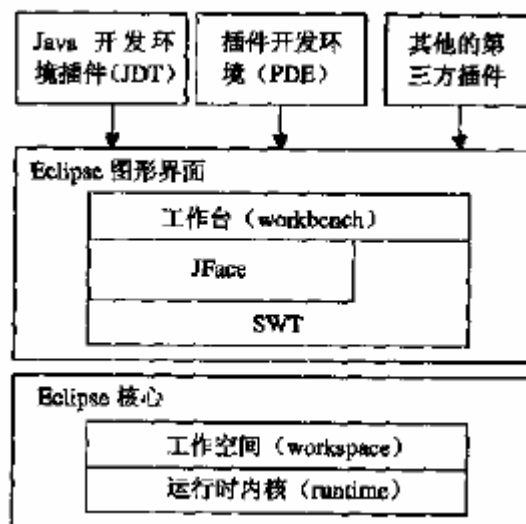


图 3-1 Eclipse 的体系结构图

Eclipse 的设计思想是：一切皆为插件。Eclipse 的核心是非常小的，其他所有的功能都以插件的形式附加到这个 Eclipse 核心之上。这样的功能插件包括它的图形 API(称为 SWT/JFace)、Java 开发环境插件(简称 JDT)、插件开发环境(简称 PDE)等。Eclipse 对插件的协同工作提供了良好的支持，不仅安装简单，而且还可以无缝结合。

Eclipse 是开放源码的。对于程序员，Eclipse 的开放源码让他们阅读最顶级的 Java 源码，学习最优秀的 Java 技术。

下面就开始介绍如何在 Eclipse 里新建一个 WEB 站点，流程如下：

1. 打开 Eclipse，其开始界面如图 3-2。

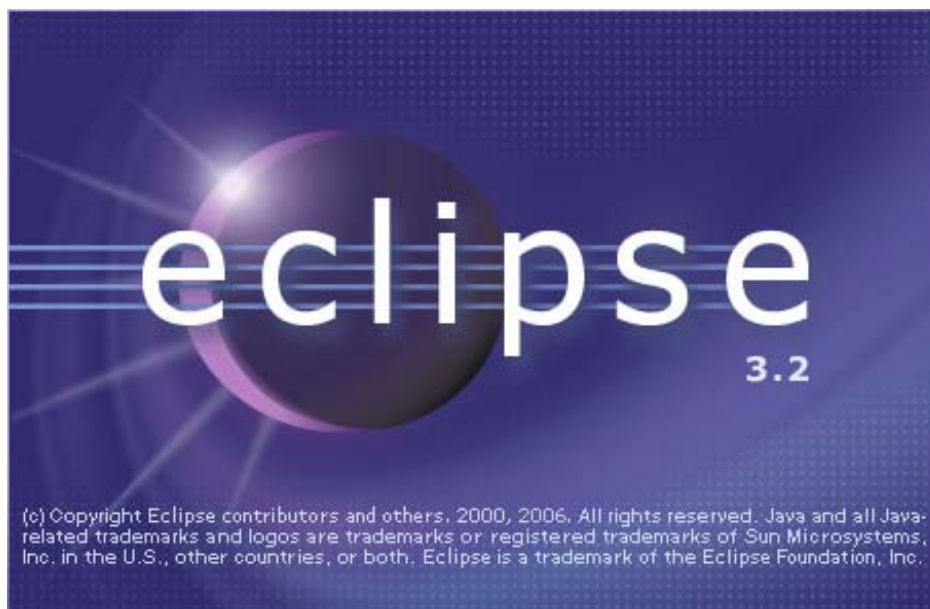


图 3-2 Eclipse 启动时界面

2. 在初次启动 Eclipse 时会跳出一个对话框，提示选择工作空间，用来存 Eclipse 项目的文件。如图 3-3，选择路径后单击“确定”就可以启动 Eclipse。如果想下次启动时不出现该窗口，则勾选对话框最下面的“将此值用作缺省值并且不再询问”选项。

启动后的界面如图 3-4。

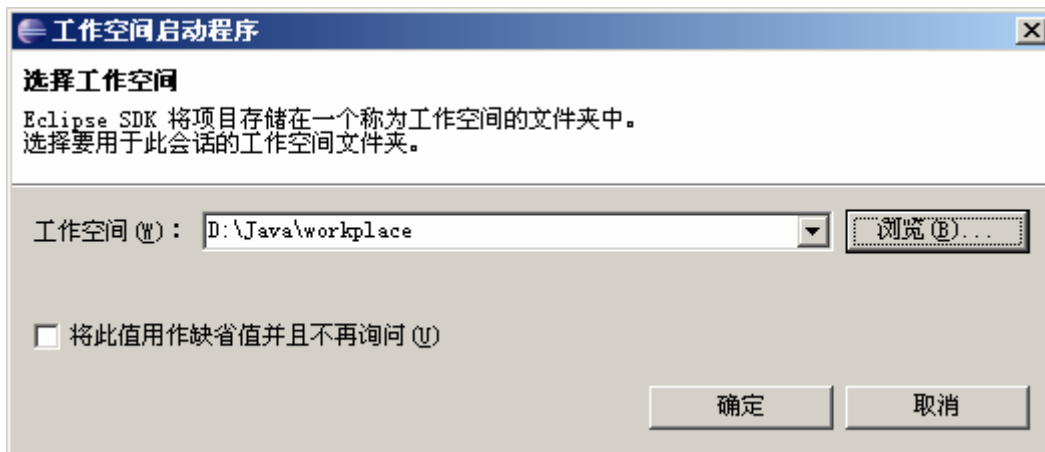


图 3-3 选择工作空间页面

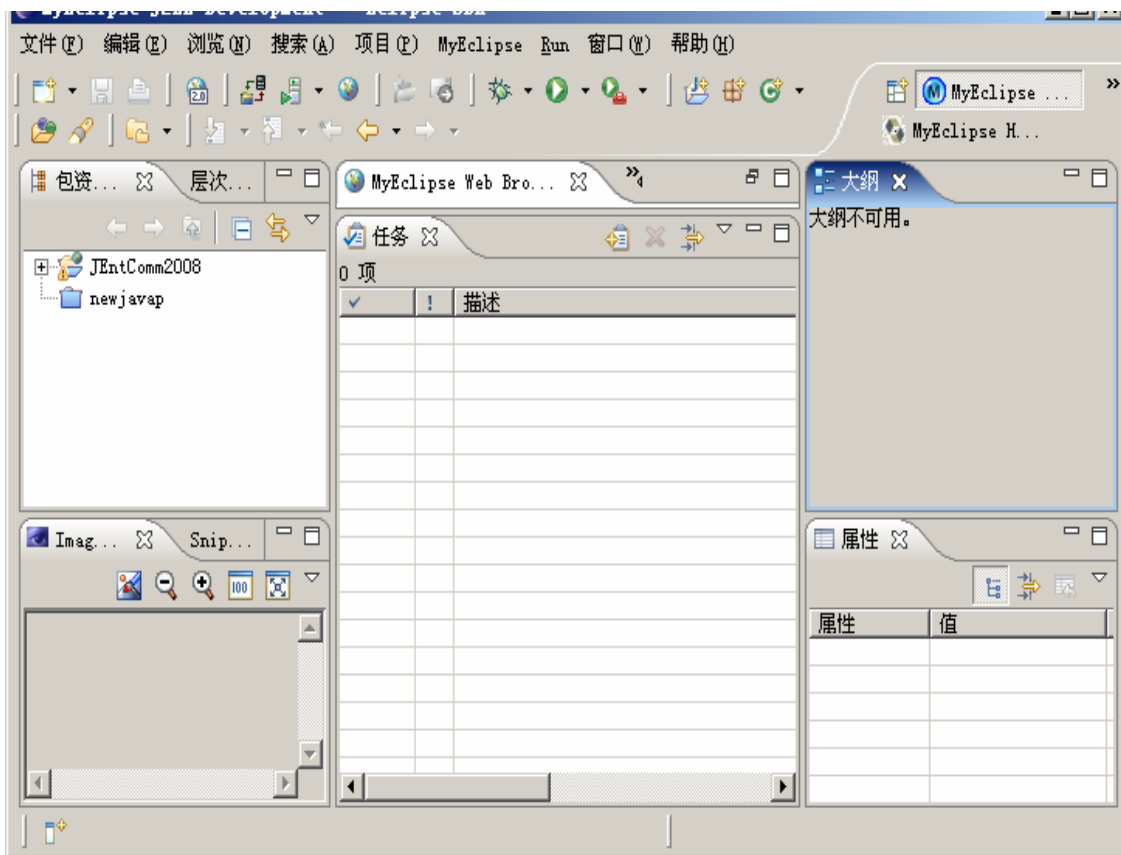


图 3-4 Eclipse 工作环境

3. 新建网站项目：选择文件→新建→项目，跳出如图 3-5 所示对话框。
4. 点击“向导”下的 MyEclipse，选择 J2EEProjects 下的 WebProject，如图 3-6。



图 3-5 Eclipse 中新建项目对话框

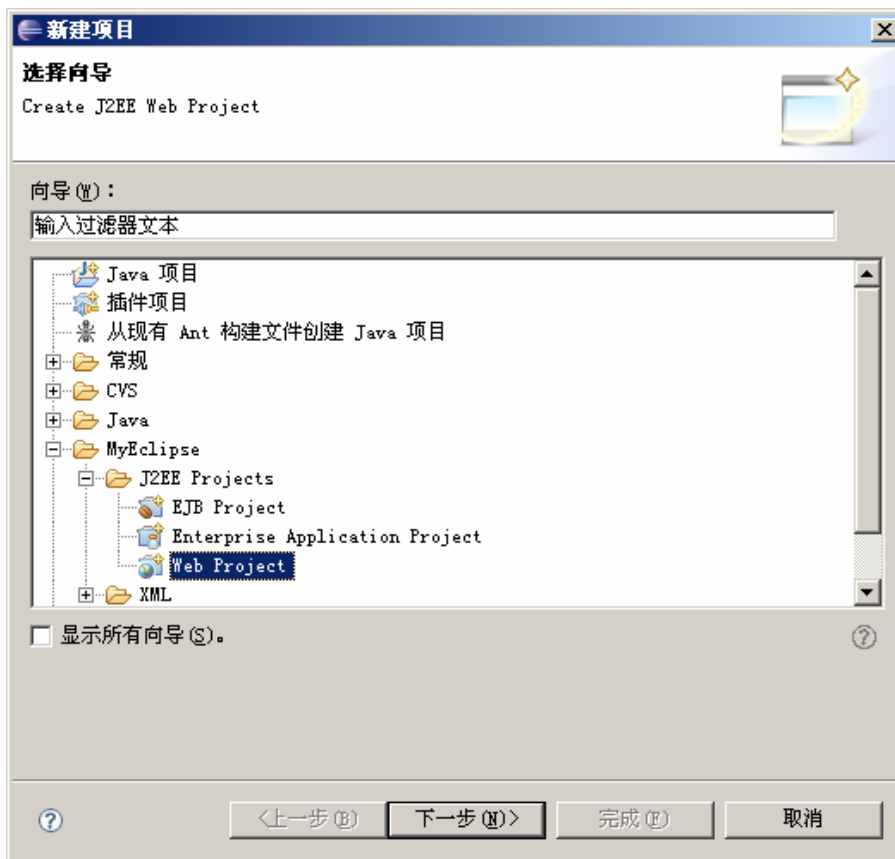


图 3-6 选择 Web Project 新建 Web 工程

5. 点击“下一步”，在跳出的对话框中的 Project Name 栏内输入项目名，例如填入 newjavap，如图 3-7。Location 栏指项目存放的目录，默认存放在选择的工作空间中，一般不需要更改。如要特殊情况下需要更改时，请把 Use default location 的钩去掉，点击 Directory 栏的 Browse 按钮，选择路径。

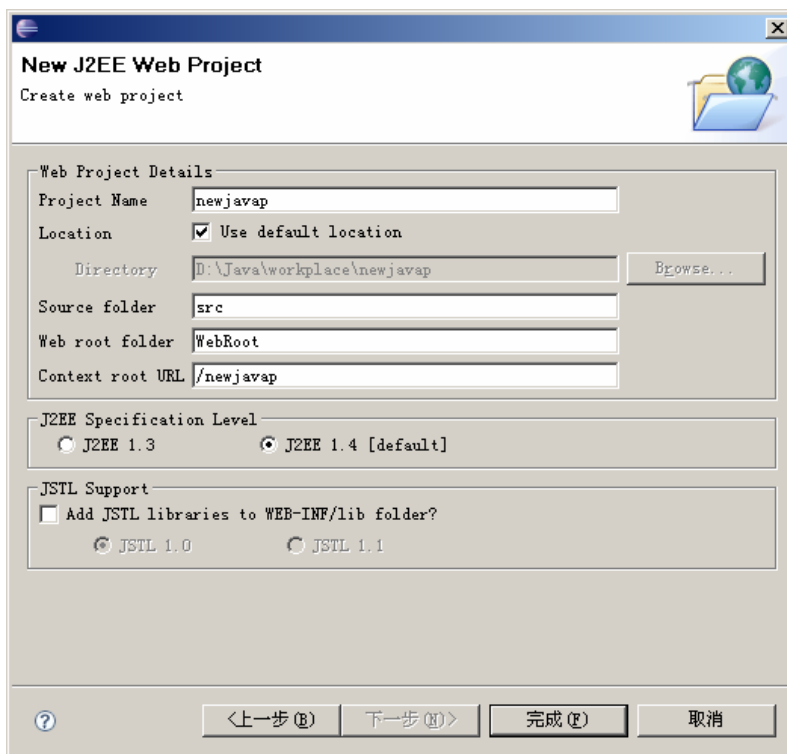


图 3-7 输入新工程名

6. 点击“完成”，新的 WEB 站点创建成功。完成后的界面如图 3-8。

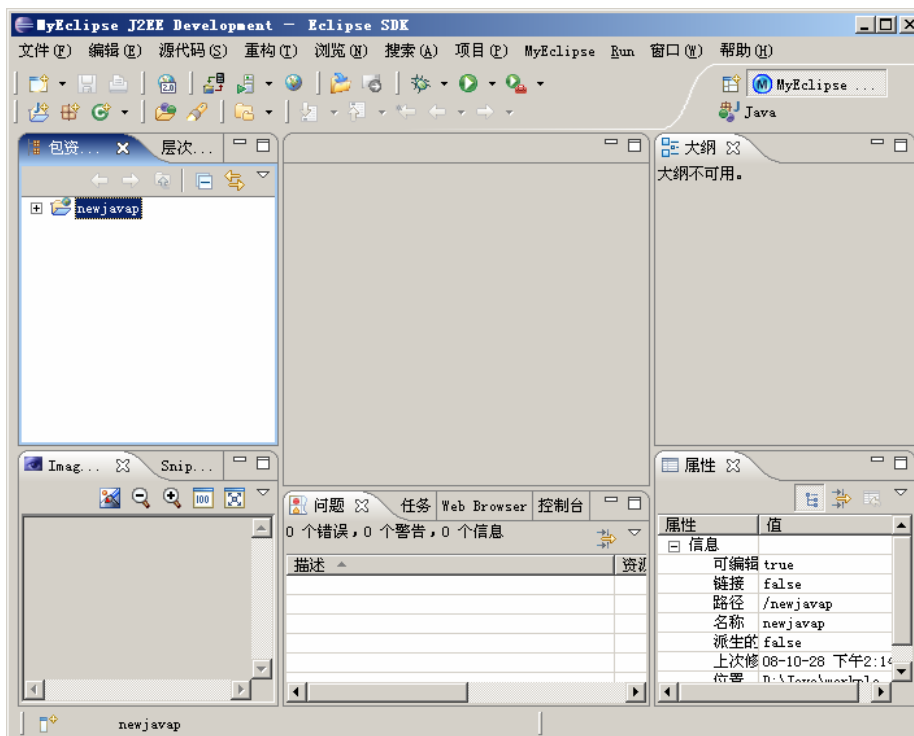


图 3-8 新建工程后的工作页面

7. 此时在包资源管理中右键点击站点，选择新建→文件，在弹出的对话框中根据站点的需要新建文件（如图 3-9）；也可以选择新建→其他，在弹出的对话框中选择要新建的文件模板。

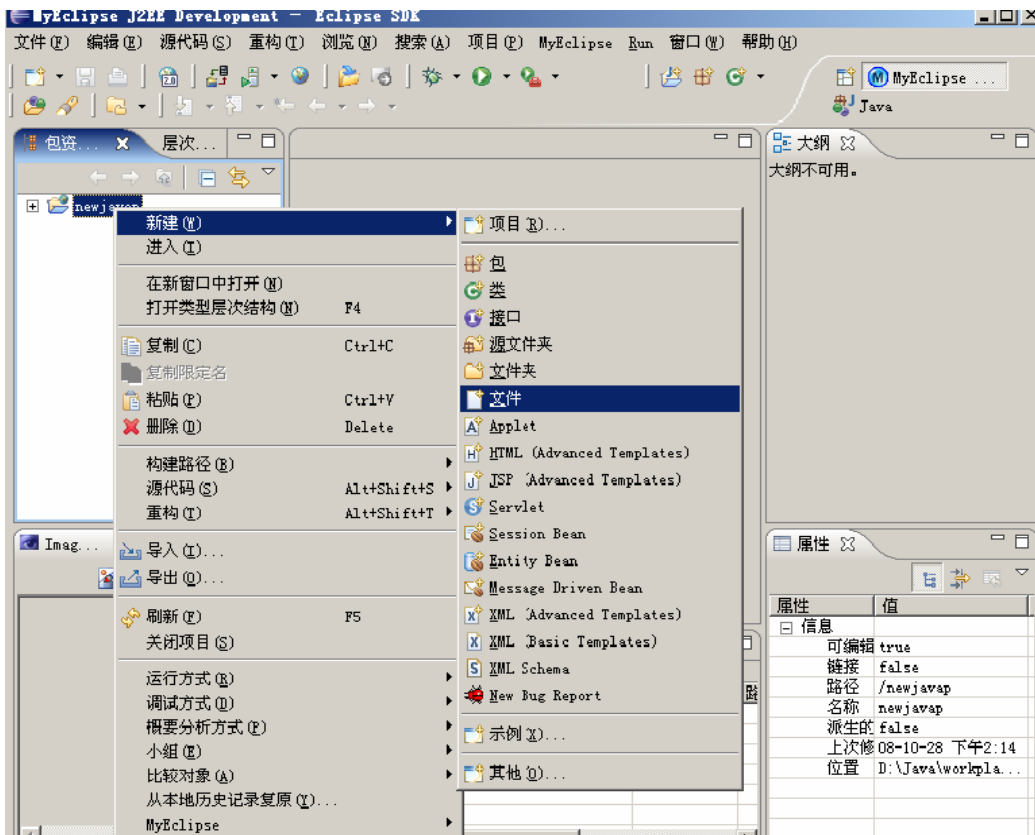


图 3-9 新建文件页面

在如下的对话框中，选择站点目录（WebRoot），将新建的文件保存到该站点目录下。

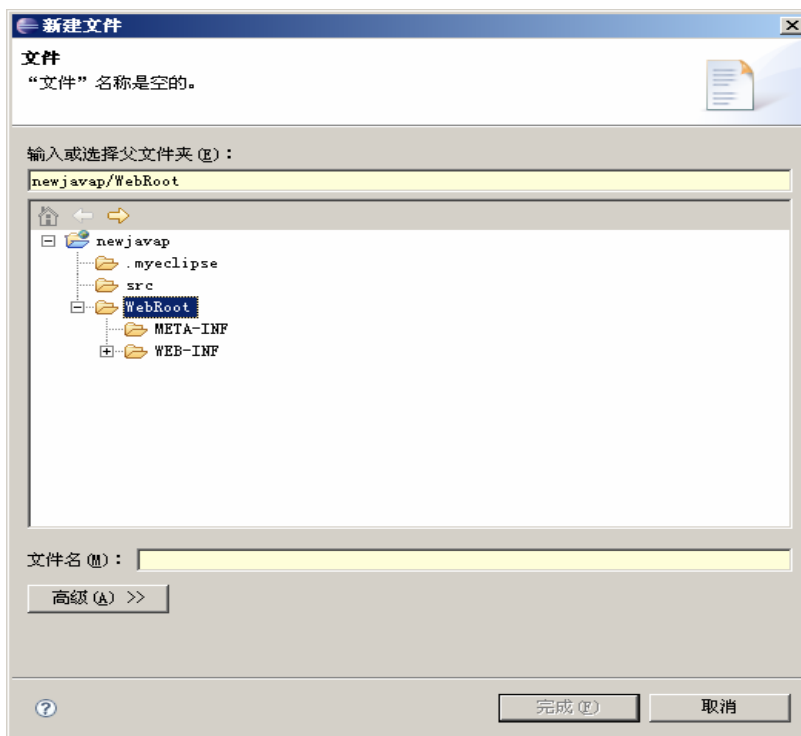


图 3-10 选择父文件夹对话框

输入文件名和后缀名，根据不同的后缀名建立不同的文件类型。网站启动时需要一个起始页，可以是.htm 文件或者是.jsp 文件。在此可新建一个 default.htm 文件作为站点的起始页（如图 3-11）。

