

MAPGIS VirtualEarth

二次开发技术篇

目录

目录.....	- 2 -
1 概述.....	- 3 -
2 开发环境介绍.....	- 3 -
2.1 开发环境.....	- 3 -
2.2 开发软件---Eclipse.....	- 4 -
3 数字地球二次开发技术.....	- 14 -
3.1 JavaScript脚本语言.....	- 14 -
3.2 Java语言.....	- 16 -
3.3 HTML语言.....	- 18 -
3.4 JSP技术.....	- 19 -
3.5 数据库（SQL Server/Oracle）.....	- 20 -
3.6 XML语言.....	- 21 -

1 概述

MapGIS VirtualEarth 是中地数码拥有自主知识产权的一套显示三维地球的软件，它依托 MapGIS 平台为基础，基于 Java applet 技术和 JOGL 技术开发。采用面向 Internet 的分布式计算技术，支持跨平台、跨网络的大型网络应用系统的集成。

为了方便用户的二次开发，下面将会从开发环境、开发中涉及的主要技术等方面入手，详细的说明 MAPGIS VirtualEarth 二次开发的有关事项。

2 开发环境介绍

MAPGIS VirtualEarth 二次开发的环境是基于开发工具 MyEclipse 6.0、开发语言 JAVA、Applet、OpenGL 的。这些工具和技术是当下 J2EE 开发的主流技术，具有各自的特点，有助于用户更好地进行 MAPGIS VirtualEarth 的二次开发。在开发之前需要配置开发环境，设置环境变量。配置好环境后，就能利用功能强大的 JAVA 工具及技术开发 WEB 项目。

2.1 开发环境

开发环境配置如下：

- 1、操作系统：WindowsXP、Windows2003 Server（推荐）
- 2、浏览器：IE 6.0 及以上版本
- 3、环境支持：jdk1.6 及以上、tomcat5.0 及以上
- 4、数据库：SQL Server、Oracle

2.1.1 MAPGIS VirtualEarth 的开发环境

MAPGIS VirtualEarth 的开发环境配置如下：

1. 按上述配置开发环境，安装好 JDK1.6 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：Eclipse3.2，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. 准备好地图数据和相关的点位信息数据，并参照相关的配置文档配置好 GIS 服务器、相关数据和数据服务器。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 JEarthDemo_Data.MDF 附加到本地 SQL Server 2000 数据库里。
5. 如果有用户自定义数据，则站点开发过程中须保持 GIS 服务器（GridMapSvr）呈启动状态。

