### java.io.File类 路径

不同的操作系统对于文件系统路径的设置各有差异。

在windows中，一个路径的表示可能是: “c:\\workspace\\CH14\\”

而在linux中的路径设置可能会像是: “/home/justin/workspace/ch14”

windows的路径指定是使用UNC(Universal Naming Convention)路径名，以\\开始表示硬盘根目录。如果没有以\\开始表示相对于当前工作目录的路径，c是可选的硬盘指定，后面跟随者:字符。而UNIX-Like系统的路径指定以/开始表示从根目录开始的绝对路径，不以/开始表示相对于当前工作目录的路径。

java.io.File类提供一个抽象的、与系统独立的路径表示。给它一个路径字符串，它会将其转换为与系统无关的抽象路径表示，这个路径可以指向一个文件、目录或是URI(Uniform Resource Identifier)。

File.*separator 提供了在windows/unix-like 中通用的路径分隔符。*

一个File的实例被建立时，它就不能再被改变内容。

不管是文件或目录，在Java中都是以File的实例来表示的

### [相对路径与绝对路径](http://www.cnblogs.com/sutengcn/archive/2005/07/28/201993.html)

文档路径类型一共有三种：绝对路径、根相对路径和文档相对路径。

#### 绝对路径

绝对路径就是你的主页上的文件或目录在硬盘上真正的路径,常用形式有：

URL形式：包括服务器协议（如http协议）的完全路径，比如“洪恩在线——电脑乐园”，完全路径为：<http://www.hongen.com/pc/index.htm>，如果所要链接当前站点之外的文档，就必须使用绝对路径。

物理路径形式：C:\xyz\test.txt 代表了test.txt文件的绝对路径，

#### 相对路径

包括根相对路径（Site Root）和文档相对路径（Document）两种。

在本地磁盘上编辑网页时，需要选定一个文件夹来定义本地站点，模拟服务器上的根文件夹，系统就根据这个文件夹来确定所有链接的本地文件位置，而根相对路径中的根就是指这个文件夹。

根相对路径（也称相对根目录）的路径以“/”开头，路径是从当前站点的根目录开始计算。比如D盘myweb目录就是名为“myweb”的站点，这时“/index.htm”路径，就表示文件位置为D:\myweb\index.htm。根相对路径适用于链接内容频繁更换环境中的文件，这样即使站点中的文件被移动了，其链接仍可以生效。

如果目录结构过深，在引用根目录下的文件时，用根相对路径会更好些。比如某一个网页文件中引用根目录下img目录中的一个图片，在当前网页中用文档相对路径表示为：“../../../../../img/a.gif”，而用根相对路径只要表示为“/img/a.gif”就行了。

注意：在预览文件时，用根相对路径链接的内容在本地浏览器中不会显示出来，这是因为浏览器不承认站点的根文件夹为服务器。

这时我们可以将站点的根目录虚拟为一个盘。比如要浏览D:\myweb下的文件，可以选择Windows“开始”菜单中的“运行”，弹出运行对话框  
　　输入“SUBST I: d:\myweb”，D盘下的myweb文件夹就被虚拟为I盘，用浏览器查看I盘下网页，用根相对路径链接的内容都可以正常显示了。

文档相对路径就是指包含当前文档的文件夹，也就是以当前网页所在文件夹为基础开始计算路径。比如，当前网页所在位置为D:\myweb\mypic，那么：

　　“a.htm”就表示D:\myweb\mypic\a.htm；

　　“../a.htm”相当于D:\myweb\a.htm，其中“../”表示当前文件夹上一级文件夹。

　　“img/a.gif”是指D:\myweb\mypic\img\a.gif，其中“img/”意思是当前文件夹下名为img文件夹。

　　文档相对路径是最简单的路径，一般多用于链接保存在同一文件夹中的文档

### 关于JSP/Servlet中的相对路径和绝对路径

#### 服务器端的地址

服务器端的相对地址指的是相对于你的web应用的地址，这个地址是在服务器端解析的   
（不同于html和javascript中的相对地址，他们是由客户端浏览器解析的）也就是说这时候在jsp和servlet中的相对地址应该是相对于你的web应用，

即相对于http://192.168.0.1/webapp/的。   
  
其用到的地方有(以tomcat为例的话：tomcat的webapps下部署了两个项目Test1和Test2)：   
forward：servlet中的request.getRequestDispatcher(address);这个address是   
在服务器端解析的，所以，你要forward到a.jsp应该这么写：

request.getRequestDispatcher(“/user/a.jsp”)这个**/**相对于当前的web应用根路径，   
其绝对地址就是：http://192.168.0.1/Test/user/a.jsp。

sendRedirect：在jsp中<%response.sendRedirect("/Test2/user/a.jsp");%>

路径是相对于服务器的根路径，其绝对地址就是：<http://192.168.0.1/Test2/user/a.jsp>

#### 客户端的地址

所有的html页面中的相对地址都是相对于服务器根目录(http://192.168.0.1/)的，   
而不是(根目录下的该Web应用的目录)http://192.168.0.1/webapp/的。   
Html中的form表单的action属性的地址应该是相对于服务器根目录(http://192.168.0.1/)的，   
所以，如果提交到a.jsp为：action＝"/webapp/user/a.jsp"或

action="<%=request.getContextPath()%>"/user/a.jsp；

提交到servlet为actiom＝"/webapp/handleservlet"    
Javascript也是在客户端解析的，所以其相对路径和form表单一样。   
  