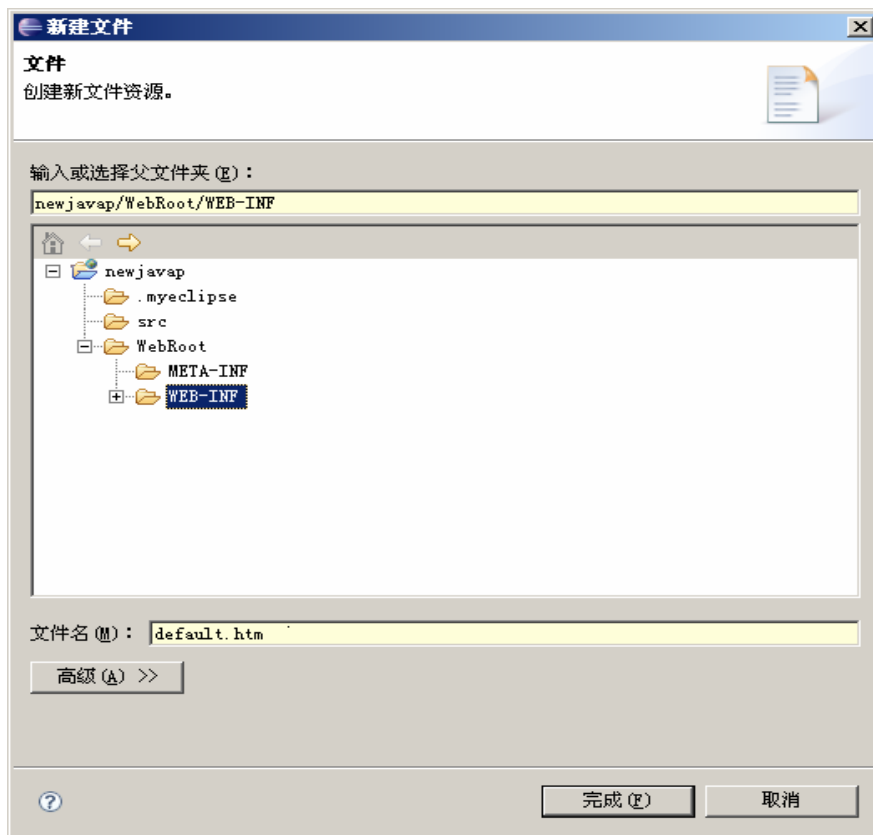


图 3-11 输入文件名

8. 文件建好后，可以在编辑区打开文件进行编程，实现所需要的功能。例如，搭建一个简单的示例，即在起始页 default.htm 中写入如下代码：


```
<html>
<head>
<title>This My First Website</title>
</head>
<body>
You Are WelCome !
</body>
</html>
```

图 3-12 default.htm 中添加代码

9. 修改配置文件 web.xml，在该文件中为站点进行相关的配置。如图，给站点设置起始页 default.htm。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.4"
  xmlns="http://java.sun.com/xml/ns/j2ee"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
  http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd">
  <welcome-file-list>
    <welcome-file>default.htm</welcome-file>
  </welcome-file-list>
</web-app>
```

图 3-13 配置 web.xml

10. 站点搭建完成后，需要进行站点部署。点击  按钮，弹出如下图 3-14 所示窗口，在“Project”中选择项目 newjavap；点击“Add”按钮，出现下图 3-15 界面，在“Server”中选择 tomcat5，在“Deploy type”中选择 Exploded Archive，点击“完成”按钮；再点击“确定”按钮即可（如图 3-16）。

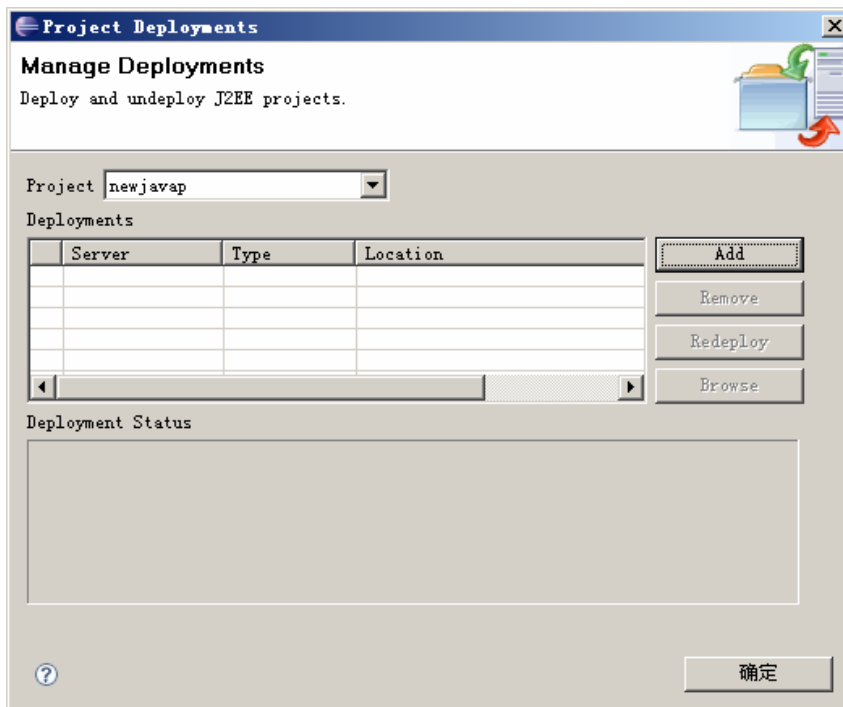


图 3-14 部署步骤一

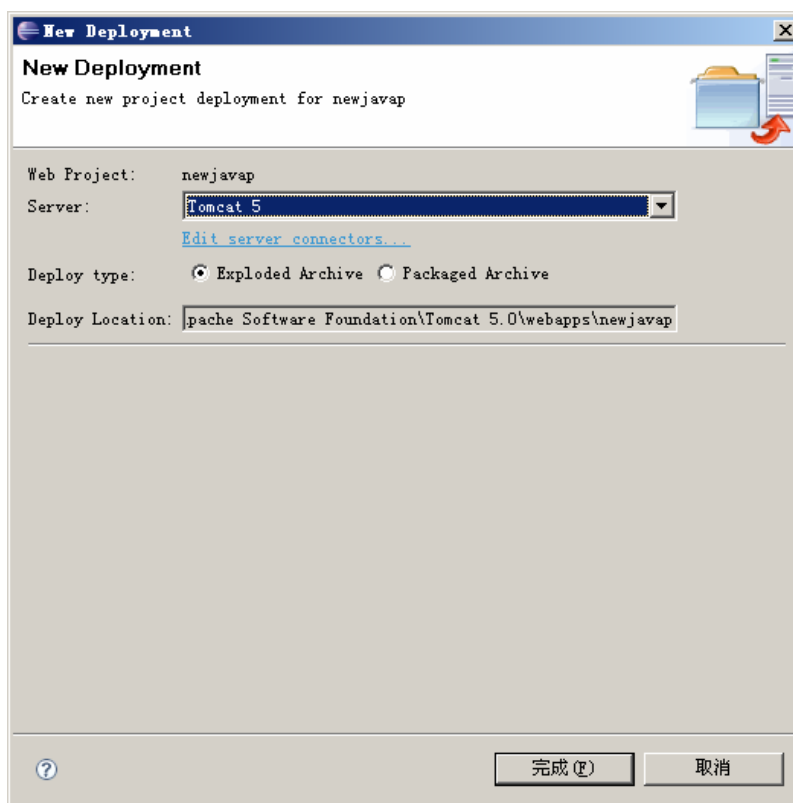


图 3-15 部署步骤二

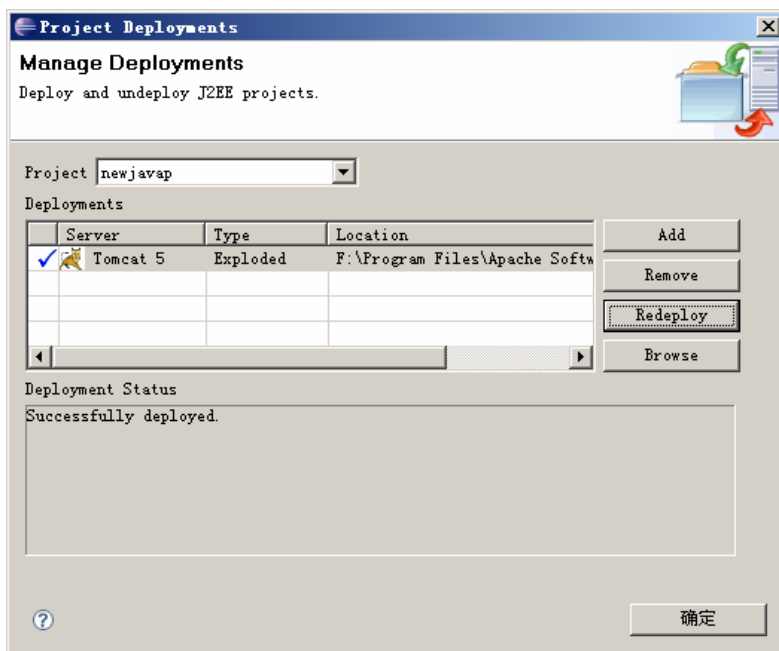




图 3-16 部署步骤三

注：部署站点的同时，Tomcat 服务器自动为 WEB 站点设置好虚拟目录，默认情况下虚拟目录名与项目名相同。

11. 站点部署后，需要运行站点。先启动Tomcat5.5，即点击的向下三角符号启动(如图 3-17)，然后点击启动Eclipse的内置浏览器，在地址栏输入http://localhost:8080/（如图 3-18），再在虚拟目录中选择站点运行，运行后的界面如图 3-19。站点运行过程中可能需要调试程序，具体的调试方法将在后面的“程序调试方法”章节中讲述。

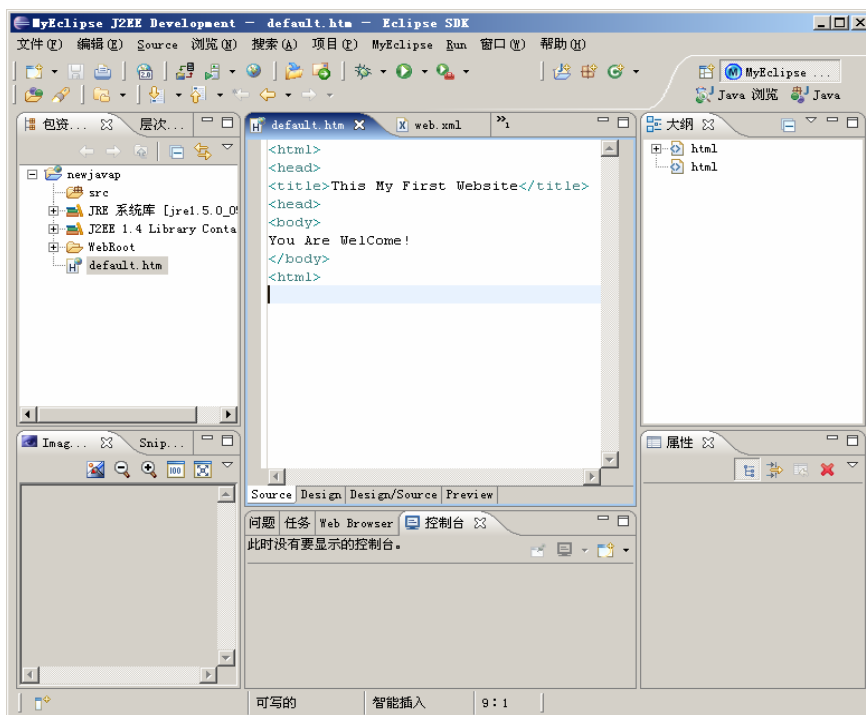


图 3-17 启动 Tomcat 服务器

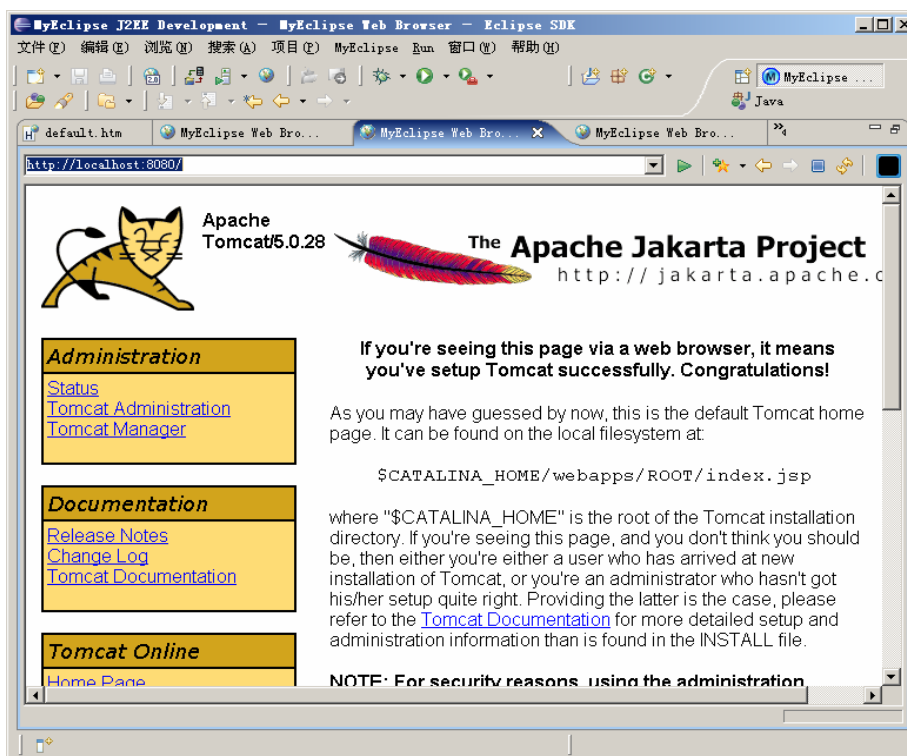


图 3-18 打开 Tomcat 首页

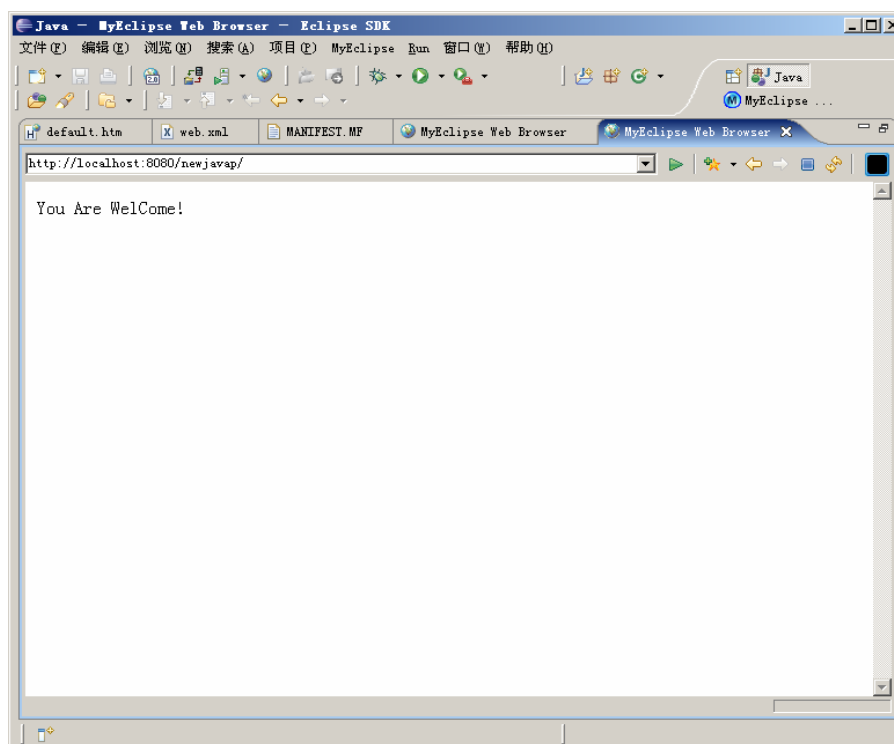


图 3-19 运行网站页面

注意：如果需要修改程序后运行站点，最好重启 Tomcat 服务器，再点击内置浏览器浏览。不然运行结果可能还是修改前的结果。

在 Eclipse 里新建一个 WEB 站点的过程如上所述，如果需要了解更详细的信息，可以参考 Eclipse 的帮助文件。

4 行业版二次开发技术

4.1 HTML 语言

HTML 语言是 Hypertext Markup Language(超文本标记语言)的缩写，是一种通用的描述文档结构的语言。它通过在用户文档中加入特定的控制字符或命令，使文档能够按照用户期望的格式输出。HTML 语言主要用于客户端的页面设计。

二次开发中 HTML 常用标签，如表 4-1 所示：

表 4-1 HTML 中常用标签

标签名	标签
字体标签	
图片标签	
超级链接标签	<A>
有列表标签	
表格标签	<TABLE><TR><TD><TH>
块级元素	<DIV>
网页框架标签	<FRAME><IFRAME>
样式标签	<STYLE>
对齐标签	<ALIGN><VALIGN>

各标签的主要属性如表 4-2 所示，标签属性示例如图 4-1 所示：

表 4-2 各标签的主要属性

标签名	属性
FON 标签	size (大小)、color (颜色)
IMG 标签	src (图像在网上的地址) alt (在浏览器尚未完全读入图像时，在图像位置显示的文字；也是图像显示以后，当鼠标放在图片上时所显示的文字) 等。
TABLE 标签	<tr> ...</tr>- 定义表行
	<th> ...</th>- 定义表头
	<td> ...</td>- 定义表元
	Colspan: 跨多列的表元
FRAME 标签	Frameborder: 设定框架的边框, 其值只有 0 和 1 , 0 就是不要边框 1 就是要显示边框。边框是无法调整粗细的。
	Scrolling: 设定是否要显示滚动条, YES 是要显示滚动条, NO 是无论如何都不要显示, AUTO 是视情况显示。
ALIGN 标签	RIGHT (向右对齐)、LEFT (向左对齐)、CENTER (居中对齐)
VALIGN 标签	TOP (向上对齐)、MIDDLE (向中对齐)、BOTTOM (向下对齐)
超级链接标签	href:超链接地址
style 标签	width (宽)、height (高)