2.1.2 MAPGIS VirtualEarth 的开发语言

MAPGIS VirtualEarth 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。

2.1.3 MAPGIS VirtualEarth 所用的文件类型

MAPGIS VirtualEarth 中用到的文件类型和重要文件如下表所示：

表 2-1 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。

2.2 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：

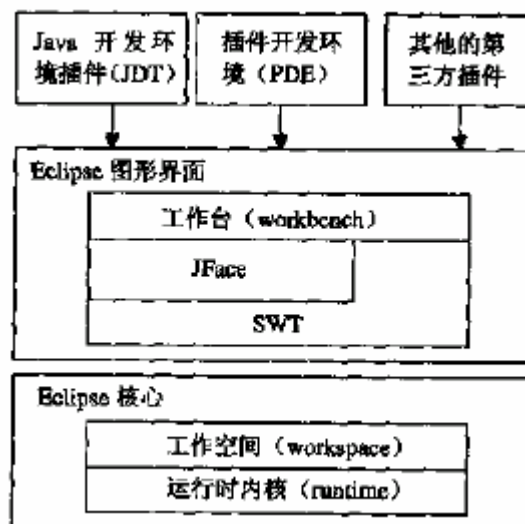


图 3-1 Eclipse 的体系结构图

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。

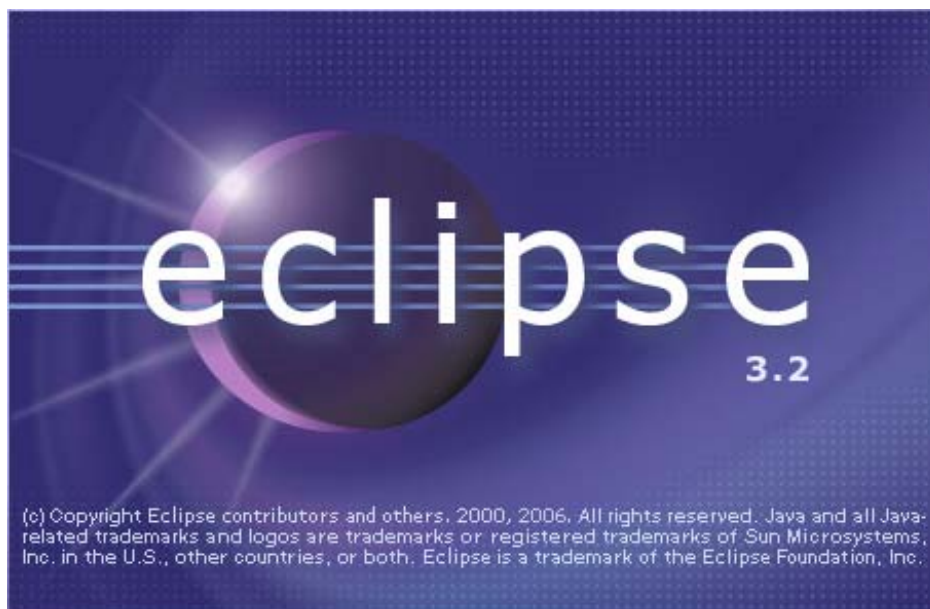


图 3-2 Eclipse 启动时界面

