J2EE 平台下 WEB 应用国际化的研究与实现

Research and Implementation on Internationalization of WEB Application under J2EE platform

周中雨　（中国民航信息网络股份有限公司　100710）
吕琳琳　（河北沧州职业技术学院　061000）

摘要：为了使企业 WEB 应用能够支持全球客户，软件开发者应该开发出支持国际化的 WEB 应用。本文讨论了 J2EE 平台下 WEB 应用国际化中的字符编码问题，并给出了比较完整的解决方案。

关键词：J2EE　WEB 应用　国际化　反射

1　概述

随着全球经济的一体化成为一种主流趋势，如今的企业已经不再仅仅着眼于本国市场，而是开始致力于在全球范围内为产品寻找客户。Internet 的迅猛发展推动了跨国业务的发展，它成为一种在全世界范围内发布产品信息、吸引客户的有效手段。为了使企业 WEB 应用能够支持全球客户，软件开发者应该开发出支持多国语言、国际化的 WEB 应用。

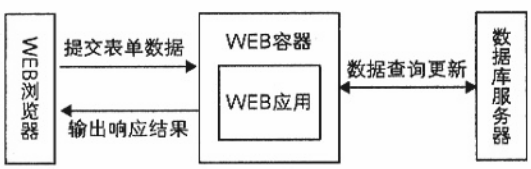

图 1　J2EE 平台下 WEB 应用输入输出流

国际化指的是在软件设计阶段就应该使软件具有支持多种语言和地区的能力。这样，当需要在应用中添加一种新的语言和国家的支持的时候，不需要对已有的软件返工，无需修改应用的程序代码。

支持国际化的 WEB 应用应具备以下特征：

（1）当应用需要支持一种新的语言时，可以方便快捷的对应用进行调整，无需修改程序代码；

（2）页面中的数据（如文本、图片、JavaScript 提示信息等），能够根据用户的语言和国家设置正确显示；

（3）根据用户的语言和国家设置，对与特定文化相关的数据，如日期、时间和货币等，能够正确格式化。

在 WEB 应用的国际化问题中，不可避免的涉及到字符编码转换问题。首先我们对 J2EE 平台下字符编码转换问题进行研究分析。

2　编码问题

J2EE 平台下，WEB 应用的各种输入输出流如图 1 所示。

假定 WEB 浏览器采用的字符编码方式为 A，WEB 容器采用的编码方式为 B，数据库服务器采用的编码方式为 C。

下面分析各个环节中编码方式对数据处理的影响。

2.1　提交表单数据

在提交表单数据环节，默认情况下，IE 浏览器发送请求时采用 iso－8859－1 字符编码，如果 WEB 应用程序要正确地读取用户发送的其他字符编码的数据，则需要进行编码转换。

一种方法是在处理请求前，先设置 HttpServletRequest 对象的字符编码：

```
request. setCharacterEncoding( " B" ) ;//设置请求编码为 B
```

另一种方法是对用户输入的请求数据进行编码转换：

```
String s =  request. getParameter( " input" ) ;//取得输入参数
if ( s!  =null)
s = new String( s. getBytes( " A" ) , " B" ) ;//输入数
```


©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

据编码方式由 A 转换为 B

2.2 数据库查询更新

在读取数据库中数据时，如果数据库字符编码与 WEB 容器相同，则无需进行进行编码转换。如果不同，则必须先对来自数据库的数据进行编码转换，然后才能使用。例如：

```
String s = rs.getString("field");//取得列 field 中的字符串数据
String str = new String(s.getBytes("C"),"B");//数据编码方式由 C 转换为 B
```

在更新数据库数据时，数据库将对数据进行编码转换，这样可能造成数据库中的数据出现乱码。

2.3 输出响应结果

对于响应结果，可以通过在 servlet、jsp 或 html 文件中设置 charset 属性来设置响应结果的编码。例如在 servlet 中，示例代码如下：

```
response.setContentType("text/html;charset=A");//设置响应结果编码为 A
```

这样，WEB 浏览器根据设定的编码方式进行显示，如果设置的编码方式与 WEB 容器的编码方式不一致，会造成乱码。

根据以上分析，不一致的编码方式会增大开发的工作量，并可能造成数据乱码。在系统中以统一的编码方式处理所有字符是从根本上解决编码问题的一种简便且有效的方案。目前，Unicode 可以处理所有目前存在的任何语言文字，但是所有的字符都以双字节形式存储，浪费了大量空间。UTF－8 编码逐步成为一种占主导地位的编码方法，它与 ASCII 完全兼容，且变长编码方式节省大量空间。JAVA 编译后产生的 class 文件也采用 UTF－8 编码。因此，我们采用 UTF－8 作为系统所有数据的统一的编码方式，包括代码、资源文件和数据库[1]。

3 实现技术

下面介绍 J2EE 平台下 WEB 应用国际化实现的关键技术。

3.1 请求数据的编码设置

调用 HttpServletRequest 的 setCharacterEncoding("UTF－8")方法，能够把用户的请求数据的字符编码设为 UTF－8。WEB 应用的输出也全部采用 UTF－8 编码，这样，WEB 应用的输入和输出都采用同一种编码，就无需在程序中进行编码转换。设置请求数据编码的功能可以由 servlet 过滤器来完成。因为它可以预处理所有的 http 请求，这样能避免在每个 servlet 中设置请求数据编码，如图 2 所示。