```

<table style="width: 795px; height: 535px">
  <tr>
    <td colspan="3">
      <ul>
        <li><a href="#"></li>
        <li></li>
      </ul>
    </td>
  </tr>
  <tr>
    <td colspan="3">
      <div id="mmPanel"><IFRAME id="toolFrame" style="BORDER: 0px; WIDTH: 100%;HEIGHT: 100%"
src="menu.htm" frameborder="0" scrolling="auto"></IFRAME></div>
      <div id="rsPanel" style="display:none"><IFRAME id="rsFrame" style="BORDER: 0px; WIDTH:
100%;HEIGHT: 100%" src="" frameborder="0" scrolling="auto"></IFRAME></div>
    </td>
  </tr>
</table>

```

图 4-1 标签属性示例

4.2 CSS 样式表

CSS 是英语 Cascading Style Sheets (层叠样式表) 的缩写, 它是一种用来表现 HTML 或 XML 等文件式样的计算机语言, 是用于 (增强) 控制网页样式并允许将样式信息与网页内容分离的一种标记性语言。

CSS 可以使页面的字体变得更漂亮, 更容易编排, 使页面真正赏心悦目; 你可以轻松地控制页面的布局, 可以将站点上所有的网页风格都使用一个 CSS 文件进行控制, 只要修改这个 CSS 文件中相应的行, 那么整个站点的所有页面都会随之发生变动。

将 CSS 文件加入到网页中主要有三种方法: 链入外部样式表文件 (link 对象)、定义内部样式块对象 (<STYLE>与</STYLE>块对象) 和内联定义 (标记内使用对象的 style 属性定义)。

在行业版的示例站点中使用了链入外部样式表的方法, 使用 HTML 的 link 对象把建立好的样式表文件 enterprise.css 直接链入到示例站点的页面中。示例如下:

```
<link rel="stylesheet" type="text/css" href="css/enterprise.css" />
```

MAPGIS7-IMS 行业版所涉及的样式表文件 (enterprise.css) 中部分 CSS 属性如下表:

表 4-3 CSS 属性

属性名	属性
字体	width、height、weight
边框	border:为“0”时无边框,width、height
鼠标光标样	cursor 为 hand 时是“手指”; 为 crosshair 时是“十字”
边界样式	margin: top、right、bottom、left
边框空白	padding: top、right、bottom、left
定位	position: top、right、bottom、left
层叠顺序	z-index
长宽属性	width、height
文本	text:
背景属性	background:color、image
布局	Layout:overflow: 设置当对象的内容超过其指定高度及宽度时如何管理内容; overflow-x、overflow-y
	float:值指出了对象是否及如何浮动
	clear:该属性的值指出了不允许有浮动对象的边
滚动条	Scrollbars

示例如下:

```
#rightPanel{position:absolute;right:2px;height:350px;width:308px;background:transparent;}+
#rsTitle{height:30px;overflow:hidden;}+
#rsTitle td{border-bottom:1px solid #005198;}+
#rsTitle th{width:80px;text-align:center;border:1px solid #005198;line-height:28px;background:#EDEEFF;
font-weight:normal;cursor:pointer;color:#B0C4DE;}+
#rsTitle .th1{font-weight:bold;border-bottom:1px solid #fff;background-image:
url(..images/bg.png);color:#528DC5;}+
#rsPanel, #mmPanel{overflow:auto;border:1px solid
#005198;border-top-width:0px;height:326px;padding-left:10px;padding-top:4px;}+
#rsPanel .page{text-align:center;margin:10px 0px;}+
#rsPanel select{width:270px;}+
#rsPanel dl{margin-top:10px;}+
#rsPanel dt{line-height:22px;margin-left:10px;}+
#rsPanel dt img{margin-right:5px;}+
#rsPanel td{padding:3px;line-height:18px;}+
#mmPanel p{margin:17px auto 14px;}+
.btnfmt {+
background-image: url(..images/button.gif);+
color: #000;+
border: 0px;+
height: 22px;+
width: 52px;+
font-family: "宋体";+
font-size: 12px;+
padding-top: 2px;+
cursor:pointer;+
}+
```

图 4-2 CSS 示例

4.3 JavaScript 脚本语言

JavaScript 是一种通用的、基于原型的、面向对象的脚本语言，它的设计目标是在不占用很多系统和网络资源的情况下提供一种可以嵌入不同的应用程序的通用代码。它不需要依赖于特定的机器和操作系统，即它是独立于操作平台的。

行业版有十个 JavaScript 文件：ext.js (ext-lp.js), flash.js, function.js, IMSMapConfig.js, IMSMaps.js, IMSMapsExt.js, modules.js, IMSPathSearch.js, prototype.js, spatialAnalyseRlt.js。这十个文件存在于 jslib 目录下，其脚本库里都是重要的功能函数，如：地图的显示宽度、初始化地图、右键菜单、显示查询结果函数、Ajax 请求函数、DOM 对象、XML 函数等很多重要的功能函数，这些都是 IMS 二次开发的所用的脚本函数。用户可以修改 function.js 文件，以实现所需的功能。

JavaScript 的核心对象如表 4-4 所示：

表 4-4 JavaScript 的核心对象

对象	描述
Array	表述数组。
Boolean	表述布尔值。
Date	表述日期。
Function	指定了一个可编译为函数的字符串 JavaScript 代码。
Math	提供了基本的数学常量和函数；如其 PI 属性包含了 π 的值。
Number	表述实数数值。
Object	包含了由所有 JavaScript 对象共享的基本功能。
RegExp	表述了一个正则表达式；同时包含了由所有正则表达式对象的共享的静态属性。
String	表述了一个 JavaScript 字符串。

JavaScript 表单对象如表 4-5 所示：

表 4-5 JavaScript 表单对象

对象	描述
Button	HTML 表单中的一个按钮。
Checkbox	HTML 表单中的一个复选框。
Form	允许用户从像复选框、单选钮、选择列表等这样的表单元素中输入文本做出选择。
Hidden	HTML 表单中的一个不可见文本对象。
Option	Select 对象的一个选项。
Password	HTML 表单中的一个文本域，其中的值以星号(*)显示。
Radio	HTML 表单中的一组单选钮。
Reset	HTML 表单中的一个重置按钮。
Select	HTML 表单中的一个选择列表。
Submit	HTML 表单中的一个提交按钮。
Text	HTML 表单中的一个文本输入域。
Textarea	HTML 表单中的一个多行输入域。

JavaScript 语句如表 4-6 所示：

表 4-6 JavaScript 语句

break	该语句用于结束当前的 while 或 for 循环，并将程序控制权交给循环后面的语句。
comment	作者书写的文字，用于解释脚本的作用。注释将被解释器忽略。
continue	该语句用于中止 while 或 for 循环中一块语句的执行，并且直接执行下一次循环。
delete	删除一个对象的属性或数组中的一个元素。
do...while	一直执行其中包含的语句，直到测试条件为假。内含语句至少被执行一次。
export	允许一个签字的脚本向其它签字或未签字的脚本提供属性、函数和对象。
for	该语句用于创建由三个可选表达式组成的循环，用分号隔开，外面包有圆括号，后面跟着一块将要在循环中执行的语句。
for...in	该语句用于遍历一个对象的所有属性的特定变量。对于每个属性，JavaScript 都将执行特定的语句。
function	该语句用于声明一个带有指定参数的 JavaScript 函数。可以接受的参数包括字符串、数值和对象。
if...else	该语句用于在指定条件为真的情况下执行一组语句。如果条件为假的话，则可执行另外一组语句。
import	允许脚本引入其它签字脚本已经导出的属性、函数和对象。
labeled	提供一个表示符，和 break 或 continue 一起使用可标明程序应该继续执行的流程。
return	该语句用于指定函数的返回值。
switch	允许程序计算一个表达式，并试图将表达式的值与某个 case 标签匹配。
var	该语句用于声明变量，可选赋初值。
while	该语句用于创建一个计算某表达式的循环，如果该表达式为真的话，则持续执行一块语句。
with	该语句用于为一组语句建立缺省的对象。

JavaScript 窗口对象如表 4-7 所示:

表 4-7 JavaScript 窗口对象

对象	描述
Frame	可以在同一屏幕上显示多个互不干涉的可滚动框架的窗口, 每个框架有其自己的 URL 。
History	包含了一组客户已经在窗口中浏览过的 URL 的信息。
Location	包含了当前 URL 的信息。
screen	包含了描述显示屏幕和颜色的属性。
Window	描述了一个浏览器窗口或框架。这是每个 document、Location 和 History 对象组的顶级对象。

JavaScript 示例如图 4-3 所示:

```
function mapResize(x,y){
    x=x?parseInt(x):0;
    y=y?parseInt(y):20;
    var mapPNode = $("zdims").parentNode;
    var w = mapPNode.clientWidth-x;
    var h = mapPNode.clientHeight-y;
    if(typeof maplet=="object")map.resize(w, h);
}
function setOpenMode(activeTool, drawAction){
    map.activeTool = activeTool;
    map.drawAction = drawAction;
}

Event.observe(window, "load", function(){initMap();});

var flashTimes=3, flashInterval=0.3, flashColor="#ffa500";
var userTarget;
var userActive;
var subParam="";

function onDisplayToolClick(con)
{
    var type = con
    switch(type)
    {
        case "reflash":
            map.refresh();
            break;
        case "Condition":
            var url = "VectorPage/SelectByCondition.aspx?rnd="+Math.random();
            renderSelectResult(url);
            break;
    }
}
}
```

图 4-3 JavaScript 示例

4.4 DOM 模型

文档对象模型（Document Object Model, DOM）是表示文档（比如 HTML 和 XML）和访问、操作构成文档的各种元素的应用程序接口（API）。一般的，支持 Javascript 的所有浏览器都支持 DOM。它以树形结构表示 HTML 和 XML 文档，定义了遍历这个树和检查、修改树的节点的方法和属性。

DOM 下，HTML 文档各个节点被视为各种类型的 Node 对象。每个 Node 对象都有自己的属性和方法，利用这些属性和方法可以遍历整个文档树。由于 HTML 文档的复杂性，DOM 定义了 nodeType 来表示节点的类型。表 4-8 列出 Node 常用的几种节点类型：

表 4-8 Node 常用的节点类型

结点名	nodeType 常量
Element	Node.ELEMENT_NODE
Text	Node.TEXT_NODE
Document	Node.DOCUMENT_NODE
Attr	Node.ATTRIBUTE_NODE

二次开发中使用到 Document 方法主要有以下几个方法：

表 4-9 Document 方法

方法	描述
createAttribute()	用指定的名字创建新的 Attr 节点。
createComment()	用指定的字符串创建新的 Comment 节点。
createElement()	用指定的标记名创建新的 Element 节点。
createTextNode()	用指定的文本创建新的 TextNode 节点。
getElementById()	返回文档中具有指定 id 属性的 Element 节点。
getElementsByTagName()	返回文档中具有指定标记名的所有 Element 节点。

二次开发中使用到 Element 的方法主要有：

表 4-10 Element 的方法

方法	描述
getAttribute()	以字符串形式返回指定属性的值。
getAttributeNode()	以 Attr 节点的形式返回指定属性的值。
getElementsByTagName()	返回一个 Node 数组，包含具有指定标记名的所有 Element 节点的子孙节点，
hasAttribute()	如果该元素具有指定名字的属性，则返回 true。
removeAttribute()	从元素中删除指定的属性。
removeAttributeNode()	从元素的属性列表中删除指定的 Attr 节点。
setAttribute()	把指定的属性设置为指定的字符串值，如果该属性不存在则添加一个新属性。
setAttributeNode()	把指定的 Attr 节点添加到该元素的属性列表中。

示例一、Document 方法：

```

//多点定位↵
MdotLocation=function () {↵
    if (locPntInfoArr.length<=0){↵
        map.removeEventListener("zoom",MdotLocation, false);↵
        queryZoomFlag = true↵
        return ;↵
    }↵
    var locPntDiv = document.getElementById("locPntDiv");↵
    locPntDiv.innerHTML = "";↵
    DotInfoShow ();↵
    for (var i=0;i<locPntInfoArr.length;i++) {↵
        var balloon=document.getElementById("balloonImage" + i);↵
        if (balloon == null) {↵
            var imgStr = (locPntInfoArr[i].layer===-1)?"p"+(i+1):"i"+locPntInfoArr[i].layer ;↵
            balloon = document.createElement("IMG") ;↵
            balloon.id = "balloonImage"+i ; ↵
            balloon.src = "images/"+imgStr+".gif" ; ↵
            balloon.style.zIndex="3";↵
            balloon.style.position="absolute";↵
            balloon.display="block";↵
            balloon.style.width = "20px"↵
            balloon.style.height = "20px"↵
            locPntDiv.appendChild(balloon);↵
            ZDEvent.addListener (balloon, "mousedown", map.OnMapMouseDown, false);↵
            // map.addEventListener ("mousedown", balloon.onmousedown, false) ;↵
            if (infoShowFlag){↵
                setValue (balloon, i); ↵
            }↵
        }↵
    }↵
}↵

```

图 4-4 Document 方法示例

示例二、Element的方法:

```

Element.Methods.Simulated = {↵
    hasAttribute: function (element, attribute) {↵
        var t = Element._attributeTranslations;↵
        attribute = t.names[attribute] || attribute;↵
        return $(element).getAttributeNode(attribute).specified;↵
    }↵
};↵

```

图 4-5 Element 的方法示例

4.5 AJAX 技术

Ajax是Asynchronous JavaScript and XML的缩写。Ajax并不是一门新的语言或技术,它实际上是几项技术按一定的方式组合在一起,在共同的协作中发挥各自的作用。

Ajax的一个最大的特点是无需刷新页面便可向服务器传输或读写数据(又称无刷新更新页面),这一特点主要得益于XMLHTTP组件XMLHttpRequest对象。只同服务器进行数据层

面的交换，而不用每次都刷新界面也不用每次将数据处理的工作提交给服务器来做，这样既减轻了服务器的负担又加快了响应速度、缩短了用户等候时间。这也是Ajax技术在IMS平台发挥的最大作用。

传统Web应用模型与Ajax模型的比较。

下图是传统的Web应用模型（采用交互式的方式）：

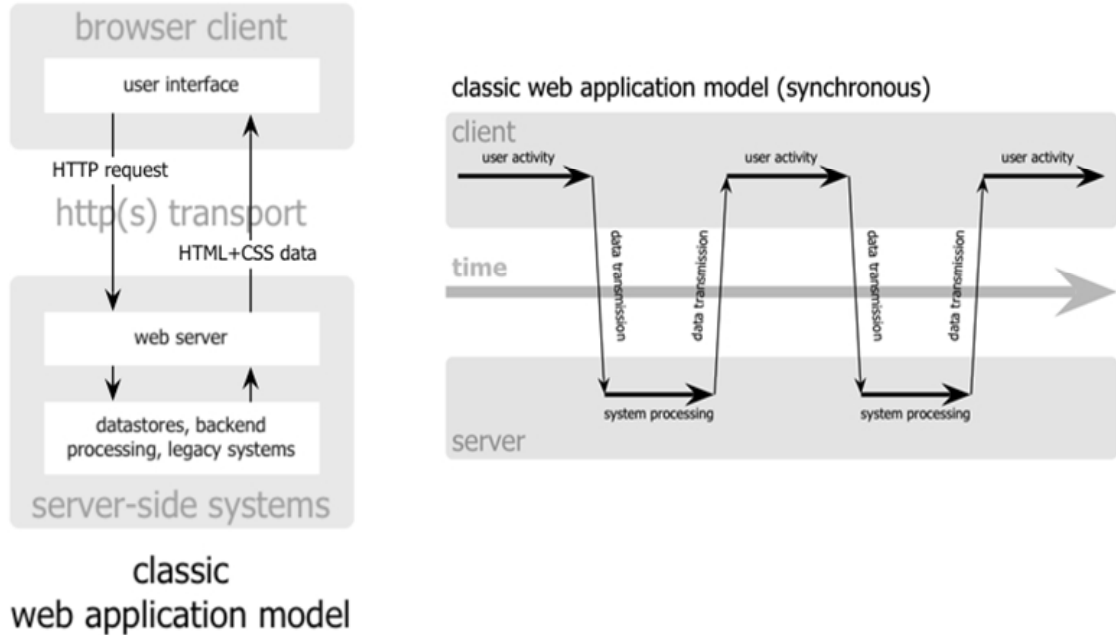


图 4-6 采用交互式的方式的传统的 Web 应用模型

下图是AJAX异构交互方式：

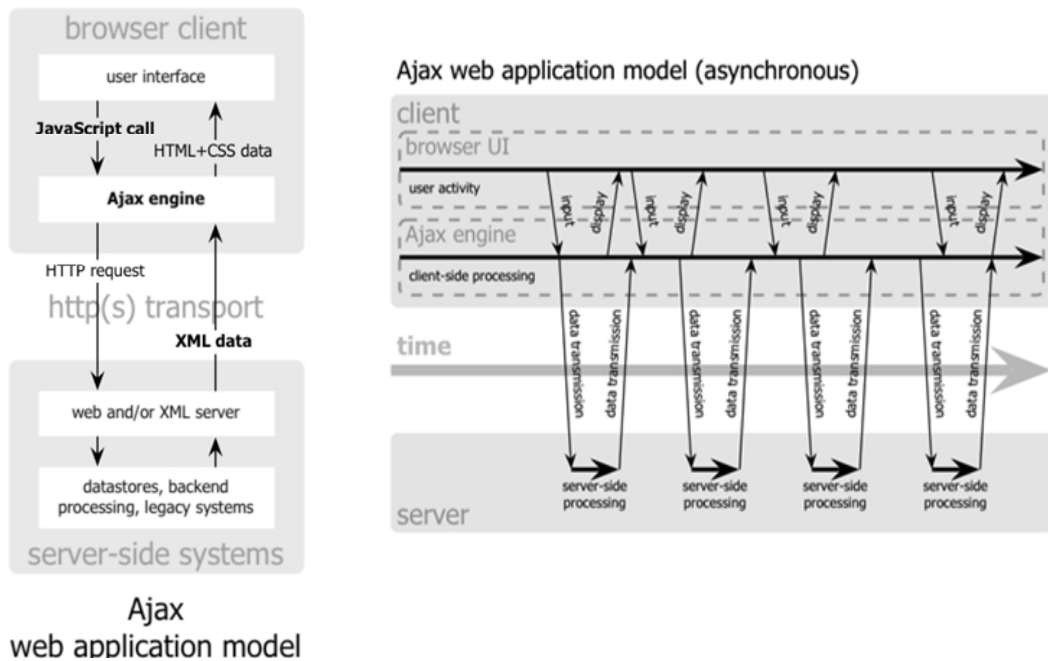


图 4-7 AJAX 异构交互方式

Ajax的核心技术:

1. Javascript: 绑定和处理所有数据
2. HTML+CSS: 页面标准化呈现
3. DOM: 实现动态显示和交互
4. XMLHttpRequest 对象: 进行异步数据读取

在前面章节分别讲述了前三个核心技术, 现在主要讲述XMLHttpRequest对象在二次开发平台中的作用。

在使用XMLHttpRequest对象发送请求和处理响应之前, 必须先用JavaScript创建一个XMLHttpRequest对象。XMLHttpRequest对象的方法如下表:

表 4-11 XMLHttpRequest 对象的方法

方法	描述
abort()	停止当前请求
getAllResponseHeaders()	作为字符串返回完整的headers
getResponseHeader("headerLabel")	作为字符串返回单个的header标签
open("method","URL"[,asyncFlag[, "userName"[, "password"]]])	设置未决的请求的目标URL, 方法, 和其他参数
send(content)	发送请求
setRequestHeader("label", "value")	设置header并和请求一起发送

XMLHttpRequest对象的属性:

表 4-12 XMLHttpRequest 对象的属性

属性	描述
onreadystatechange	状态改变的事件触发器
readyState	对象状态(integer): 0 = 未初始化 1 = 读取中 2 = 已读取 3 = 交互中 4 = 完成
responseText	服务器进程返回数据的文本版本
responseXML	服务器进程返回数据的兼容DOM的XML文档对象
status	服务器返回的状态码, 如: 404 = "文件未找到"、200 = "成功"
statusText	服务器返回的状态文本信息

AJAX示例如下:

```
//AJAX请求↵
function AJAXRequest(url, isAjax, method, data, callback, charSet)↵
{↵
    var xmlhttp, result ;↵
    if(window.XMLHttpRequest)↵
        xmlhttp = new XMLHttpRequest() ;↵
    else if(window.ActiveXObject)↵
        xmlhttp = new ActiveXObject("Microsoft.XMLHTTP") ; ↵
    xmlhttp.onreadystatechange = function() {↵
        if(xmlhttp.readyState == 4) {↵
            if(xmlhttp.status == 200){↵
                if(xmlhttp.responseXML != null && xmlhttp.responseXML.xml != null &&
xmlhttp.responseXML.xml != '')↵
                    result = xmlhttp.responseXML ;↵
                else ↵
                    result = xmlhttp.responseText ;↵
                if(isAjax&&typeof(callback)=='function')↵
                    callback(result);↵
            }↵
            xmlhttp = null;↵
        }↵
    }↵
    if(url.indexOf("?")>0)↵
        url += "@random =" +Math.random() ;↵
    else↵
        url += "?@random =" +Math.random() ;↵
    xmlhttp.open(method, url, isAjax);↵
    xmlhttp.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");↵
}
```

图 4-8 AJAX 示例

4.6 VML 语言

VML的全称是Vector Markup Language(矢量可标记语言)，矢量的图形，意味着图形可以任意放大缩小而不损失图形的质量，这在制作地图上有很大用途。

VML以最简洁的标记代码，在WEB页快速绘制输出矢量图型（线段、圆形、矩形、圆矩形、矢量图像、曲线、多边形、弧型）、并修饰这些图形，在MAPGIS7-IMS行业版中，可以利用VML方便的画直线和封闭多边形，从而测量其长度和面积、公交线路显示等。

shape 是VML 最基本的对象，利用它可以画出所有你想要的图形。也就是说它能绘制圆形、矩形、圆角矩形、多边形、甚至于曲线等形状……不过它的形状用标记参数描述起来很复杂，而且VML 又有专门用于设计圆形oval、矩形rect、圆角矩形roundrect、曲线curve、多边形polyline 的标记。shape 中最主要的属性是Path，它是个功能强大的画笔。



图 4-9 VML 示例

4.7 XML 语言

XML是Extensible Markup Language的简写，一种扩展性标识语言。XML文档主要由一系列的元素(elements)和属性(attributes)组成。XML有很强的扩展性,我们可以根据自己文档的需要建立自己的标记(tags)，方便使用和记忆。

XML语言在MapGIS7-IMS行业版中主要用在哪些方面呢？

1. 单独的 XML 语言是不能显示页面的，我们使用 XML 语言与前面章节介绍过的各种技术（CSS 等）结合显示 XML 文档使我们的站点具有更多更好的功能和界面。
2. 跟 DOM（文档对象模型）完美的结合。DOM 就是如何用 HTML 或者 XML 对这个对象进行操作和控制的标准。XML 创建了标识，而 DOM 的作用就是告诉 script 如何在浏览器窗口中操作和显示这些标识。

XML文档在IMS中使用的常用对象的方法属性如下：

1. DOMDocument 对象常用方法在前面的 DOM 模型里已有讲述，如有需要请查看前述“DOM 模型”章节。
2. IXMLDOMNode 对象常用方法如下：

表 4-13 IXMLDOMNode 对象常用方法

方法	描述
appendChild(newChild)	该方法把节点 newChild 加到该方法使用的节点的子节点的末尾。
insertBefore(newNode, refNode)	该方法把一个新节点 newNode 插到一个已存在节点 refNode 之前