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。

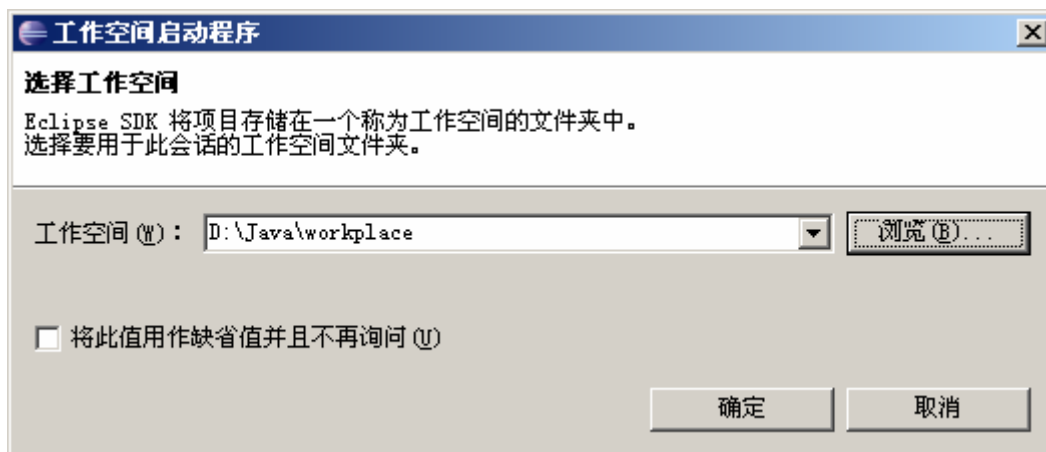


图 3-3 选择工作空间页面

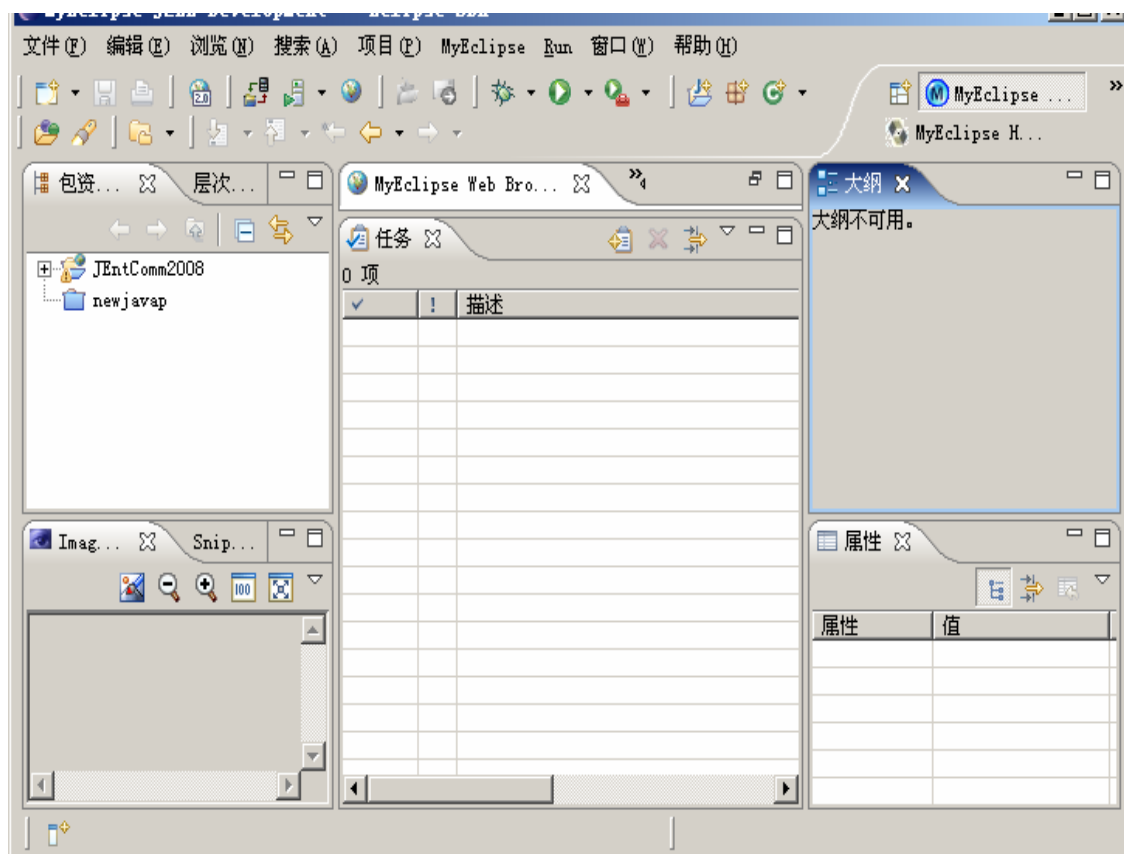


图 3-4 Eclipse 工作环境

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

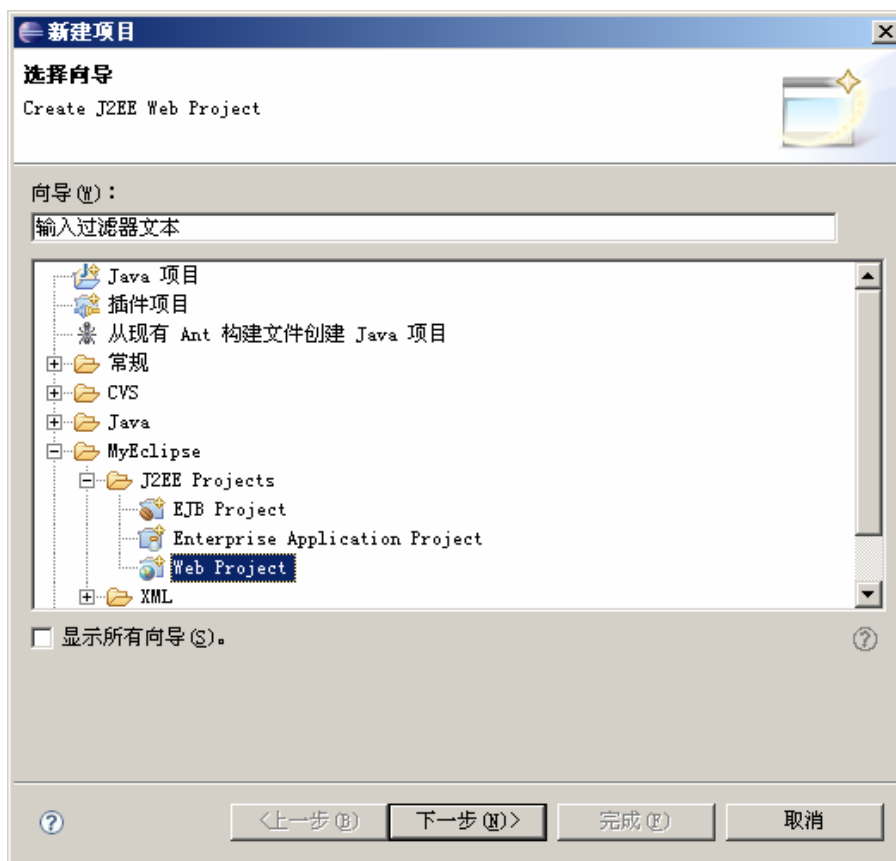


图 3-6 选择 Web Project 新建 Web 工程

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。

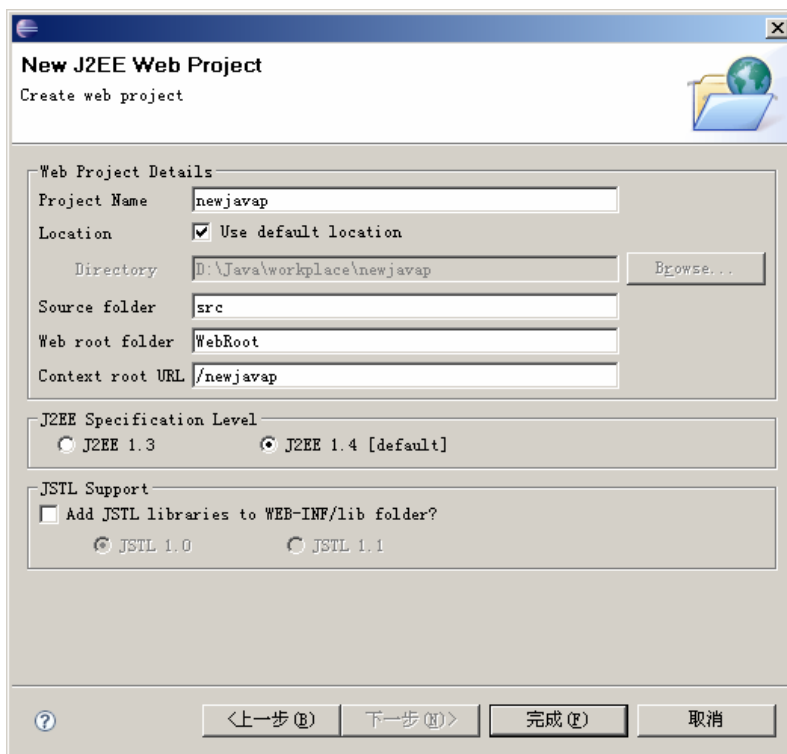


图 3-7 输入新工程名

6. 点击“完成”，新的 **WEB** 站点创建成功。完成后的界面如图 3-8。

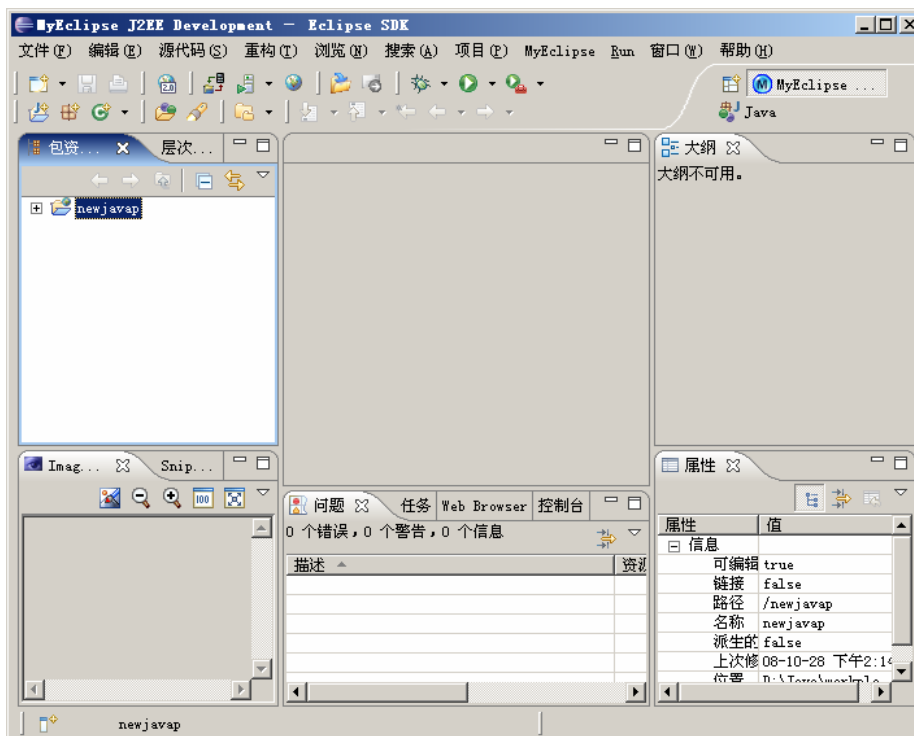