  
　　因此，一般情况下，在JSP/HTML页面等引用的CSS,Javascript.Action等属性前面最好都加上   
<%=request.getContextPath()%>,以确保所引用的文件都属于Web应用中的目录。   
另外，应该尽量避免使用类似"**.**","**./**","**../../**"等类似的相对该文件位置的相对路径，这样当文件移动时，很容易出问题。

### JSP/Servlet中获得当前应用的相对路径和绝对路径

在jsp和class文件中调用的相对路径不同。 在jsp里，根目录是WebRoot 。

在class文件中，根目录是WebRoot/WEB-INF/classes 当然你也可以用

System.getProperty("user.dir")获取你工程的绝对路径。

#### jsp中取得路径

以工程名为TEST为例：   
  
(1)得到包含工程名的当前页面全路径：request.getRequestURI()   
结果：/TEST/test.jsp  (url部分路径)

(3)得到工程名：request.getContextPath() 或

request.getServletContext().getContextPath()   
结果：/TEST   
(4)得到当前页面所在目录下全名称：request.getServletPath()   
结果：如果页面在jsp目录下 /TEST/jsp/test.jsp

(5)得到页面所在服务器的全路径：application.getRealPath("页面.jsp")   
结果：D:/resin/webapps/TEST/test.jsp

(6)得到页面所在服务器的绝对路径：

absPath=new java.io.File(application.getRealPath(request.getRequestURI())).getParent();   
结果：D:/resin/webapps/TEST

(7) request.getRealPath("/") 与 (request.getServletContext().getRealPath("/")

结果(eclipse下部署在tomcat中的项目)：

E:\wks\_study\.metadata\.plugins\org.eclipse.wst.server.core\tmp0\wtpwebapps\itcastweb\

#### 在Servlet中取得路径

(1)得到工程目录：request.getSession().getServletContext().getRealPath("") 参数可具体到包名。   
结果：E:/Tomcat/webapps/TEST   
(2)得到IE地址栏地址：request.getRequestURL()

如servlet(<url-pattern>/servlet/MyServlet/url-pattern>) :

结果：http://localhost:8087/itcastweb/servlet/MyServlet  
(4）根目录所对应的绝对路径:request.getServletPath();

（5）文件的绝对路径:

request.getSession().getServletContext().getRealPath (request.getRequestURI())     
（6）当前web应用的绝对路径 :servletConfig.getServletContext().getRealPath("/");   
(ServletContext对象获得几种方式：   
       javax.servlet.http.HttpSession.getServletContext()   
       javax.servlet.jsp.PageContext.getServletContext()   
       javax.servlet.ServletConfig.getServletContext()   
     )

### java 的Class中获得相对路径，绝对路径的方法

单独的Java类中获得绝对路径   
根据java.io.File的Doc文挡，可知: 默认情况下new File("/")代表的目录为：System.getProperty("user.dir")。   
已下方式获得执行类的当前路径   
package org.cheng.file;   
import java.io.File;   
  
public class FileTest {   
    public static void main(String[] args) throws Exception {        
  
  System.out.println(

Thread.currentThread().getContextClassLoader().getResource("")

);       
  System.out.println(FileTest.class.getClassLoader().getResource(""));          
 System.out.println(ClassLoader.getSystemResource(""));          
  System.out.println(FileTest.class.getResource(""));          
  System.out.println(FileTest.class.getResource("/")); //Class文件所在路径    
  System.out.println(new File("/").getAbsolutePath());          
  System.out.println(System.getProperty("user.dir"));      
 }

}

### 服务器中的Java类获得当前路径

(1).Weblogic   
WebApplication的系统文件根目录是你的weblogic安装所在根目录。   
例如：如果你的weblogic安装在c:\bea\weblogic700.....   
那么，你的文件根路径就是c:\.   
所以，有两种方式能够让你访问你的服务器端的文件：

a.使用绝对路径：   
比如将你的参数文件放在c:\yourconfig\yourconf.properties，   
直接使用 new FileInputStream("yourconfig/yourconf.properties");

b.使用相对路径：   
相对路径的根目录就是你的webapplication的根路径，即WEB-INF的上一级目录，将你的参数文件放   
在yourwebapp\yourconfig\yourconf.properties，   
这样使用：   
new FileInputStream("./yourconfig/yourconf.properties");   
这两种方式均可，自己选择。   
  