removeChild(nodeName)	该方法删除指定的节点
replaceChild(newNode, oldNode)	该方法用新节点 newNode 替换旧节点 oldNode
hasChildNodes()	如果 IXMLDOMNode 对象有子节点，该方法返回值为真（true）

3. IXMLDOMNode 对象常用属性:

表 4-14 IXMLDOMNode 对象常用属性

方法	描述
nodeType	返回节点类型
nodeValue	返回节点包含的文本
childNodes	返回用 IXMLDOMNode 对象表示的节点的子节点集
firstChild	返回该节点的第一个子节点
lastChild	返回该节点的最后一个子节点
text	返回 IXMLDOMNode 在 IXMLDOMNode 对象中的文本
xml	返回 IXMLDOMNode 对象的 XML 代码

4. IXMLDOMNodeList 对象常用方法:

表 4-15 IXMLDOMNodeList 对象常用方法

方法	描述
item(number)	该方法返回指定数值对应位置的节点
nextNode()	该方法返回节点集的下一个节点

4.8 JSP 技术

JSP (JavaServer Pages) 是一种基于Java的脚本技术。在JSP 的众多优点之中, 其中之一是它能将 HTML 编码从 Web 页面的业务逻辑中有效地分离出来。用 JSP 访问可重用的组件, 如 Servlet、JavaBean 和基于 Java 的 Web 应用程序。JSP 还支持在 Web 页面中直接嵌入 Java 代码。可用两种方法访问 JSP 文件: 浏览器发送 JSP 文件请求、发送至 Servlet 的请求。

在一个JSP文件第一次被请求时, JSP引擎先把该JSP文件转换成一个Java源文件, 在转换时如果发现JSP文件有任何语法错误, 转换过程将中断, 并向服务器和客户端输出出错信息; 如果转换成功, JSP引擎调用Java虚拟机的javac程序把该Java源文件编译成相应的class文件, 该class文件也就是一个Servlet程序; 然后创建一个该Servlet的实例, 提供服务响应用户的请求。

IMS中JSP技术指令:

表 4-16 IMS 中 JSP 技术指令

指令名	描述
JSP 页面构成指令	page 指令、include 指令、taglib 指令.在 IMS 中主要用到了第一个指令 page 指令, 描述了和页面相关的指示信息。
language	文件中所使用的脚本语言
Method	由嵌入的 Java 代码 (scriptlet) 生成的方法的名称。生成的代码会成为指定方法名的主体。缺省的方法是 service。当多次使用该指令时, 只有第一次使用是有效的。
Import	Servlet 导入的 Java 语言软件包名或类名列表, 该列表是用逗号分隔的。在 JSP 文件中, 可以多次指定该指令来导入不同的软件包。

IMS中JSP访问JavaBean的方法：利用Session内置对象访问JavaBean，Bean的生命期设置成Session。就是说如果当前对话中存在 Bean，则 Bean 已被重用。如果不存在 Bean，且创建属性设置为是，则它已被创建并作为会话的一部分存储起来。

IMS中使用的JSP内置对象：

1. request 对象

该对象封装了用户提交的信息，通过调用该对象相应的方法可以获取封装的信息，即使用该对象可以获取用户提交信息。

2. response 对象

对客户请求做出动态的响应，向客户端发送数据。常用 `setAttribute()` 方法，将一个特性存储在请求的上下文中。

3. session 对象

session 对象是一个 JSP 内置对象，它在第一个 JSP 页面被装载时自动创建，完成会话期管理。

4. application 对象

application 对象常用方法：

(1) `public void setAttribute(String key, Object obj)`: 将参数 Object 指定的对象 obj 添加到 application 对象中，并为添加的对象指定一个索引关键字。

(2) `public Object getAttribute(String key)`: 获取 application 对象中含有关键字的对象。

5. out 对象

out 对象是一个输出流，用来向客户端输出数据。out 对象用于各种数据的输出。

6. page java.lang.Object

对应 this 关键字。JSP 网页本身 page 对象是当前页面转换后的 Servlet 类的实例。从转换后的 Servlet 类的代码中，可以看到这种关系。

7. config