图 3-8 新建工程后的工作页面

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。

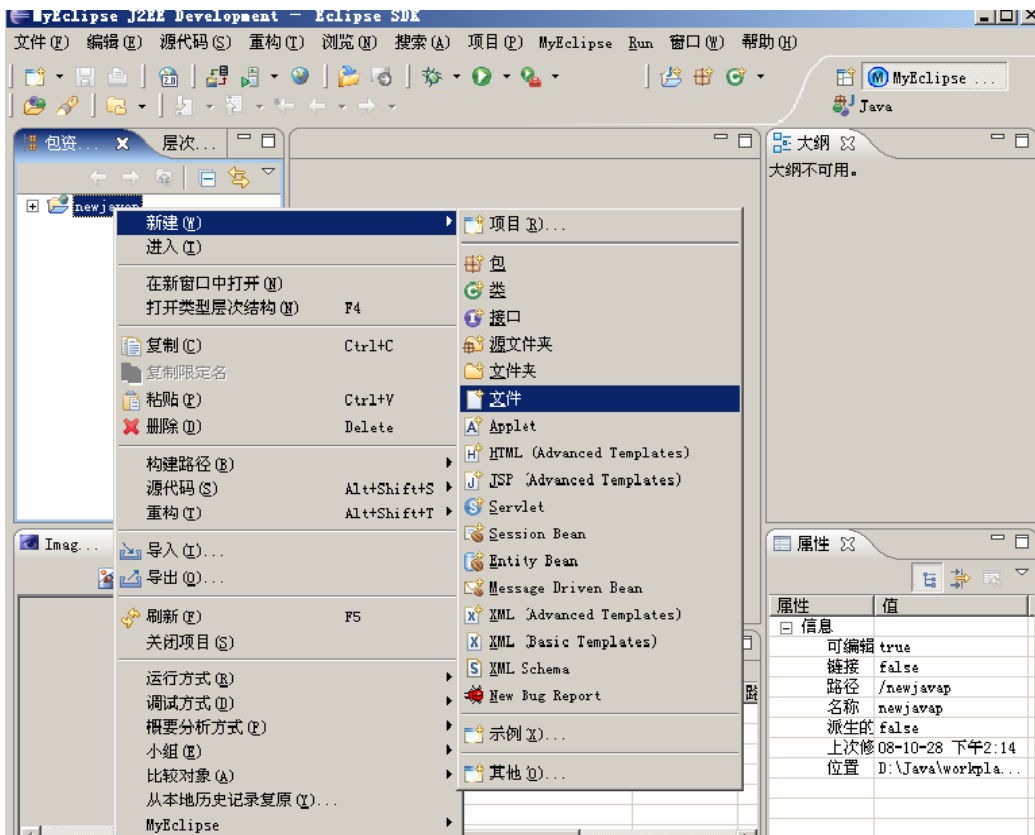


图 3-9 新建文件页面

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。

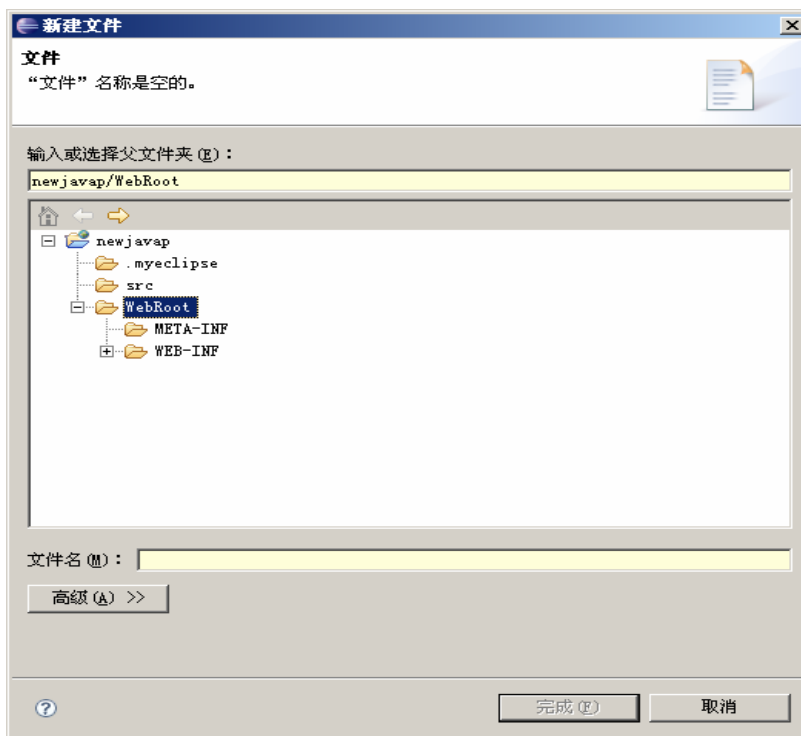


图 3-10 选择父文件夹对话框

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）。

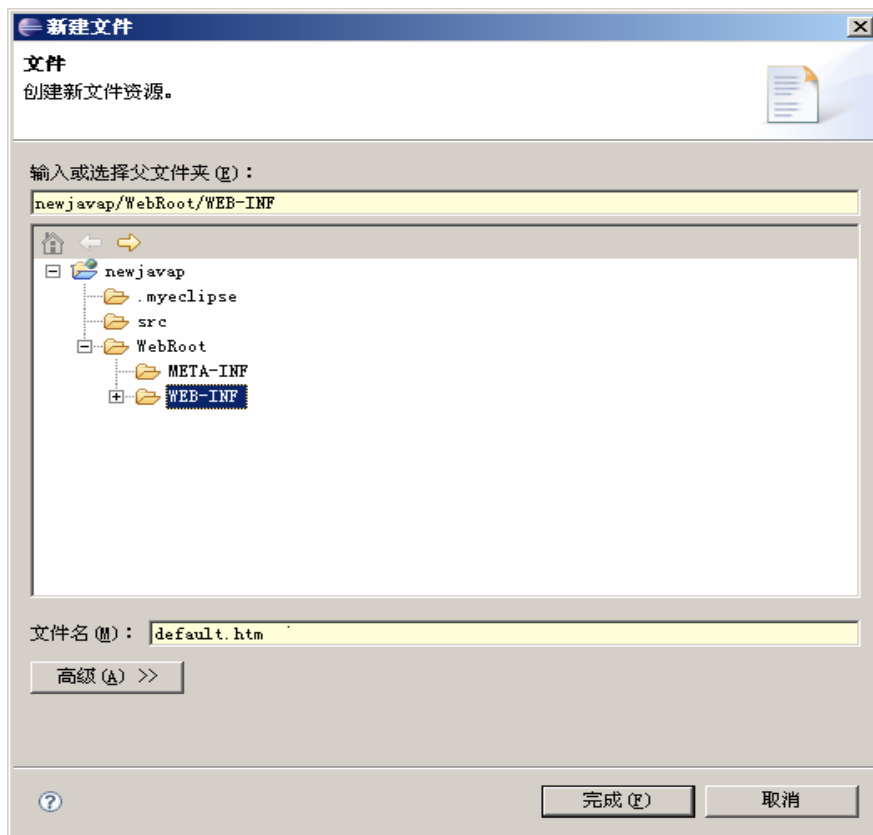


图 3-11 输入文件名

- 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

- 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）。