(2).Tomcat   
在类中输出System.getProperty("user.dir");显示的是%Tomcat\_Home%/bin   
  
(3).Resin   
不是你的JSP放的相对路径,是JSP引擎执行这个JSP编译成SERVLET   
的路径为根.比如用新建文件法测试File f = new File("a.htm");   
这个a.htm在resin的安装目录下   
  
(4).如何读相对路径哪？   
在Java文件中getResource或getResourceAsStream均可   
例：**getClass().getResourceAsStream(filePath)**;

//filePath可以是"**/filename**",这里的/代表web 发布根路径下WEB-INF/classes   
默认使用该方法的路径是：WEB-INF/classes。已经在Tomcat中测试。

### Java中getResourceAsStream用法

#### 基础用法

InputStream ips = **new** FileInputStream("config.properties");

//也可以用类加载器来管理资源和配置文件

//针对项目根路径,默认则是从ClassPath根下获取，path不能以’/'开头，最终是由ClassLoader获取资源。

//InputStream ips = ReflectTest2.class.getClassLoader().getResourceAsStream("cn/itcast/day1/config.properties");

//path 不以’/'开头时默认是从此类所在的包下取资源，以’/'开头则是从ClassPath根下获取。

//ClassPath路径是：WEB-INF/classes。已经在Tomcat中测试

//其只是通过path构造一个绝对路径，最终还是由ClassLoader获取资源。

//InputStream ips = ReflectTest2.class.getResourceAsStream("resources/config.properties");

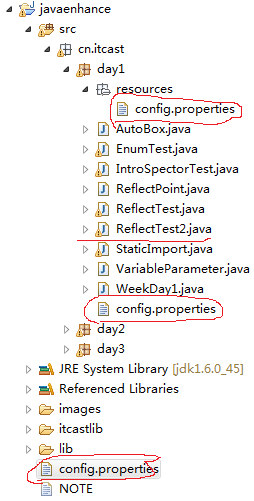
Properties props = **new** Properties();

props.load(ips);

ips.close(); //把所用的操作系统的资源干掉，而自身会被jvm垃圾回收器回收

//上面代码在ReflectTest2类中，

config.properties配置文件在eclipse工程中的结构如下图：



#### 详解

**首先，Java中的getResourceAsStream有以下几种：**   
1. Class.getResourceAsStream(String path) ： path 不以’/'开头时默认是从此类所在的包下取资源，以’/'开头则是从ClassPath根下获取。其只是通过path构造一个绝对路径，最终还是由ClassLoader获取资源。   
2. Class.getClassLoader.getResourceAsStream(String path) ：默认则是从ClassPath根下获取，path不能以’/'开头，最终是由ClassLoader获取资源。   
3. ServletContext. getResourceAsStream(String path)：默认从WebAPP根目录下取资源，Tomcat下path是否以’/'开头无所谓，当然这和具体的容器实现有关。  
4. Jsp下的application内置对象就是上面的ServletContext的一种实现。   
**其次，getResourceAsStream 用法大致有以下几种：**   
第一： 要加载的文件和.class文件在同一目录下，例如：com.x.y 下有类me.class ,同时有资源文件myfile.xml   
那么，应该有如下代码：   
me.class.getResourceAsStream("myfile.xml");   
  
第二：在me.class目录的子目录下，例如：com.x.y 下有类me.class ,同时在 com.x.y.file 目录下有资源文件myfile.xml   
那么，应该有如下代码：   
me.class.getResourceAsStream("file/myfile.xml");

第三：不在me.class目录下，也不在子目录下，例如：com.x.y 下有类me.class ,同时在 com.x.file 目录下有资源文件myfile.xml   
那么，应该有如下代码：   
me.class.getResourceAsStream("/com/x/file/myfile.xml");   
总结一下，可能只是两种写法   
第一：前面有 “   / ”   
“ / ”代表了工程的根目录，例如工程名叫做myproject，“ / ”代表了myproject   
me.class.getResourceAsStream("/com/x/file/myfile.xml");   
第二：前面没有 “   / ”   
代表当前类的目录   
me.class.getResourceAsStream("myfile.xml");   
me.class.getResourceAsStream("file/myfile.xml");   
**最后，自己的理解：**   
getResourceAsStream读取的文件路径只局限于工程的源文件夹中，包括在工程src根目录下，以及类包里面任何位置，但是如果配置文件路径是在除了源文件夹之外的其他文件夹中时，该方法是用不了的。

### request.getRequestURI()代码示例

<%  
String url = request.getRequestURI();  
if(url.endsWith("index.jsp")){  
 response.sendRedirect(request.getScheme()+"://"+request.getServerName()+"/");  
}  
%>

这段是什么意思？

jsp 重定向 如果浏览器中的网址以index.jsp结尾 则跳转到servername（）接收到的路径。

getRequestURI() 返回值类似：/mypage.jsp ，

getScheme() 返回协议的名称http，和"://" 拼起来 http:// getServerName() 这是获取服务器的名称如localhost 意思就是，如果请求以index.jsp结尾，就跳转到网站的首页