`javax.servlet. ServletConfig` 的实例，该实例代表该 JSP 的配置信息。常用的方法有 `getInitParameter(String paramName)` 及 `getInitParameterNames()` 等方法。事实上，JSP 页面通常无须配置，也就不存在配置信息。因此，该对象更多地是在 Servlet 中有效。

8. exception

`java.lang.Throwable` 的实例，该实例代表其他页面中的异常和错误。只有当页面是错误处理页面，即编译指令 page 的 `isErrorPage` 属性为 true 时，该对象才可以使用。常用的方法有 `getMessageO` 和 `printStackTraceO` 等。

以上对 JSP 技术在 IMS 中的应用只选了很小的一部分作了一个简要的介绍，如需了解更多，请自行查阅相关资料。

4.9 数据库

选择的数据库为：Oracle 9i 和 SQL Server 2000。这里采用的商用数据库是 SQL Server 2000。

IMS 示例站点主要涉及到一个点位信息数据 CityPosInfo 数据库。CityPosInfo 数据库是示例站点中所用到的数据库，包括设施点信息等数据。其存放各设施点的数据表包括 ImportantPntPos 表（各大城市所在位置，城市代号），各城市信息点表（如 C027, C871, 存储每个城市的具体信息点，例如：武汉的公交站，酒店等信息），EmailPntPos 表（存储用户通过邮件发送点的信息），LayerContrast 表、Line 表、Stop 表等。

GIS 服务器层的平台是 UNIX 或 Windows/NT 以及地理数据库。通过空间数据引擎

(Spatial Data Engine)，系统和商用数据库进行无缝联结。它完成数据的定义存储、检索、完整性约束以及有关的数据库管理工作，同时接收 WebGIS 服务器发送来的数据请求，并将处理结果交送 WebGIS 服务器。

系统中的数据可以采用文件系统（MAPGIS 的空间数据文件 SDF）方式存储，也可以采用商用关系数据库（如 SQL Server 或 Oracle）方式存储。一般建议使用数据库方式，方便管理、检索。MAPGIS 采用空间数据引擎(SDE)来管理数据库中的数据，它接收来自 WebGIS 服务器的数据请求，并将处理结果交送 WebGIS 服务器。

由于空间数据量大而结构复杂，且每一项图形元素代表了不同的物理含义，对其进行存储和有效检索存在复杂性。MapGIS 数据的底层存储系统采用 MS SQLServer 或 Oracle 系统，对此建立有针对网络数据源的管理机制，即对于 SQLServer 新建的一个数据库，MapGIS 对其建立 MapGIS 管理过程，在库中形成对点、线、区信息进行管理的一些管理表和存储过程。

5 程序调试方法

程序调试分为客户端脚本调试和服务端调试两部分。行业版二次开发提供两种不同环境下的开发方式，相应的服务器端程序调试的方法也有两种：.NET 平台下的调试方法和 JAVA 平台下的调试方法。

5.1 客户端脚本调试

借助.NET 二次开发工具 Visual Studio.NET 2005,可以很方便的对服务器端 JavaScript 脚本进行调试，调试步骤如下：

1. 打开 IE 浏览器-->工具-->Internet 选项... --> 高级，往下拉滚动条，拉到“浏览”后， 去掉“禁用脚本调试(Internet Explorer)”和“禁用脚本调试（其他）”上的选项（如图 5-1）。

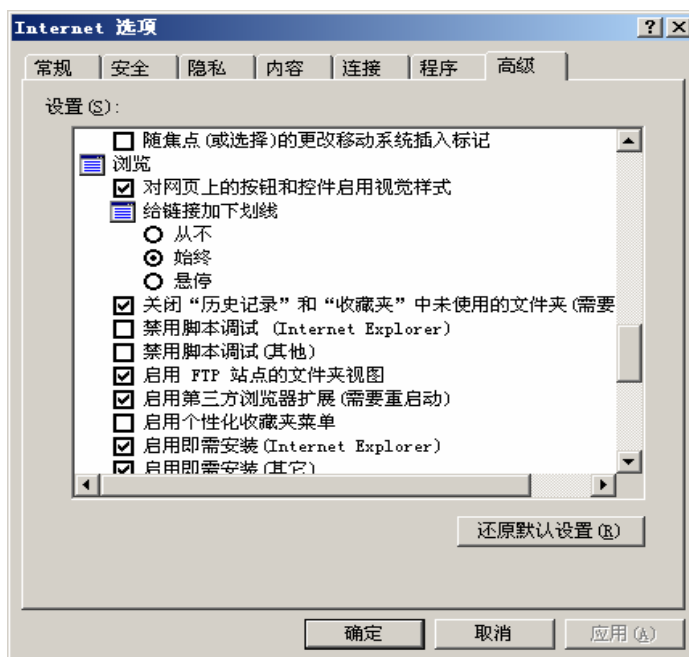


图 5-1 修改 Internet 选项

2. 直接在 IE 浏览器中打开要调试的网页（如图 5-2）。

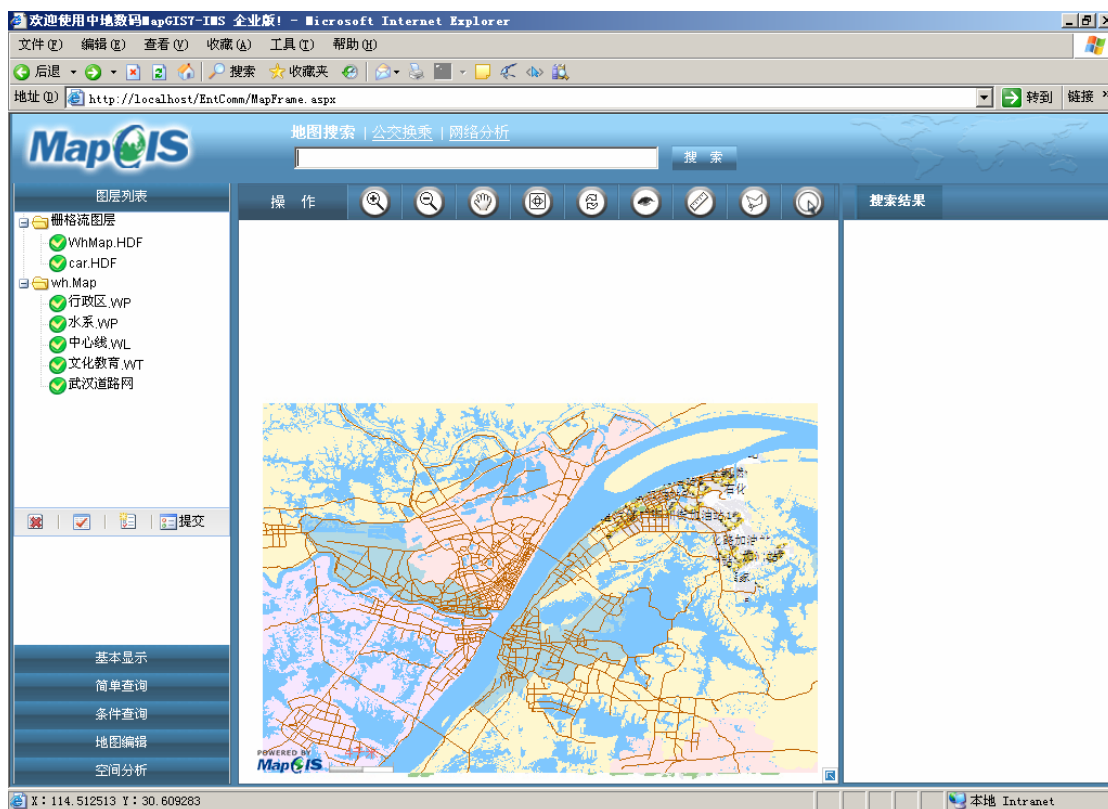


图 5-2 需调试的网页

3. 在 VS05 中打开你调试网页的项目，选择“工具”→“附加到进程”（Ctrl+Alt+p），如图 5-3 所示。

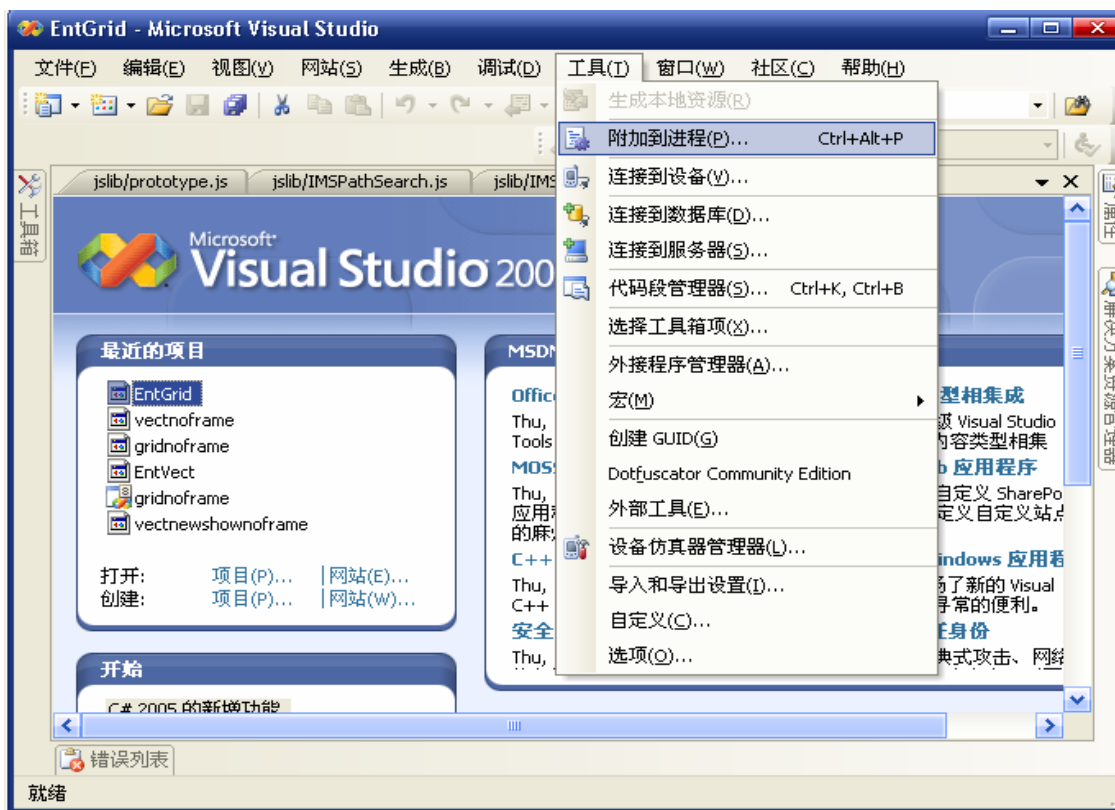


图 5-3 附加进程

4. 在附加到进程对话框的“可用进程”下选择需要附加的网页（如图 5-4）。

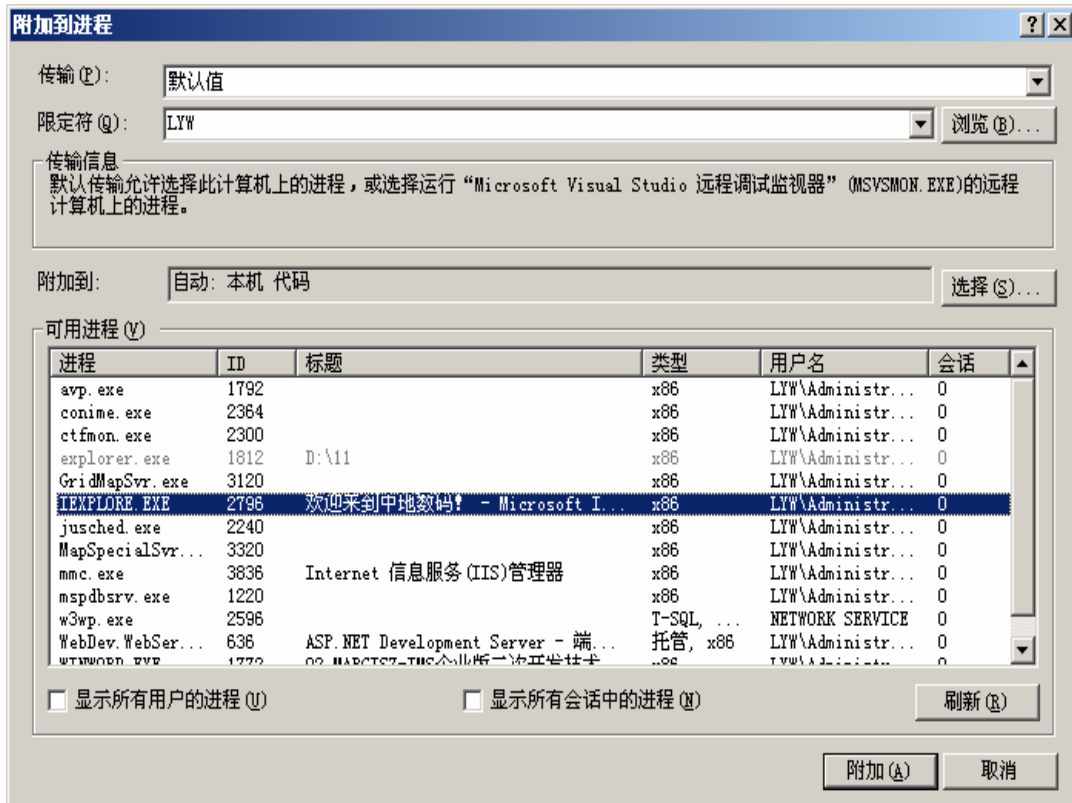


图 5-4 选择需要附加的网页

5. 点击附加到进程对话框中的<选择...>按钮，选择“脚本”选项，确定，关掉对话框。

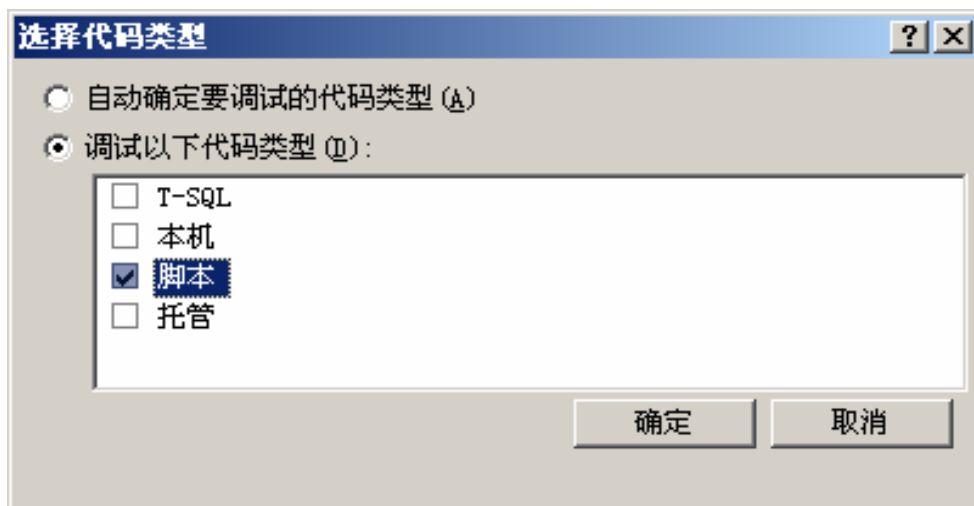


图 5-5 选择代码类型对话框

6. 选择你需要调试的 JavaScript 语句，并设置断点（如何设置断点将在后面的“设置断点”中讲述）。

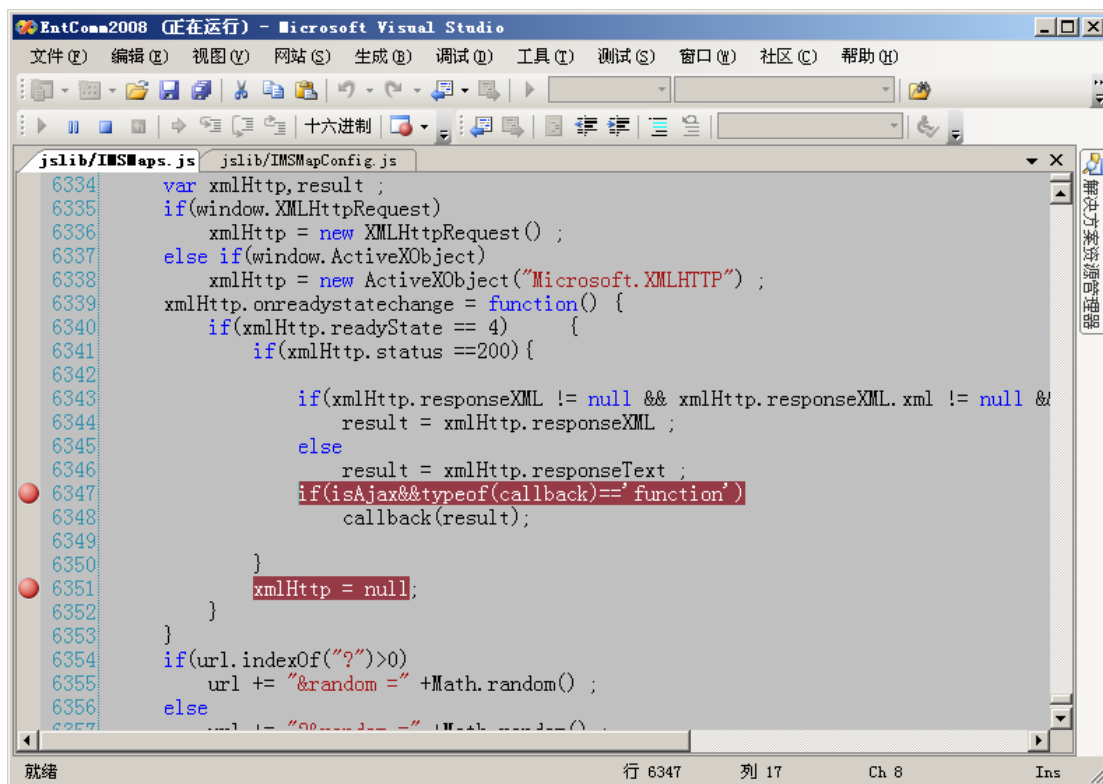