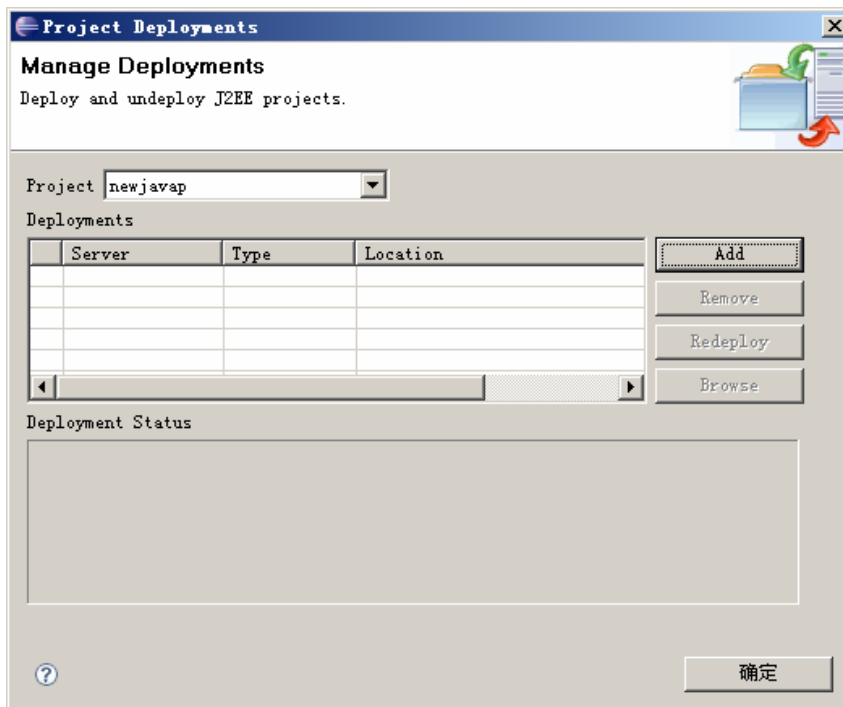


图 3-14 部署步骤一

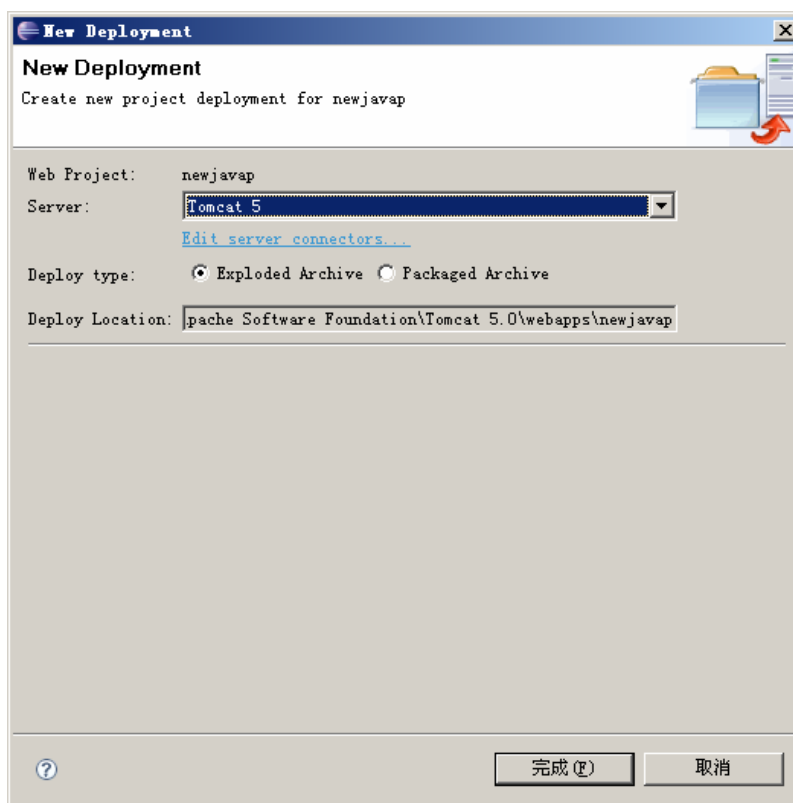


图 3-15 部署步骤二

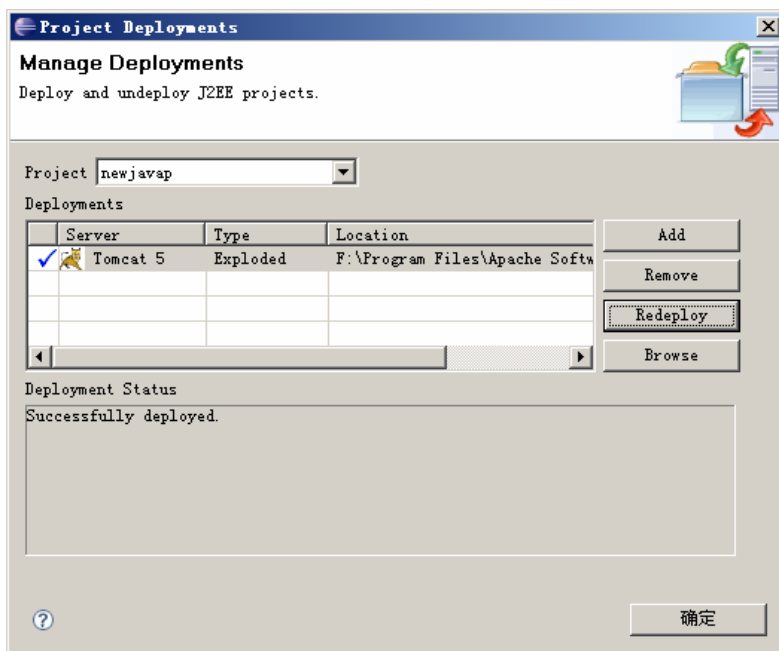




图 3-16 部署步骤三

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。

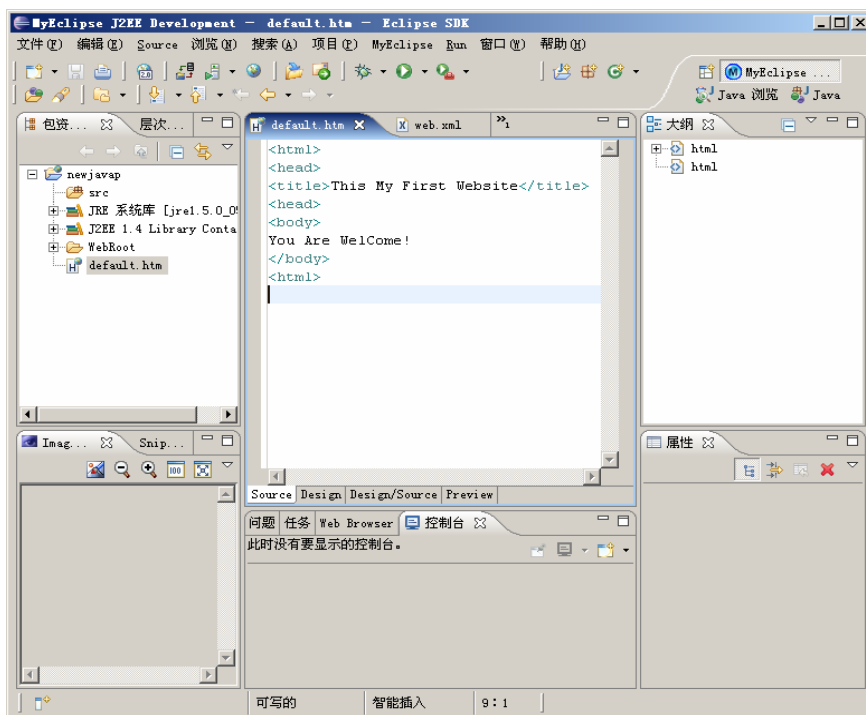


图 3-17 启动 Tomcat 服务器

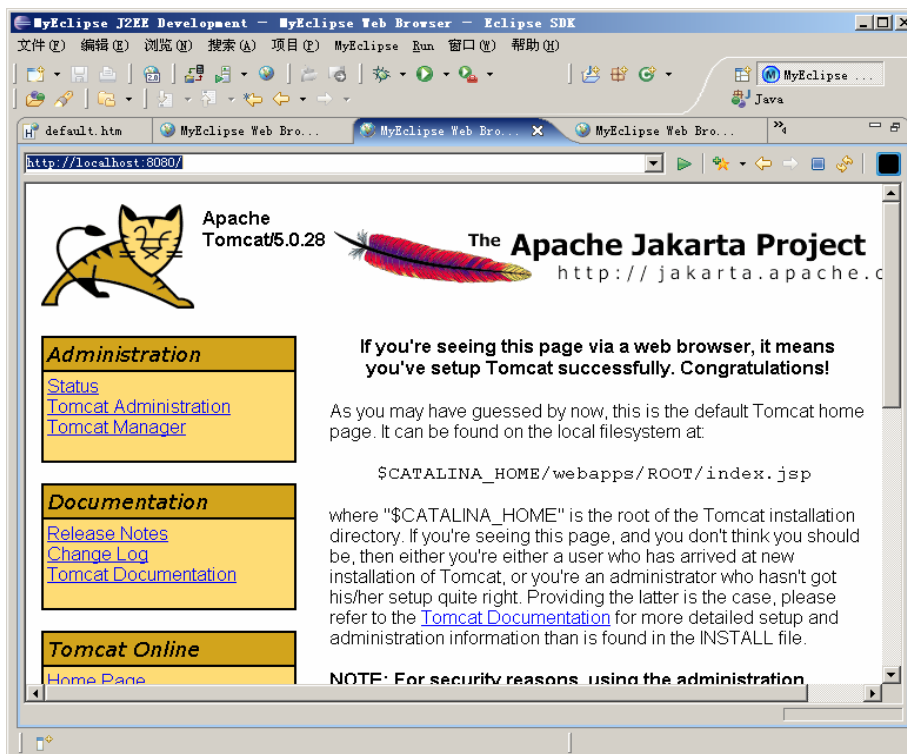


图 3-18 打开 Tomcat 首页

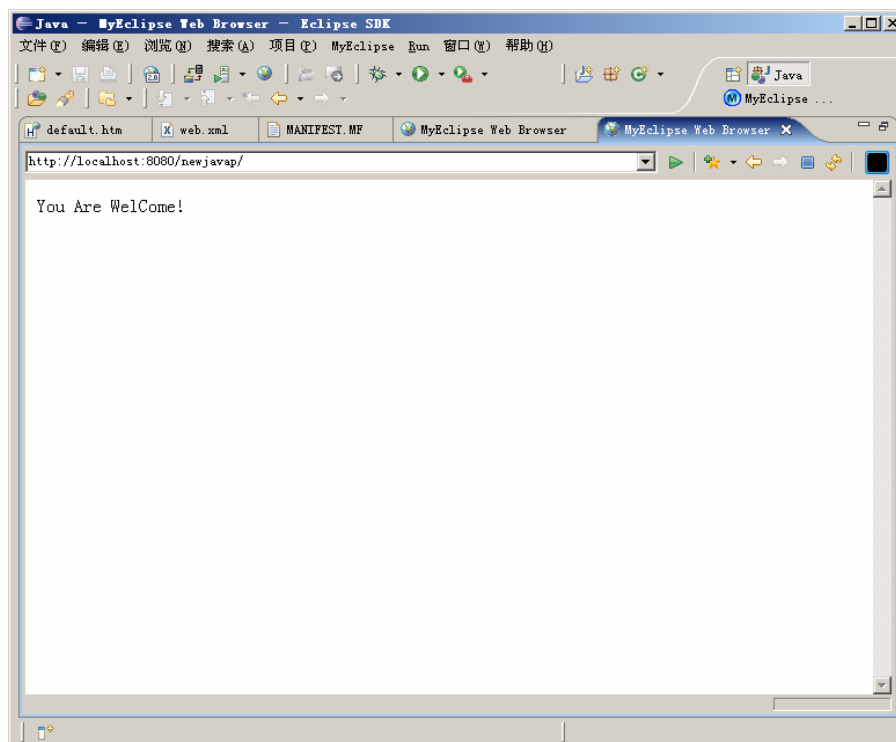


图 3-19 运行网站页面

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。

3 数字地球二次开发技术

