

solaris上开发J2EE应用中文问题的解决 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/289/2021_2022_solaris_E4_B8_8A_c104_289953.htm 关于JSP和J2EE的中文问题和处理方法，

网上已经有很多文章。一般在中文内核的操作系统（如中文NT，windows2000）上，该问题不是很突出，而在其他一些系统（如linux,solaris）中就比较明显了。近日在solaris上开发一个J2EE的应用系统，环境是solaris 2.7 minij2ee mysql。系统在windows和linux下均没有出现中文处理的问题，但部署到solaris上面后所有中文显示为?。后来经过分析，发现原因出在系统编码上。由于安装solaris时默认的系统编码为ASCII，因此以默认的系统编码处理字符串时汉字高位信息丢失。

下面一段简单的jsp程序说明了这个问题：

```
String  
str=request.getParameter("i").getBytes("iso-8859-1").  
byte[] b=str.getBytes(). out.println(new String(b,"gb2312")). %> 在  
浏览器中输入foo.jsp?i=中文，结果显示为??。如果将byte[]  
b=str.getBytes().换成上面注释掉的byte[]
```

```
b=str.getBytes("iso-8859-1").，则正常显示出“中文”二字。查  
阅了mysql JDBC的驱动程序，问题相同。考虑解决的方法有  
两个，一个是修改JDBC驱动程序，另一个是将汉字编码成7  
位，从实现方便的角度选择了后者。不过后者的缺点是字符  
串长度增加，并且无法直接通过sql工具来修改数据库了。网  
上有一种汉字编码的方法，是将汉字高位去1，英文则补一  
个0表示。这种方法有缺陷，因为特定的汉字编码后会出现“  
”等SQL语句中有歧义的字符，导致sql失败。我摘取
```

了minij2ee中uniString的编码方法，该方法将字节表示为其16

进制编码，下面是源代码：

```
public String encode() { try {
StringBuffer sb=new StringBuffer(). byte[]
bytes=m_enc.compareTo("")==0?m_str.getBytes():m_str.getBytes(
m_enc). for(int i =0;i { char ch.
ch=Character.forDigit((bytes[i]>>4)amp.0xF,16). sb.append(ch). }
return sb.toString(). }
catch(java.io.UnsupportedEncodingException e) { throw new
RuntimeException("Unsupported encoding type."). } } public void
decode(String encodestr) { StringBuffer sb=new StringBuffer(). int
i=0. while(i!=encodestr.length()) {
sb.append((char)Integer.parseInt(encodestr.substring(i,i 2),16)). i
=2. } m_str=new uniString(sb.toString(),"iso-8859-1").cvt(m_enc).
} 使用编码后，问题解决。 另外提一下，minij2ee最新版本中
提供了一个uniString类，解决了在所有操作系统上的中文问题
。使用uniString对象，无需关心字符串本身编码，使用时只要
调用函数来获得需要的编码即可，如在jsp里调用uniString.gb()
即可以以gb2312输出字符串，在数据库存储时调
用uniString.iso()即可以以iso-8859-1编码输出字符串，无论在
中文内核还是英文内核的操作系统上均通用。 100Test 下载频
道开通，各类考试题目直接下载。详细请访问
www.100test.com
```