图 5-6 设置断点页面

7. 回到 IE 页面，运行与你所设断点的功能，自动中断到 VS05 上你所设断点上（如图 5-7）。
8. 选中要监视的参数，按快捷键 (Ctrl + D, Ctrl+ Q)，或者点击“调试”→“快速监视”添加快速监视，即可查看相关信息。如图 5-8 和图 5-9 所示。

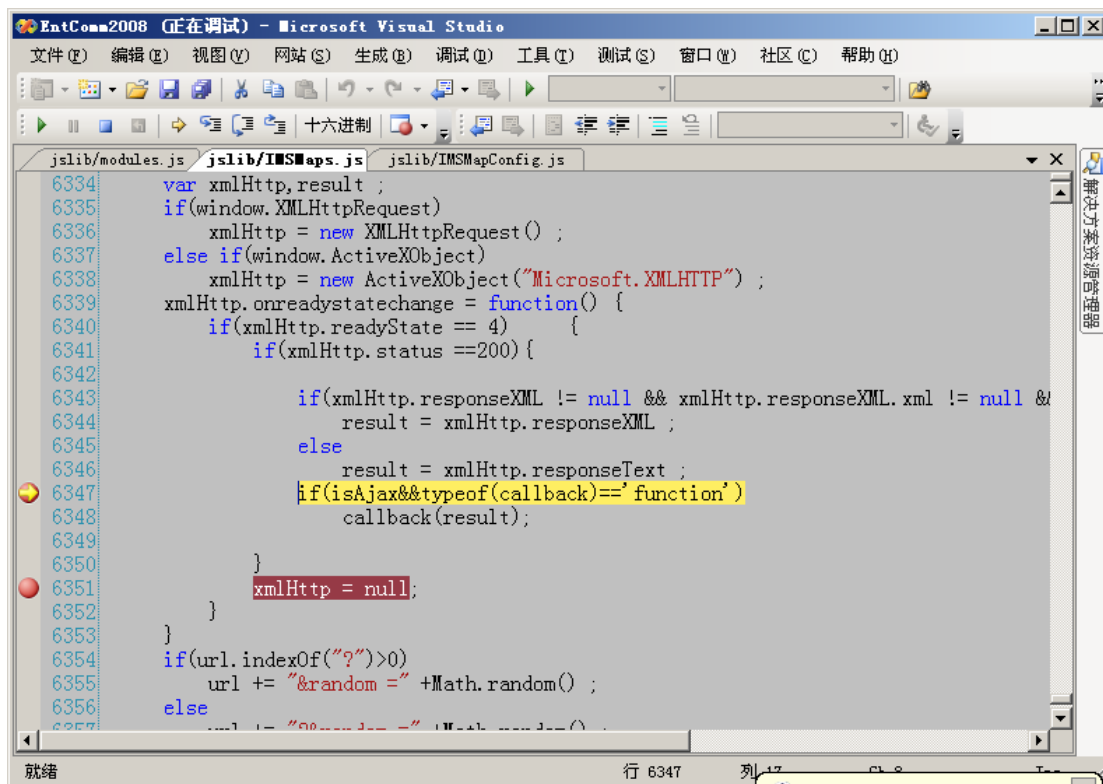


图 5-7 中断页面

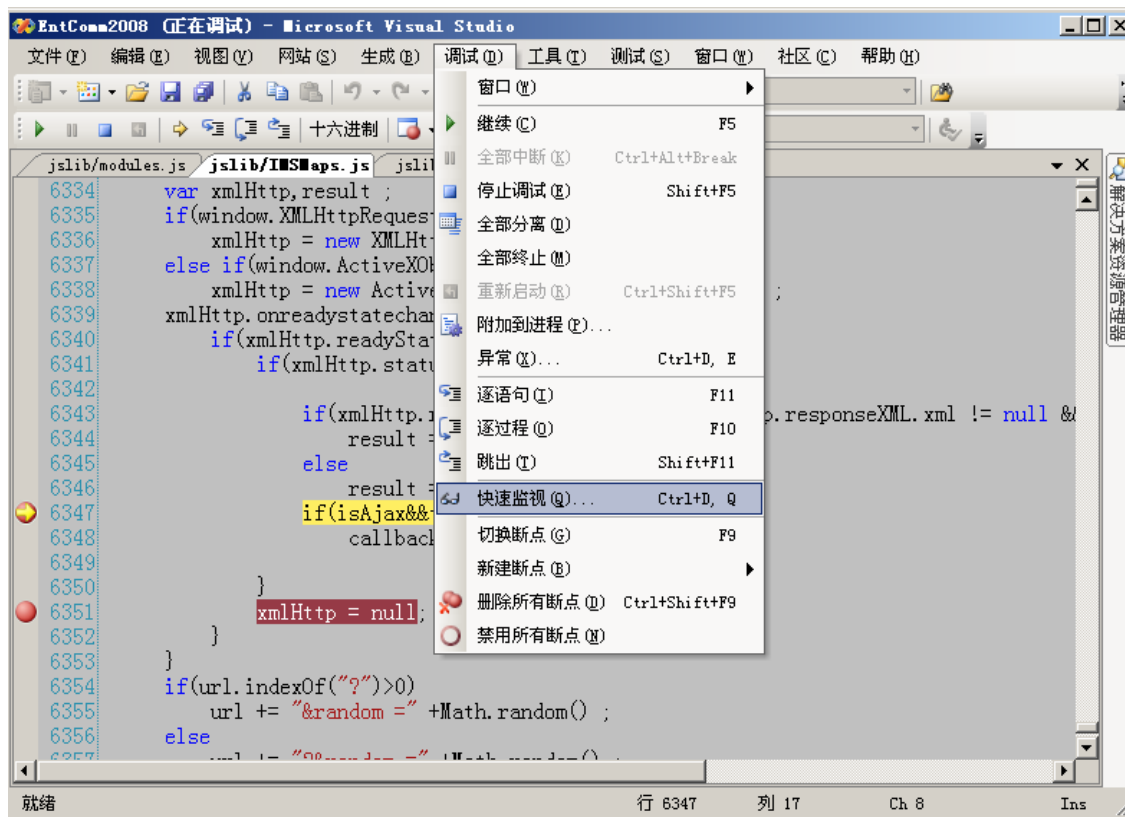


图 5-8 选择快速监视页面

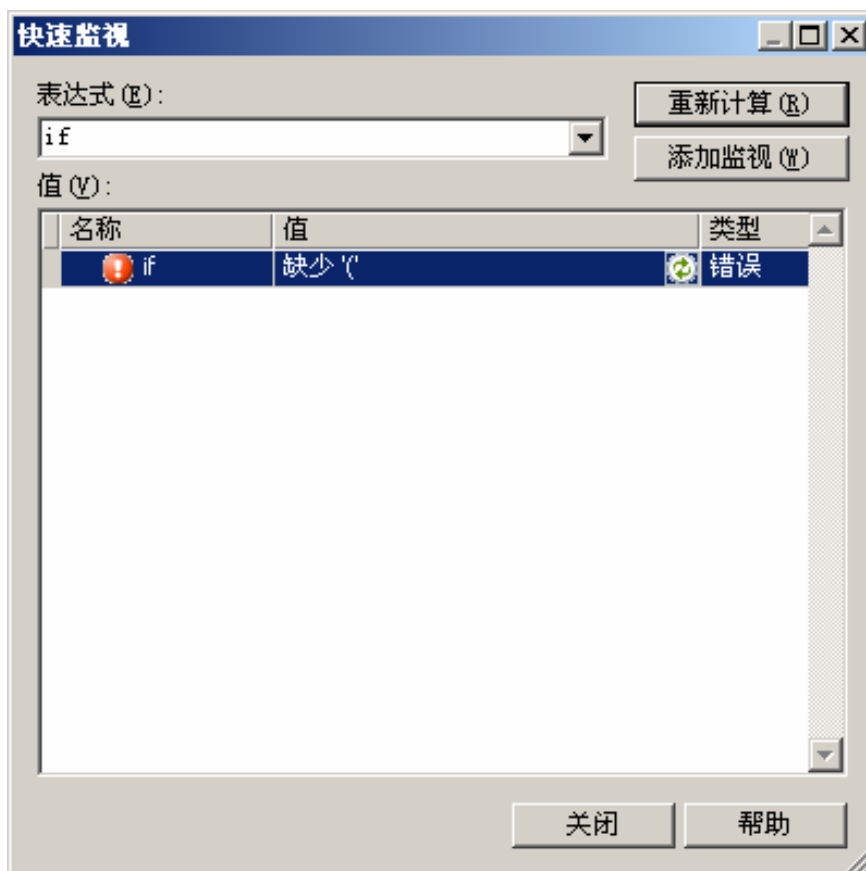


图 5-9 添加快速监视

另外，也可以借助 Microsoft Office 提供的脚本调试器（Microsoft Script Editor）。具体文件路径为：...\\Microsoft Office\\OFFICE11\\MSE7.EXE。界面如下：

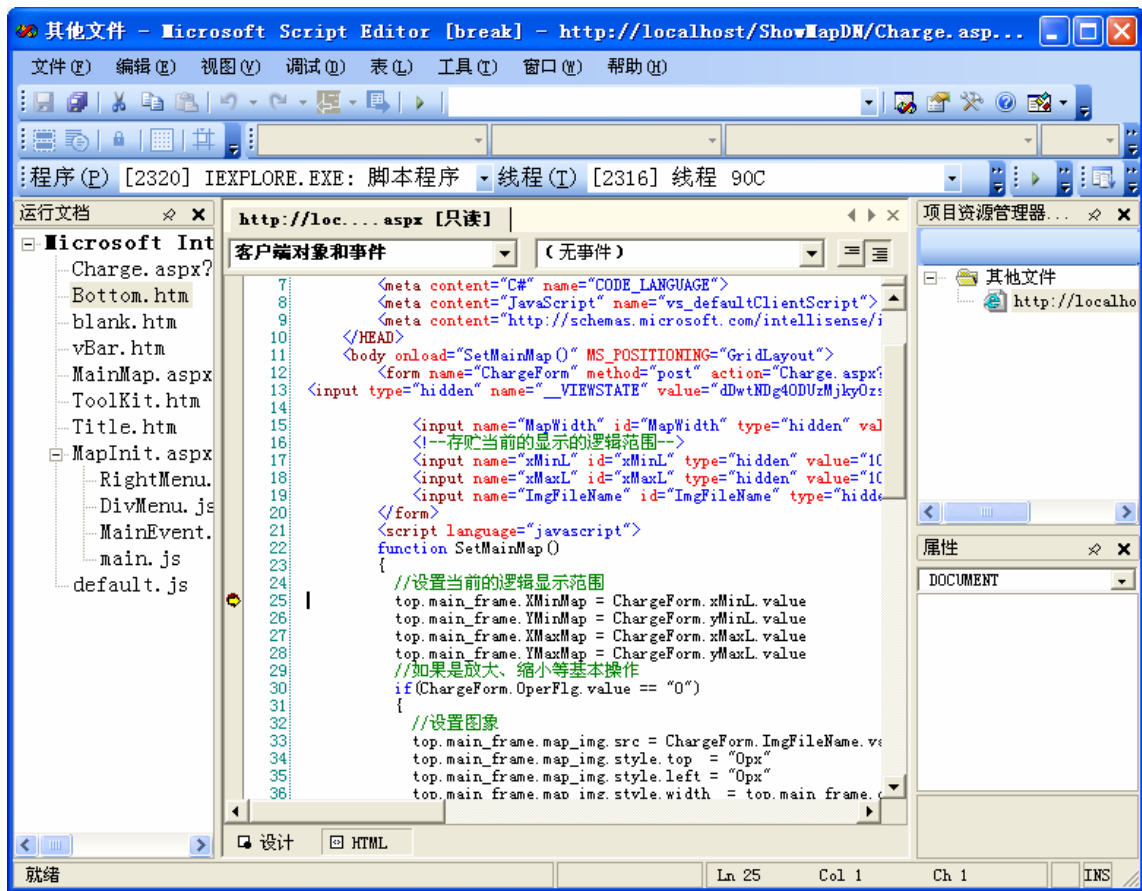


图 5-10 Microsoft Office 脚本调试器页面

其环境与功能基本上与 vs.net2005 相同，具体调试步骤可参照上面的.net 环境下的调试方法。

5.2 .NET 服务器端程序调试

进行服务器端程序调试前必须先配置环境为允许调试状态，下面是正式开始调试之前要做的准备：

1. 如果还没有为站点的配置文件 (Web.config) 设置调试模式，则先设置调试模式。要保证 debug="true"，即为允许调试模式。调试模式通知 ASP.NET 为动态生成的文件生成符号，并使调试器能够附加到 ASP.NET 应用程序。

如下图所示：

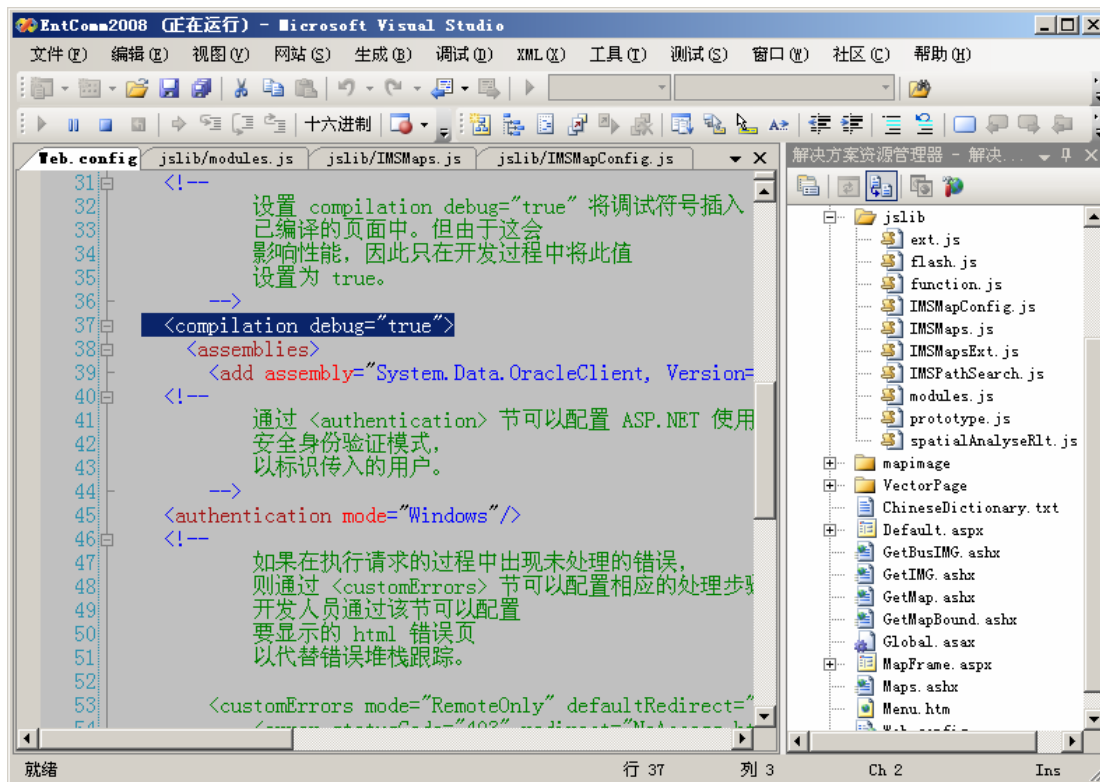


图 5-11 .NET 服务器端调试时修改 Web.config 页面

2. 在 Visual Studio 2005.NET 中，使用菜单“视图→属性页”设置用于 Web 应用程序调试的网站属性，如图 5-12 所示。

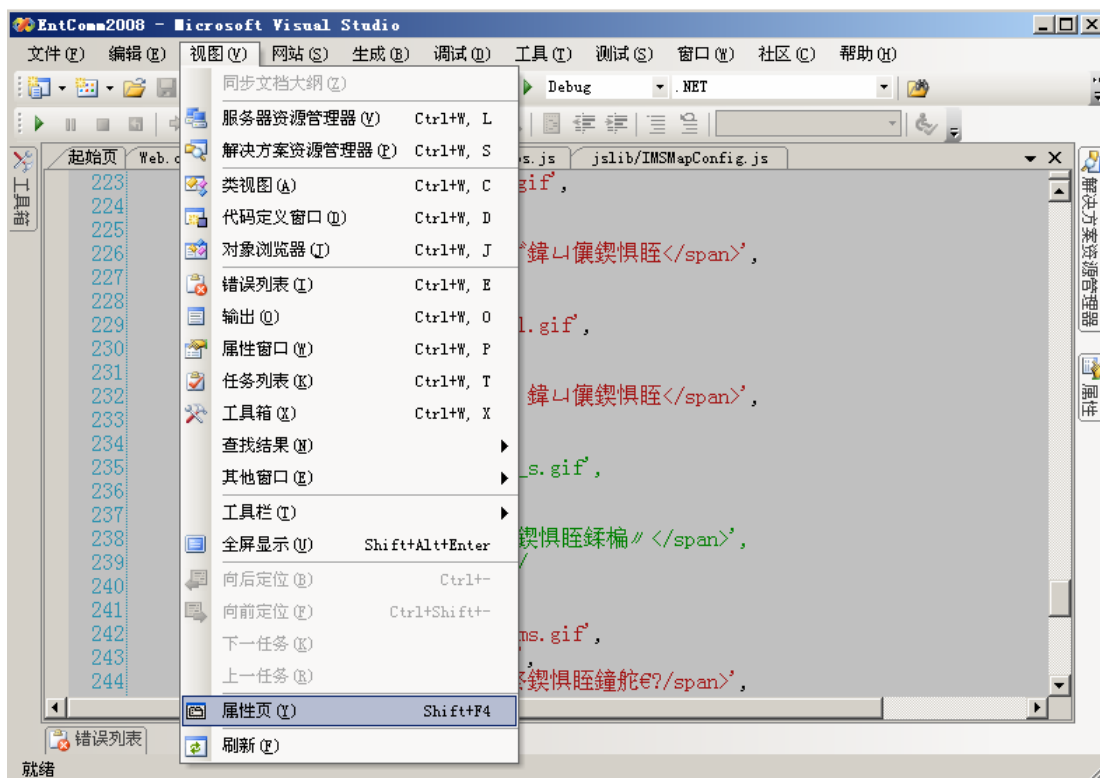


图 5-12 选择属性页

3. 点击“属性页”时跳出如下对话框，如图 5-13 所示。

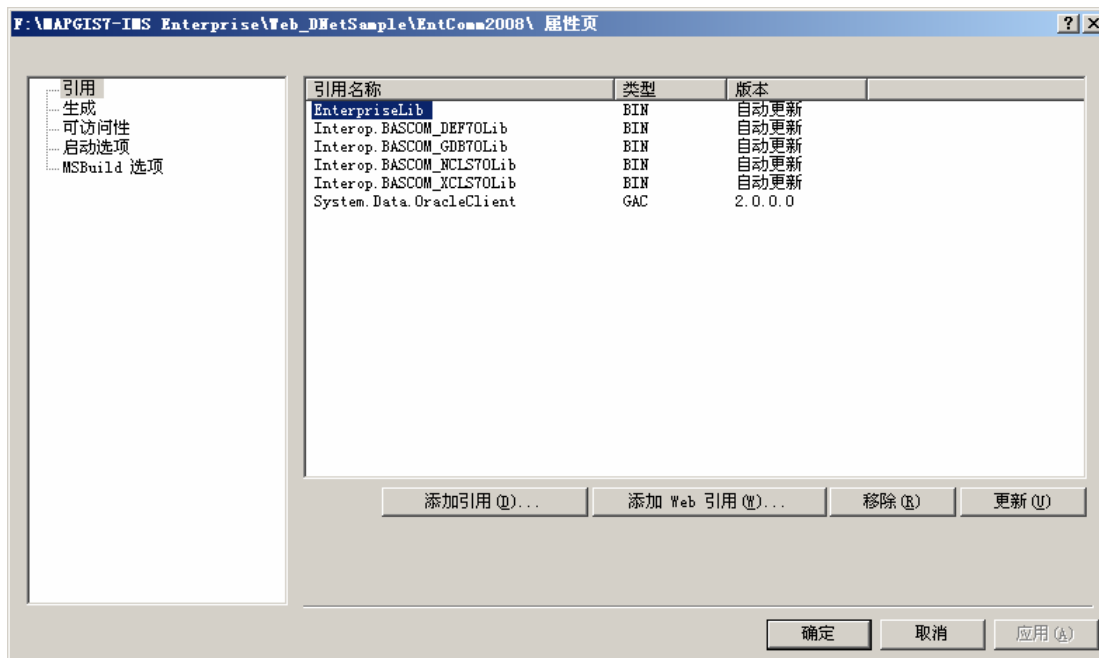


图 5-13 属性页面

4. 再选择左边的“启动选项”后，再选择右边的“特定页”，如图 5-14。

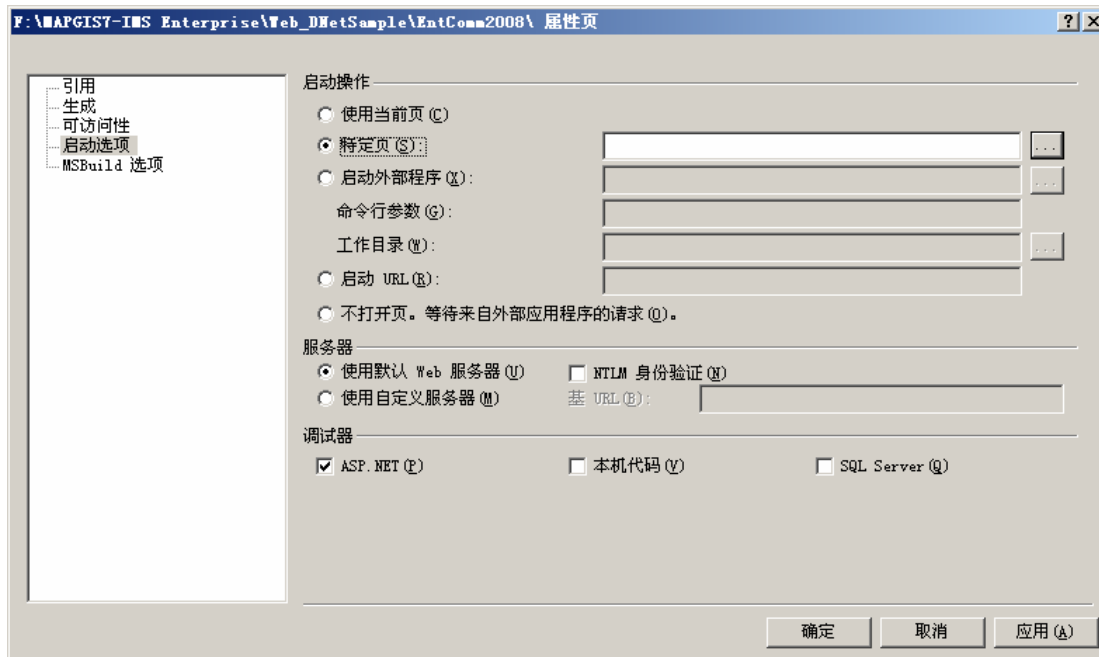


图 5-14 选择启动选项

5. 点击“...”浏览你所要选择的页面，点击一个页面（以 default.aspx 为例），如图 5-15。



图 5-15 选择特定页

6. 点击“确定”。

5.2.1 调试的快捷键

在 Visual Studio.NET 2005 中，调试所用快捷键及其功能如下：

- F5：开始调试
- F9：切换断点
- F10：逐过程进行调试
- F11：逐语句调试
- Shift+F5：结束调试

5.2.2 调试方法

调试方法为：先设置调试断点，然后启动调试，开始调试，找出程序错误，达到调试目的时就停止调试。设置多处断点时，需调出调试窗口查看。就以上调试方法我们将在下面章节详细介绍。

1. 设置断点

确保程序处于可调试状态后，在需要调试的地方设置断点，方法有两个：

- (1) 在光标停顿处按 F9。
- (2) 点击左边灰色区域。

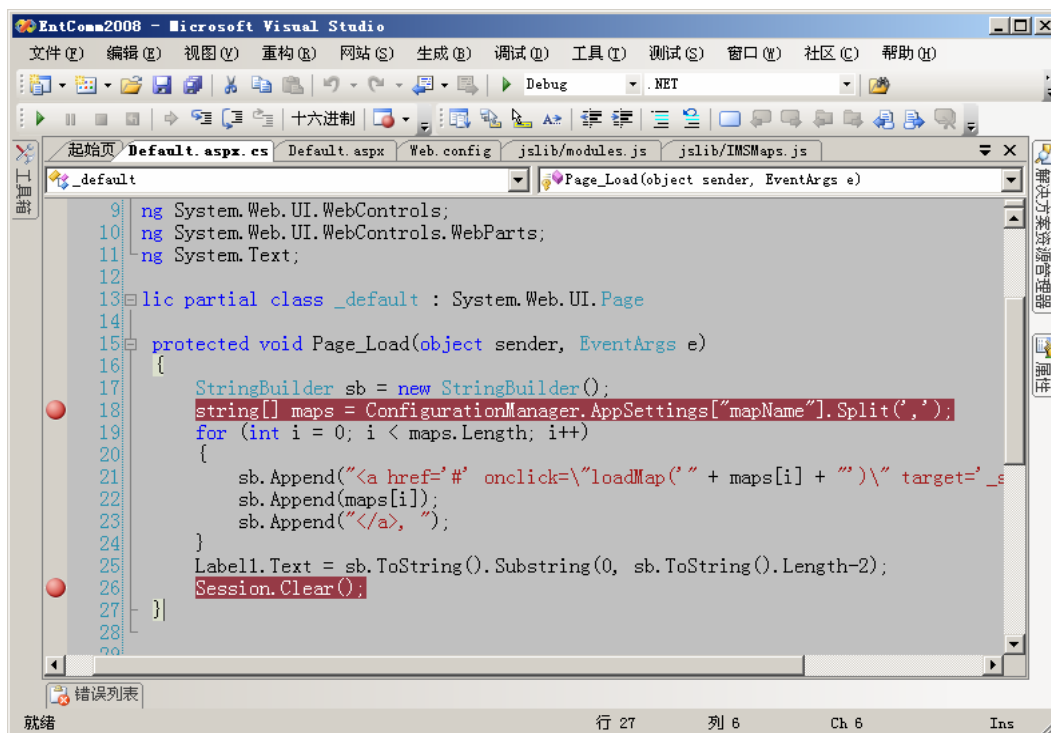


图 5-16 VS05 中设置断点

2. 启动调试

若要开始调试 Web 应用程序，请从“调试”菜单中选择“启动调试”或按快捷键 F5。

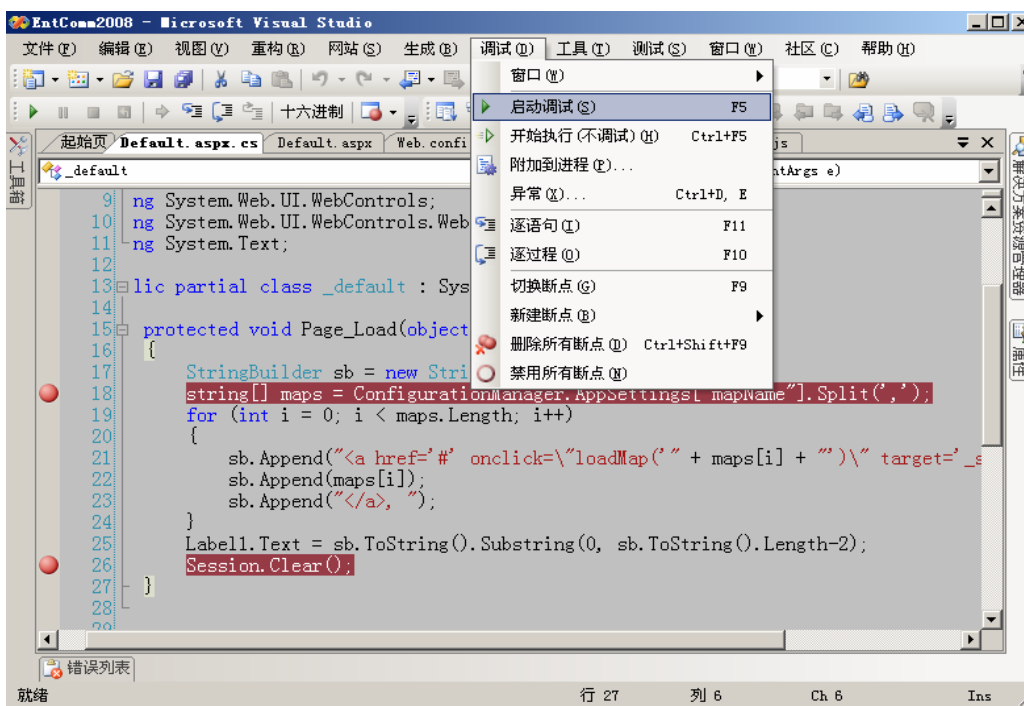


图 5-17 VS05 中启动调试

3. 开始调试

启动调试后程序就进入开始调试状态，程序进入断点处后，即可添加监视或其他操作进行调试。点鼠标右键，可选择进行几种快速调试，查看相应的变量。

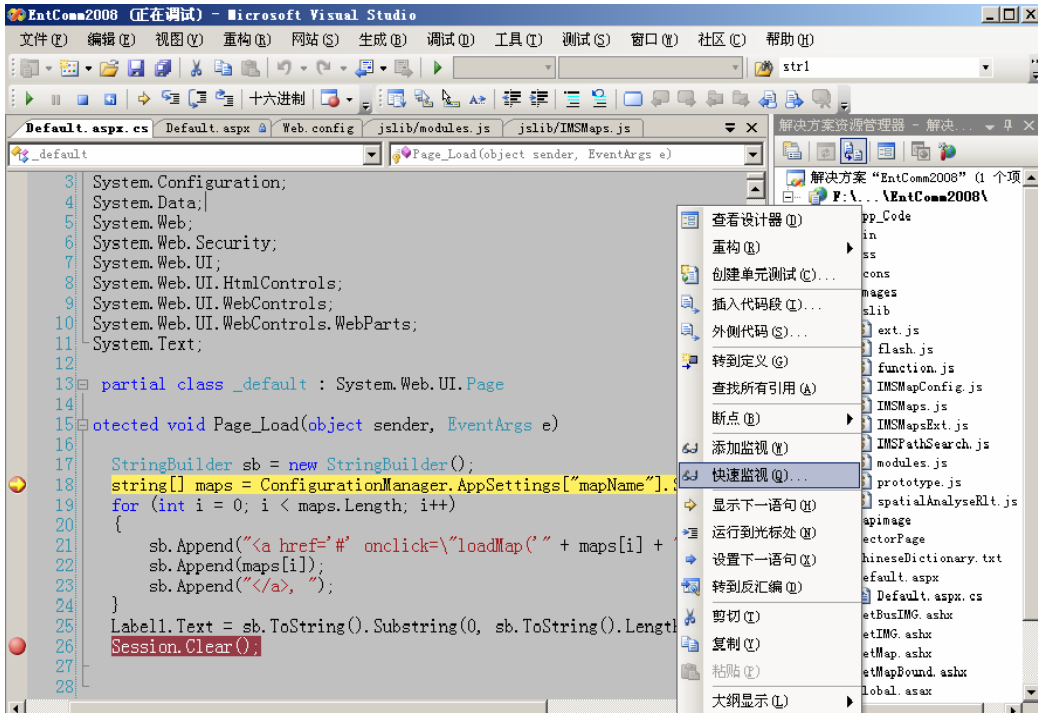


图 5-18 VS05 中开始调试

4. 停止调试

若要结束调试会话，请在 Visual Studio 中的“调试”菜单上选择“停止调试”命令或 Shift+F5。

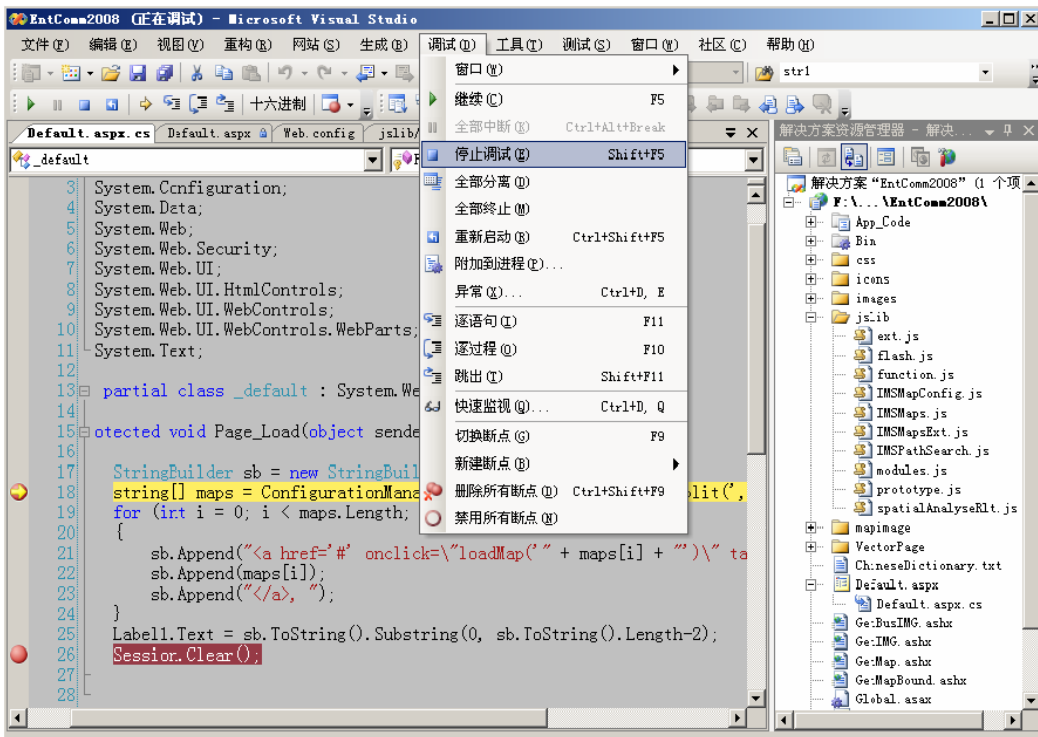


图 5-19 VS05 中停止调试

如果想更复杂准确的进行调试，您可以参看 VS05.NET 软件的帮助 msdn，里面有详细的调试步骤和方法。

调试小技巧——如何查看所设断点：