3.1 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。

JavaScript 的核心对象如表 4-1 所示：

表 4-1 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 4-2 所示：

表 4-2 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 4-3 所示：

表 4-3 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	允许一个签名的脚本向其它签名或未签名的脚本提供属性、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名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 4-4 所示：

表 4-4 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 4-1 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 setOpeMode(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="";

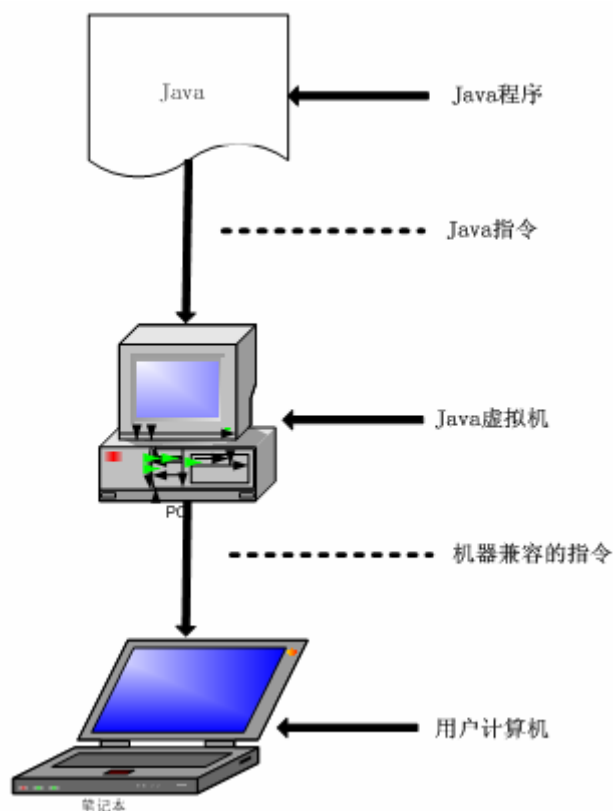
function onDisplayToolClick(con)
{
    var type = con
    switch(type)
    {
        case "reflash":
            map.refresh();
            break;
        case "Condition":
            var url = "VectorPage/SelectByCondition.aspx?rnd="+Math.random();
            renderSelectResult(url);
            break;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