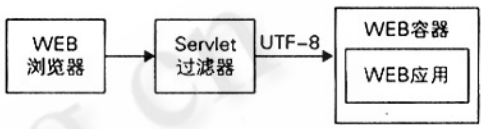

图 2 过滤器设置请求编码

实现过滤器的主要代码包括：

```
//从 web.xml 配置中读取编码类型参数
private String defaultEncode = "UTF－8";
public void init(FilterConfig config) throws ServletException {
    if(config.getInitParameter("Charset")! = null){
    defaultEncode = config.getInitParameter("Charset");
    }
}
//在 Filter 主程序中设置 request 编码
public void doFilter(ServletRequest request, ServletResponse response, FilterChain chain) throws IOException, ServletException {
    ServletRequest srequest = request;
    srequest.setCharacterEncoding(defaultEncode);
    chain.doFilter(srequest,response);
}
```

需要在 web.xml 增加 filter 配置，过滤器才能发挥作用。在配置中初始化编码类型参数，过滤器按照此参数来对请求进行编码。

```
<filter>
    <filter－name> Encoding </filter－name>
    <filter－class> com.travelsky.web.filter.CharsetFilter </filter－class>
    <init－param>
    <param－name> Charset </param－name>
```


©《计算机系统应用》编辑部 http://www.c-s-a.org.cn

```
    <param-value>UTF-8</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>Encoding</filter-name>
    <servlet-name>action</servlet-name>
</filter-mapping>
```

3.2 文本数据的国际化

为了实现 WEB 应用的国际化，需要将静态文本数据与程序代码分离，采用独立的资源文件保存这些数据。对于每一个语言的版本，和语言相关的内容保存在相应的资源文件中。每增加一种语言的版本，只要增加相应的资源文件即可。

JAVA 在其核心库中提供了支持国际化的类和接口，主要包括 Locale 类和 ResourceBundle 类[2]。最常用的 WEB 开发框架 Struts 就依赖于这些 JAVA 组件来实现对国际化的支持。

Jsp 文件中不应该直接包含本地化的消息文本，而应该通过 <bean:message> 标签从资源文件中获取静态文本数据。在资源文件中，所有的字符也应采用 UTF-8 编码。

3.3 数据库数据的国际化

在数据库中，数据表的主键应使用英文字符串或数字，这样在不同语言下都能查询得到正确的数据。

对于基础数据，如航空公司代码对应的名称，在数据表中对应多个列，列名中包含语言信息。例如航空公司默认名称对应列 AirlineName，英文名称对应 AirlineName_en，简体中文名称对应列 AirlineName_zh，繁体中文名称对应列名为 AirlineName_zh_tw。

基础数据表对应的 JAVA 对象，其属性名应对应数据表列名。这样可以利用 JAVA 的反射机制，通过统一的接口获得语言相关的航空公司名称，代码如下：

```
public static String localize(String localeString, Object obj, String method){
    Class cla=obj.getClass();//取得 Class 对象
    Method m=cla.getMethod(method+localeString,null);//取得与语言信息相关的方法
    String str=String.valueOf(m.invoke(obj,null));//执行方法，结果为 String
    if(str==null) str="";
    return str;
}
```

接口中，localeString 为标识语言信息的字符串，如"_en"。obj 为 JAVA 对象，method 为获取默认属性的方法名，此接口返回 String 对象。

应测试，此方法执行 10 万次的时间小于 1 秒钟，执行效率可以得到保证。

3.4 日期、数字、货币的国际化

在不同的国家和地区，日期、数字、货币的表示方法也不同。JAVA 在其核心库中提供了支持日期、数字、货币的国际化支持。支持日期国际化的类主要是 SimpleDateFormat，支持数字国际化的类主要是 NumberFormat，支持货币国际化的类主要是 Currency。例如下面的代码可以对货币进行国际化：

```
double amount = 5000.25;
NumberFormat usFormat = NumberFormat.getCurrencyInstance(Locale.US);//设置国家为美国
System.out.println(usFormat.format(amount));
NumberFormat german =
NumberFormat.getCurrencyInstance(Locale.GERMANY);//设置国家为德国
System.out.println(german.format(amount));
```

程序执行结果为：

$5,000.25

5.000,25

从结果可以看出，程序对数字和货币进行了正确的国际化。

4 结束语

针对 J2EE 平台下的 WEB 应用，本文提出了比较完整的国际化解决方案。当然，国际化涉及的问题还有很多，需要继续深入研究。本文提出的解决方案已经应用于一个为航空代理人提供机票查询预订服务的网站，并取得很好的效果。

参考文献

1 许晖、李涓子，J2EE 系统国际化问题的解决方案[J]，计算机工程，2005，18：79-80。

2 李华宇，JAVA 的国际化和本地化原理及解决方法[J]，微型机与应用，2001，11：31-32。

©《计算机系统应用》编辑部 http://www.c-s-a.org.cn