由于所设断点可能会在不同的页面中，也有可能上次调试时所设的断点还未删除，又想了解自己所设的断点都分散在何处。这时你只需查看所有的断点就可以满足你的要求。方法为：点击调试—>窗口—>断点，这时在 VS05 框架下方会出现“断点”窗口，窗口中显示的是本项目中所设置的所有断点；也可以选择组合键 **ctrl+D+B**，跳转到断点窗口。双击某个断点，就会跳转到断点所在页面。

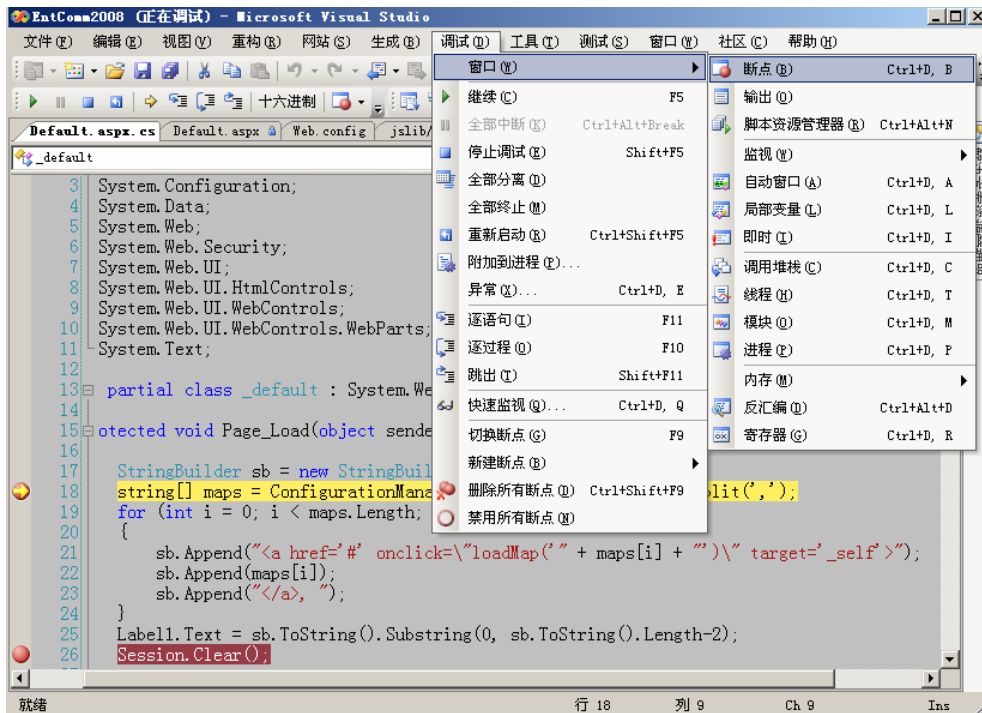


图 5-20 VS05 中选择断点窗口

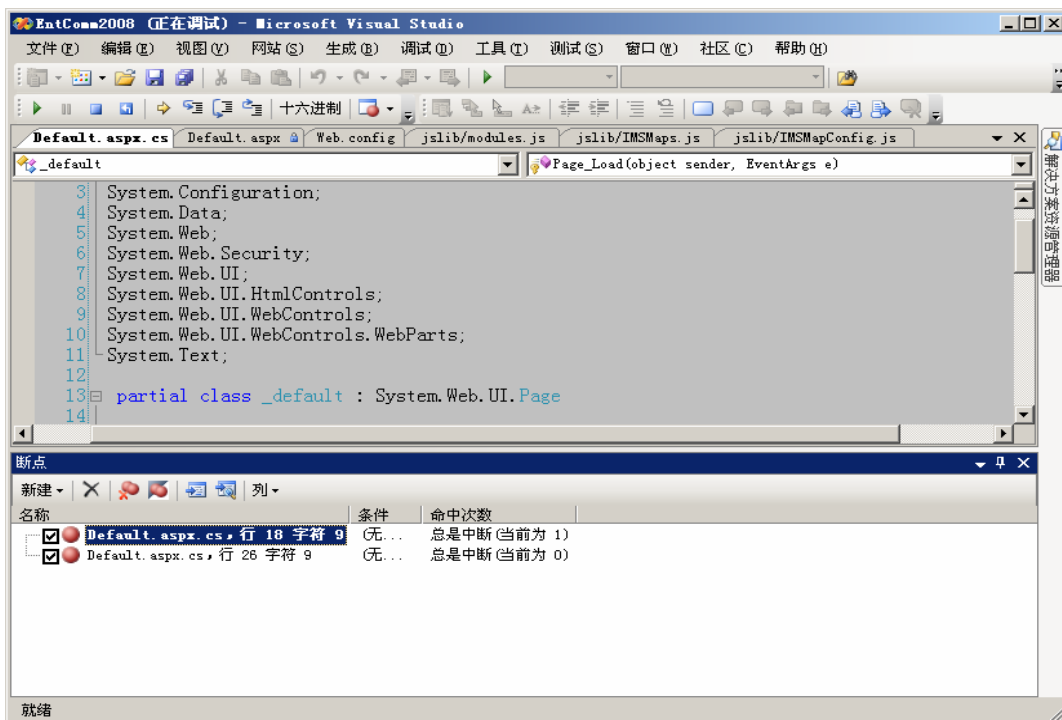


图 5-21 VS05 中断点窗口页面

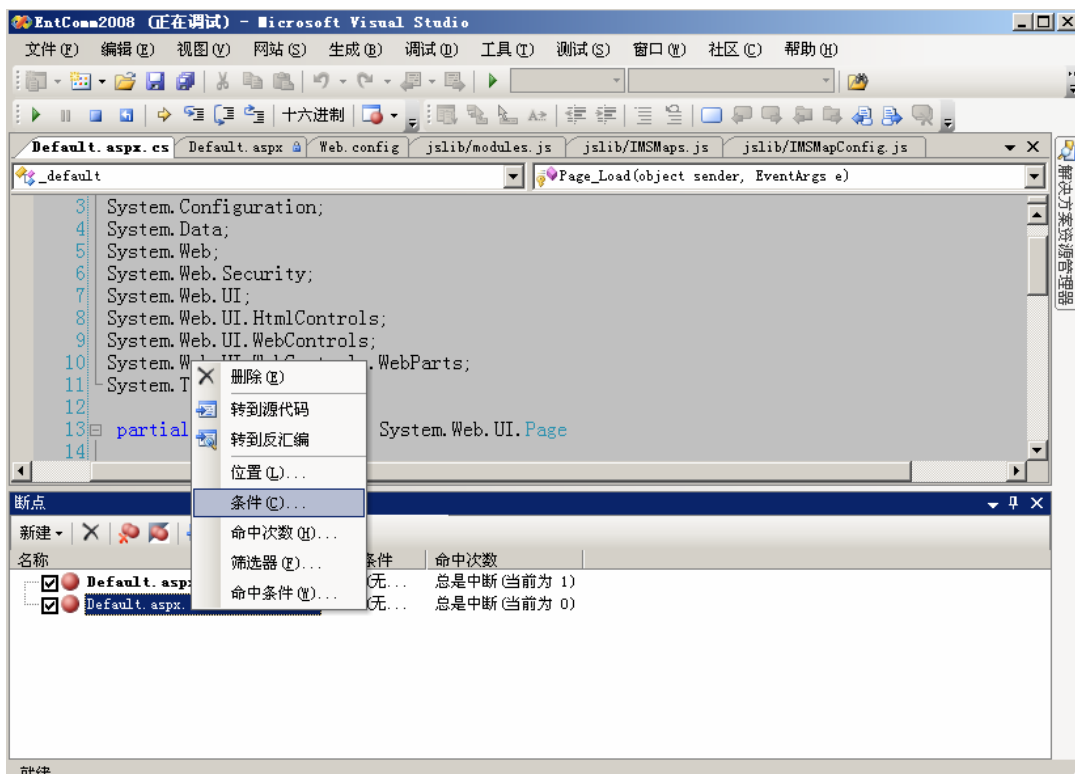


图 5-22 断点窗口选择某一点断点右键操作

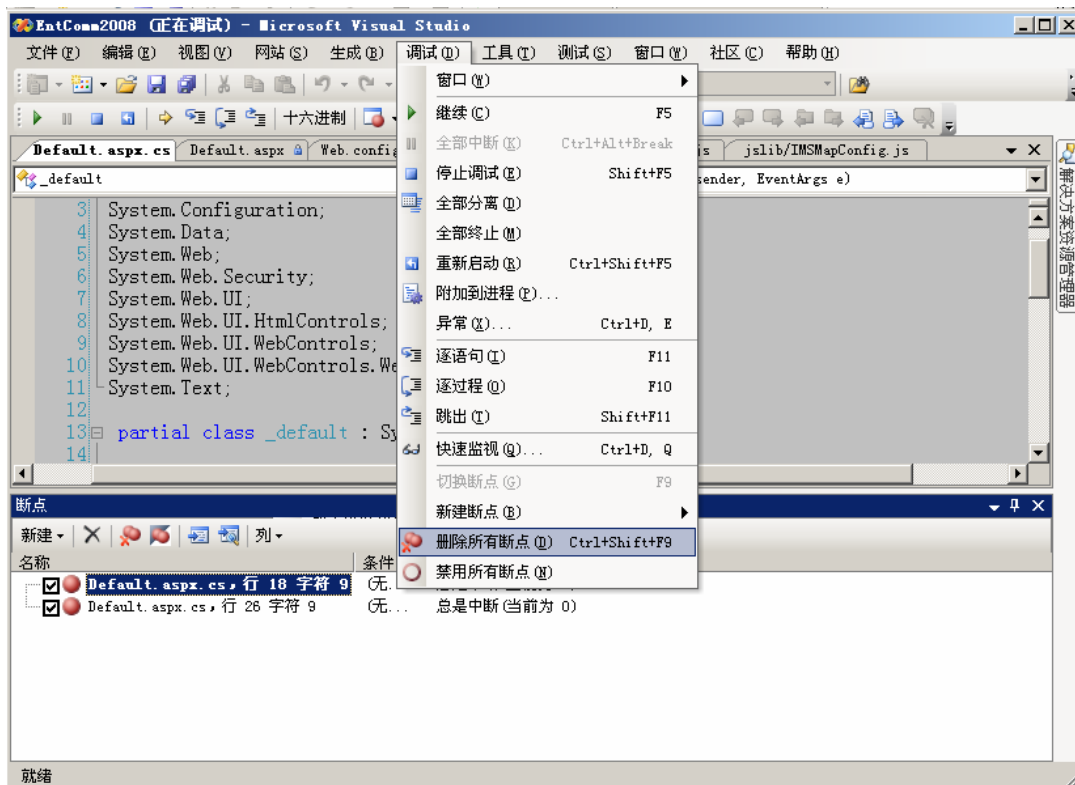


图 5-23 在 VS05 中删除所有断点

查看全部断点操作如上图 5-20 和图 5-21。在断点窗口中，选中某个断点后，右键可以对断点进行一些操作，如上图 5-22。也可以选择调试→删除所有断点，对断点进行删除，如上图 5-23。

5.3 JAVA 服务器端调试方法

借助 JAVA 开发环境中的 Eclipse 平台，可以非常方便的对 JSP 或 Java 程序进行调试。下面将详细的介绍 JAVA 服务器端调试方法。

5.3.1 调试的快捷键

在 Eclipse 开发平台中，调试所用的快捷键及其功能如下：

F6 单步跟踪；

F8 执至下一个断点如没有断点了则执行余下的程序；

F5 进入函数体；

F7 跳出函数体。

5.3.2 调试方法

调试的方法流程为：先设置好断点，启动调试后，就可正式开始调试。在调试过程中把鼠标放到调试语句上就可以看到相应的调试信息，调试完成后，就结束调试。下面将对上述步骤作详细的介绍。

1. 设置断点

在有疑问处，鼠标双击代码行左侧灰色边框可设置断点，如图 5-24。断点设置好后，就可以启动调试了。

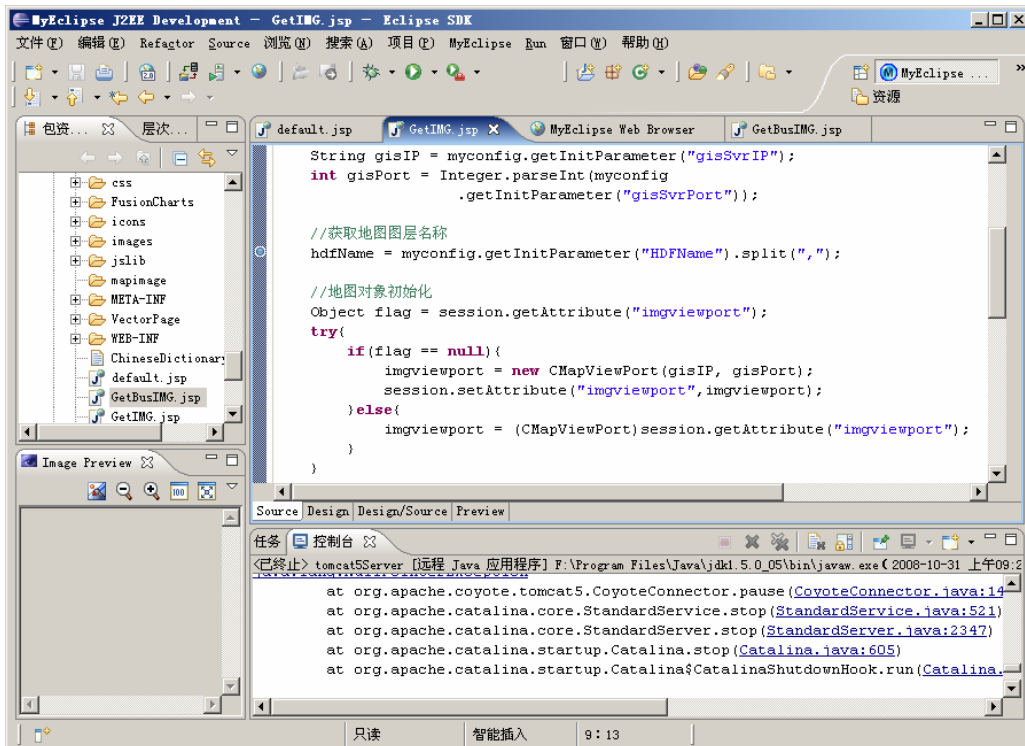


图 5-24 Eclipse 下设置断点

2. 启动调试

(1) 单击工程在菜单工具栏中的  选择开启 Tomcat 服务器，如图 5-25 所示。

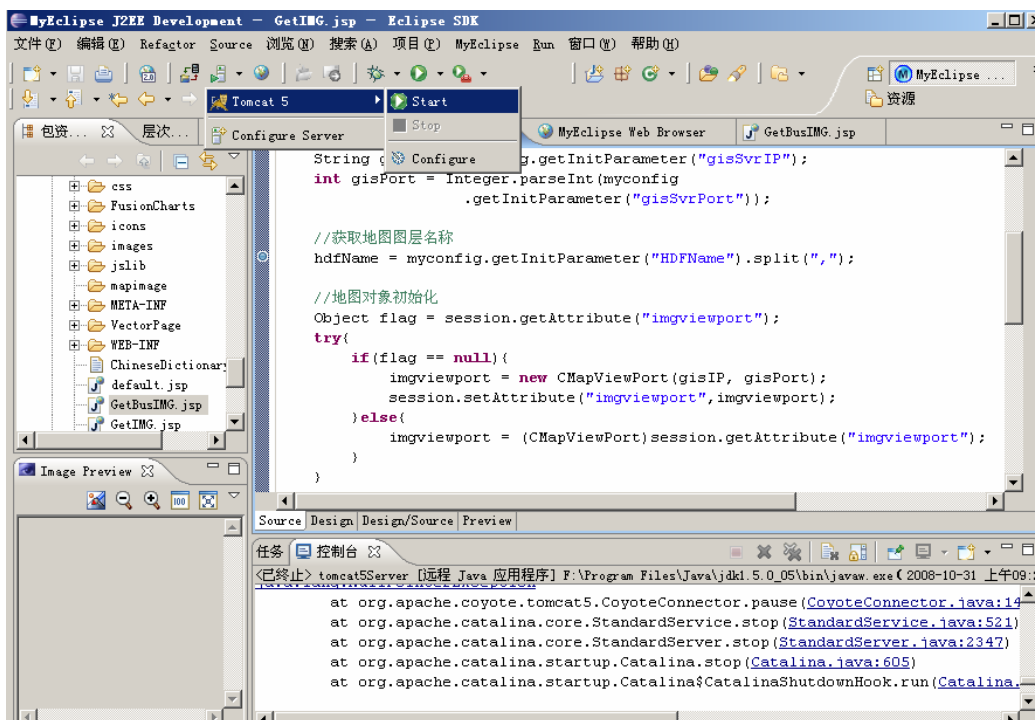



图 5-25 Eclipse 下启动 Tomcat

(2) 点击工具栏  打开浏览器，在地址栏输入网站地址，链接后如图 5-26 所示。

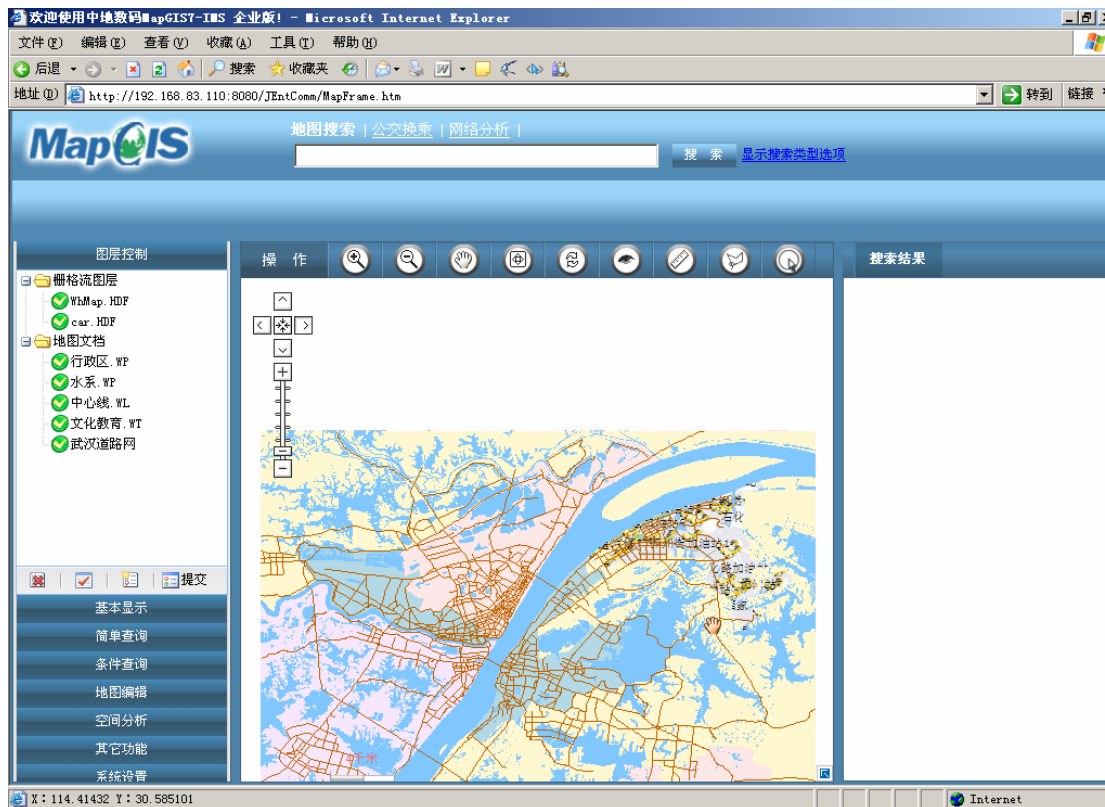


图 5-26 打开 JAVA 环境调试下的调试网页

3. 开始调试

在浏览器地址栏输入网站地址后，程序将进入调试界面，如图 5-27 所示。此时我们就可以使用逐步调试 F6；F5 进入函数体内；F7 跳出函数体；F8 执行完毕（相当于 VC 中的 F5）来对程序进行调试，并监视程序段中的变量值 Variables。开始后的界面如图 5-27 所示。

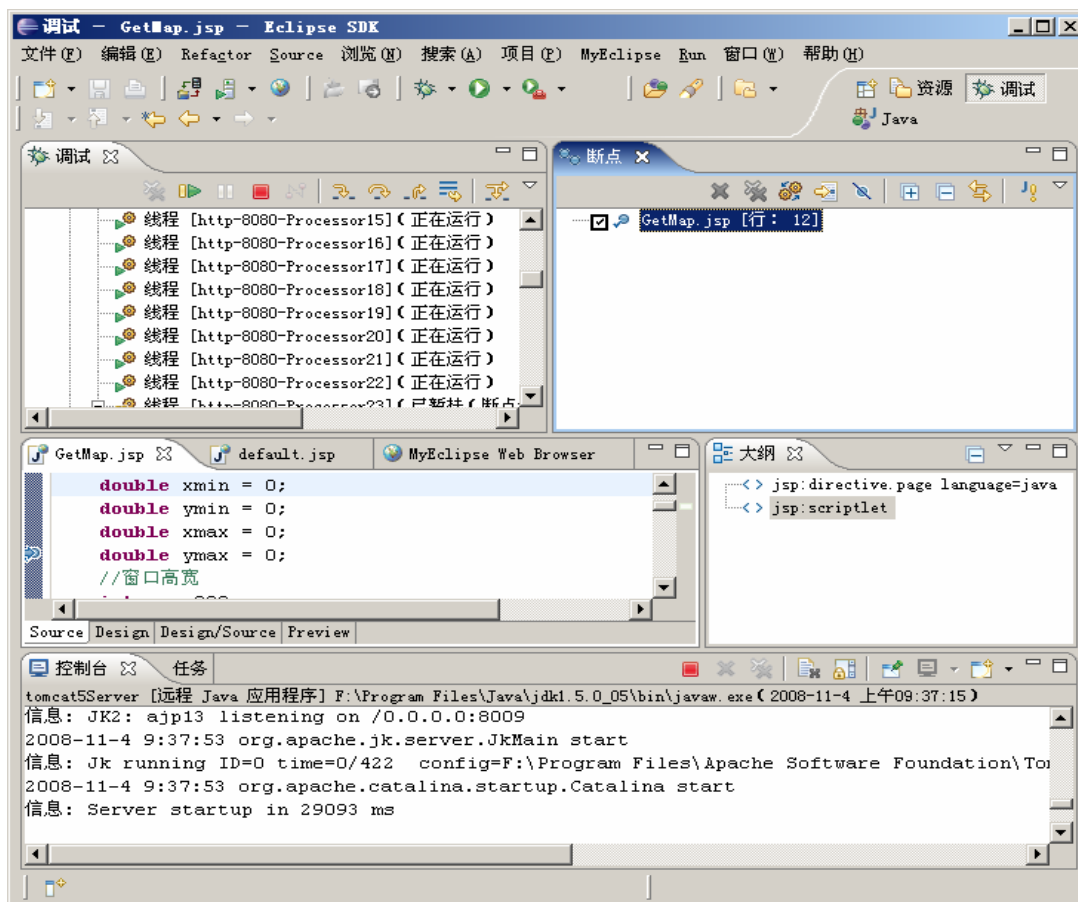



图 5-27 Eclipse 下启动调试页面

4. 停止调试

停止调试有两种方法：

- (1) 点击工具栏上的“Run”选择“终止”，如图 5-28 所示。
- (2) 选择左上角或右下角的红色小方块 ，停止调试。

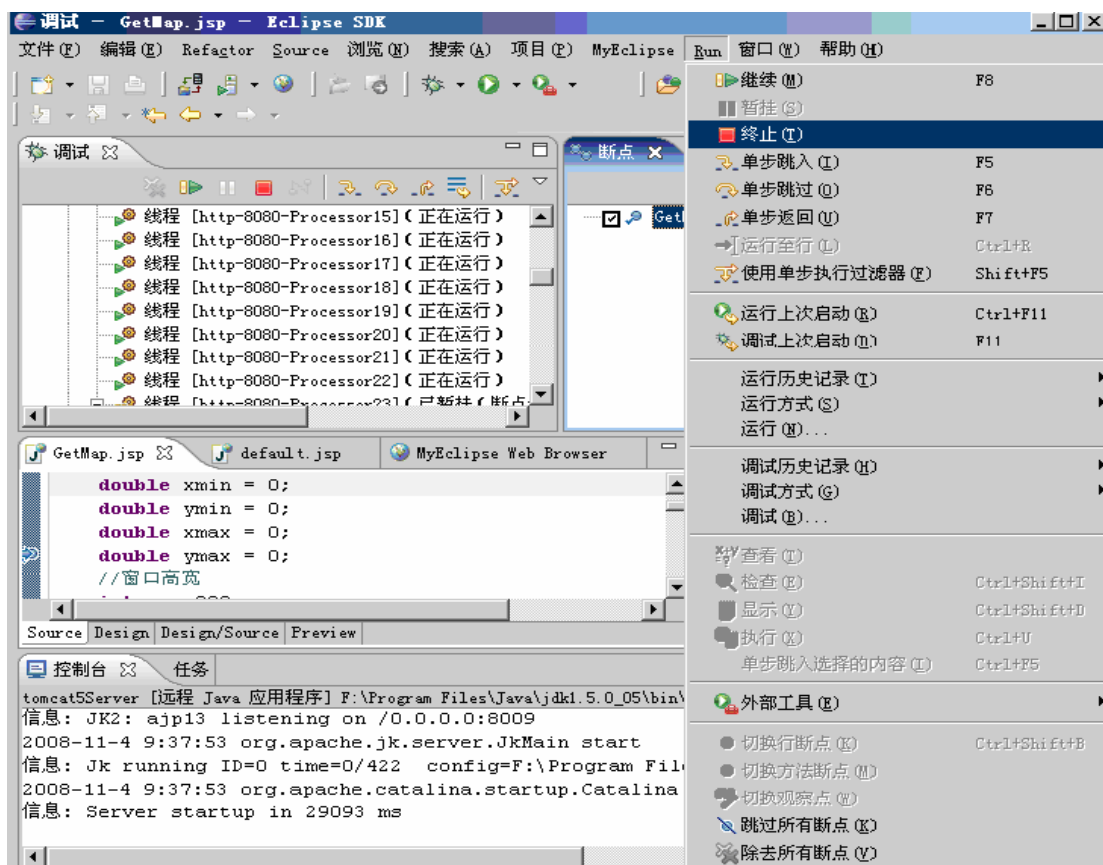


图 5-28 Eclipse 下停止调试

调试小技巧——如何查看所设断点：

由于所设断点可能会在不同的页面之中，也有可能上次调试时所设的断点还未删除，这样您希望能够了解自己所设的断点都分散在何处。点击框架中的 Window(窗口)->Show View (显示视图) -> others(其他) ->调试->Breakpoints，这些步骤完成后你会发现框架下方又多开了一个子断点窗口，窗口中显示的是所有设置的断点，在调试代码前最好清除上次设置的断点，尽量避免由于断点设置过多或是断点设置混乱而带来的不便。流程如下：

窗口->显示视图->其他->调试->断点，如图 5-29 和 5-30，单击“确定”，就可以看见你设置的所有断点，如图 5-31 所示。

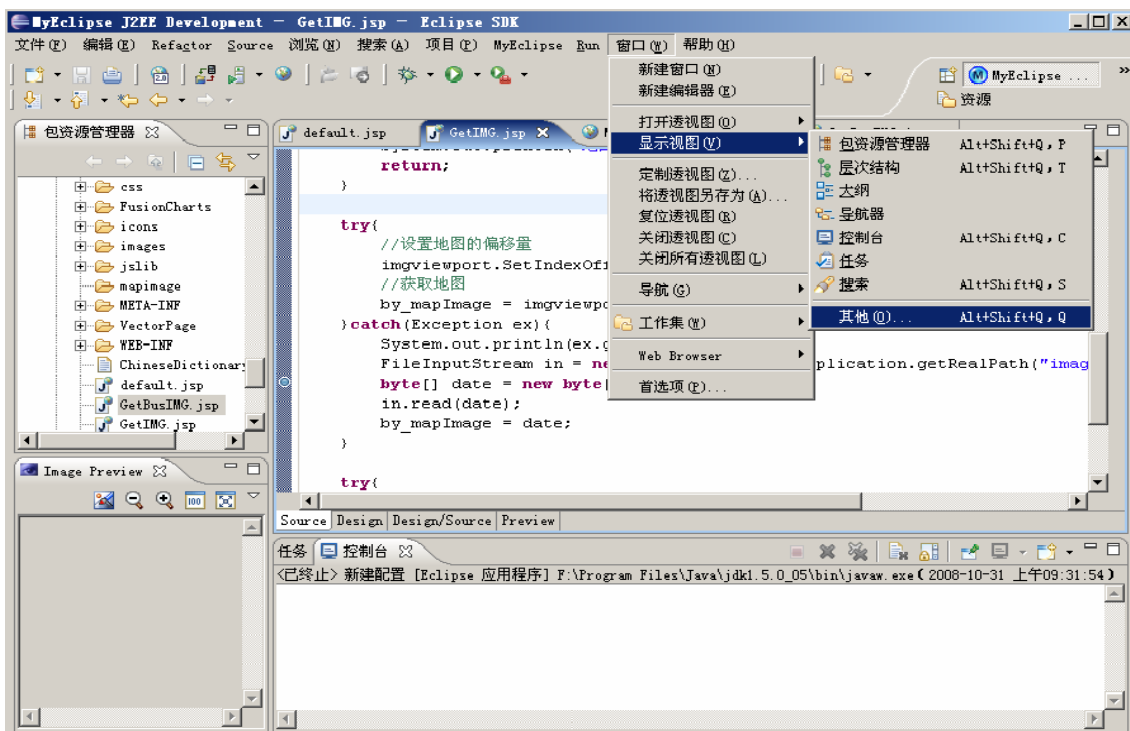


图 5-29 Eclipse 下选择断点窗口步骤 1



图 5-30 Eclipse 下选择断点窗口步骤 2

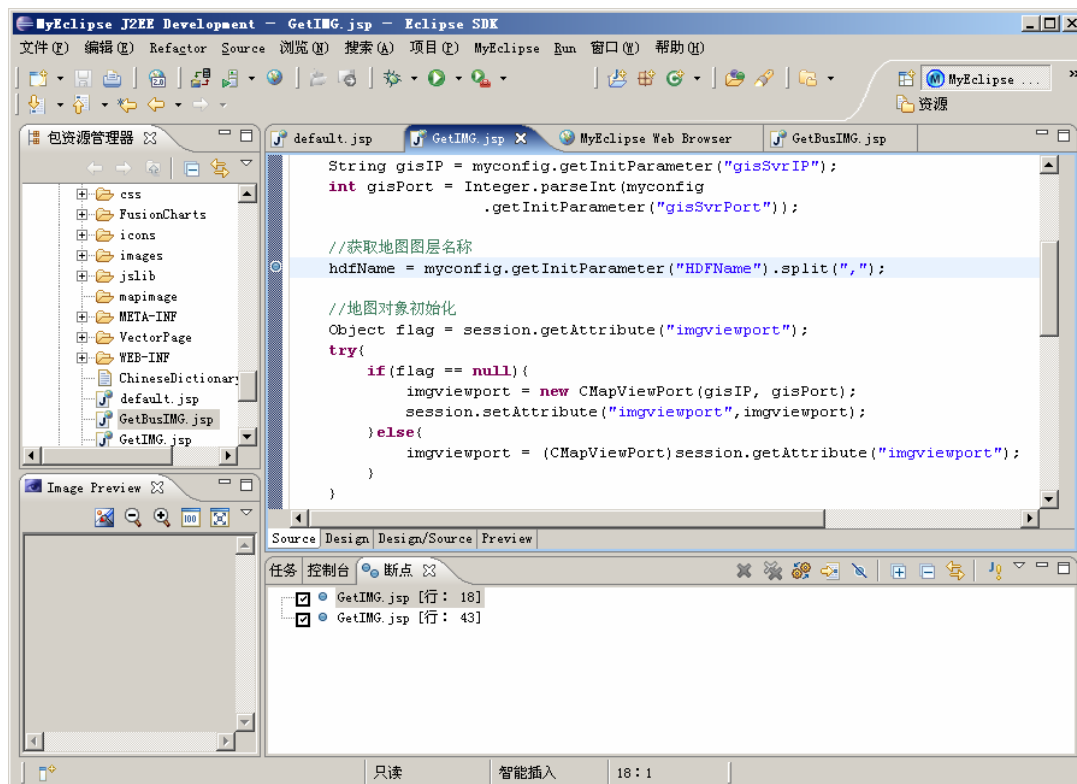


图 5-31 Eclipse 下断点窗口

以上就是 JAVA 环境的程序调试方法，如需了解更加详细的调试方法，你可以参照 Eclipse3.2 的帮助手册，里面说明了详细的调试方法和调试信息。