图 4-1 JavaScript 示例

3.2 Java 语言

Java 语言是在 1995 年由 Sun Microsystems 公司提出的一种编程语言，它具有了革命化的“编写一次，随处运行”的跨平台能力，并具有强大的网络能力。

Java 之所以能够“编写一次，到处运行”，其原因就在于 Java 虚拟机 (Java Virtual Machine, JVM)。JVM 位于 Java 程序和用户的计算机系统之间，如下图所示。



所谓虚拟机，其实在现实中并不存在，而是计算机的一种抽象设计，是由计算机程序来实现的。Java 虚拟机，指的就是一种知道 Java 语言的各种指令，并能够将这些指令翻译成用户计算机所能理解的指令的计算机程序，也就是一种“抽象计算机”。

Java 语言在执行时既被解释又被编译，这是由 JVM 结构所决定的。用户编写的 Java 程序首先被编译成能在 Java 虚拟机上运行的虚拟指令，编译后的 Java 程序像用其他任何语言编写的程序一样，被当成可执行文件，不同的是，这种可执行文件必须运行于 JVM 上。JVM 在不同的系统中以同样的方式工作，因此 Java 程序不必为每种平台单独进行编译。JVM 为不同的平台解释 Java 程序，所有和移植有关的问题都在 JVM 中得到解决，这样用户的 Java 程序在理论上就可以不加修改地运行在任何拥有 JVM 的平台上。

Java 语言的特点：

- a) 简单 (Simple)
- b) 面向对象 (Object-oriented)
- c) 分布式 (Distributed)
- d) 健壮性 (Robust)
- e) 安全性 (Secure)
- f) 中性结构 (Architecture-neutral)
- g) 可移植 (Portable)
- h) 解释型 (Interpreted)
- i) 高性能 (High performance)
- j) 多线程 (Multithreaded)
- k) 动态 (Dynamic)

3.3 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 4-5 所示:

表 4-5 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 4-6 所示, 标签属性示例如图 4-2 所示:

表 4-6 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 4-2 标签属性示例

3.4 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。

数字地球中JSP技术指令：

表 4-7 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:

- 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。
- 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。
- 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。
- 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。
- 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。
 - 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。
 - 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。
 - 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 getMessage() 和 printStackTrace() 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。

3.5 数据库 (SQL Server/Oracle)

选择的数据库为: SQL Server 和 Oracle。这里采用的商用数据库是 SQL Server 2005。

数字球示例站点主要涉及到一个点位信息数据 JEarthDemo 数据库。JEarthDemo 数据库

是示例站点中所用到的数据库。数据库中的数据表包括 CityInfo 表（存放各个定位城市的信息），shapeInfo 表， UserLabel 表（存放用户添加的地标点、文字标注点、图标点信息）。

GIS 服务器层的平台是 Linux 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。

数据服务器的平台是 Linux 或 Windows/NT 以及地理数据库。其功能是读取地球的显示数据。客户端数据请求都由数据服务器获得，包括两部分数据：一部分是全球数据：基本数据、详细数据、国界数据、地名数据、高程数据；另一部分是：用户自裁剪的影像数据和高程数据。

系统中的数据可以采用文件系统（MAPGIS 的空间数据文件 HDF）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。

Hibernate 是一个面向 Java 环境的对象/关系数据库映射工具。对象/关系数据库映射(object/relational mapping (ORM))这个术语表示一种技术，用来把对象模型表示的对象映射到基于 SQL 的关系模型结构中去。

Hibernate 不仅仅管理 Java 类到数据库表的映射，还提供数据查询和获取数据的方法，可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间。Hibernate 的目标是对于开发者通常的数据持久化相关的编程任务，解放其中的 95%。

3.6 XML 语言

XML是Extensible Markup Language的简写，即可扩展置标语言。XML是Internet环境中跨平台的、依赖于内容的技术，是处理分布式结构信息的技术。XML是可扩展的，它是一种元标记语言，所谓“元标记”，就是开发者可以根据自己的需要定义自己的标记。

一个 XML 文档包含三个部分：

1. 一个 XML 文档声明；
2. 一个关于文档类型的定义；
3. 用 XML 标识创建的内容。

XML是一种能够让你自己创造标识的语言，它可以将数据与格式从网页中分开，它可以储存数据和共享数据的特性使得XML无所不能。

XML的语法有着严格的要求，对于XML文档有两层要求：格式良好的（Well-Formed）和有效的（Valid）。

格式良好的XML文档在创建时，必须满足一下几个准则，才是格式良好的：

1. 除非省略不写，否则XML文档的第一行必须是文档声明。
2. 在XML文档中有且只有一个根元素。
3. 在XML文档这个的标记必须正确关闭。

4. 在XML文档中标记之间不允许交叉。
5. 属性值必须要使用双引号或单引号括起来。
6. XML文档中的标记，指令和属性严格区分大小写。
7. 有效的XML文档。