关于 PB 里乱码和 pbidea 里使用 utf8 的那些事

对于一直从事 pb 开发的程序员而言，平时从事的基本上现有系统维护，大多数是 CURD 工作。以前只是在 windows 平台上开发 pb 系统，后台也是 windows 平台上的 sql server 数据库，一切顺利。随业务发展，需要使用到跨 平台业务时，不可避免出现“乱码”现象，这时候就很慌乱，不知道该如何处理。

在剖析原因前，我们要初步了解一个基本概念：字符集！

（PB9 及以下版本以下简称 PB9，PB10 及以上版本以下简称 PB10，PB9 和 PB10 使用了不同的字符集编码，这是 PB 字符集编码的一个里程碑）

常用字符集说法都比较混乱：ansi GBK gb2312 utf16 utf8 unicode ……咱们不管它有多少种说法，我们只要明白：

1.PB9 使用的是 ansi 编码，一个汉字由 2 个字符组成。每个字符就是我们最初学的 ascii 编码，每个字符编码范围是 0-255。

2.PB10 使用的是 utf16 编码，通常我们也称之为 unicode 编码，一个汉字由一个字符组成，unicode 编码字符范围是 0-65535。

3.当我们使用一些和接口有关的功能时通常默认使用 utf8 编码。不管是 PB9，还是 PB10 都不能直接使用 utf8 编码，都需要转换，而且 pb9 和 pb10 转 utf8 的代码也不一样，因此 pb9 要是升级 pb10，这部分代码要重写。

为什么会出现乱码现象？通常导致乱码的原因有 3 个：

一．错误使用了 pb 函数或转换功能

1.例如：PB9 里：

```
messagebox("",mid("当前目录及其子目录",2))
```

必然会出现乱码。原因出在错误地使用 mid 函数身上。参数 2 也许你认为是从“前”开始取，实际上不是，每个汉字 2 个字符组成，实际上是从“当”的第二部分开始取，取了半个汉字。那你又想当然地认为：那也只是“当”字变成乱码，为什么后面全部乱码了呢？很简单，系统会把相邻的两个字符组成一个汉字，也就是说“当”的第二部分和“前”的第一部分组成一个汉字，“前”的第二部分和“目”的第二部分组成一个汉字……以此类推，当然就面目全非了。

正确用法应该是这样的：

```
messagebox("",midW("当前目录及其子目录",2))
```

2.例如：PB10 里：

PB10 里，string(…) blob(…) 函数，都有了第 2 个参数 encoding：

```
EncodingANSI!
EncodingUTF8!
```

EncodingUTF16LE (default)

EncodingUTF16BE

这个参数用于字符集编码转换，不能正确使用编码转换就必然会出现乱码。

二．未能正确配置数据库服务端和客户端字符集

Oracle、mysql、sqlite 等都会碰 到这个问题。以 Oracle 为例：

如果服务端：NLS_LANG=SIMPLIFIED CHINESE_CHINA.UTF8

而客户端是：NLS_LANG=SIMPLIFIED CHINESE_CHINA.ZHS16GBK

那么恭喜你：你所有的汉字出来，都会是乱码。

正确方法：客户端也必须是 NLS_LANG=SIMPLIFIED CHINESE_CHINA.UTF8，与服务端严格保持一致。

三．在各种接口使用中没有认真阅读开发文档，没有正确进行字符集处理

拿到一个开发任务，首先要仔细阅读开发文档。如果无特殊说明，**通常接口开发中都默认 utf8 字符集**，与接口有关的加密、签名之类，也都要使用 utf8 字符集。

PB9 使用各种转换 DLL，或者调用 winapi 直接转换，PB10 中可以使用 blob 和 string 进行转换，具体可以百度查询。

有一点必须注意：PB9 里使用 unicode 和 utf8，PB10 里使用 ansi 和 utf8，都不能使用 string 类型保存，只能使用 blob 类型保存。Pb9 里的 string 只能是 ansi 编码，pb10 里的 string 只能是 unicode 编码。切记切记！！！

四．Pbidea 中如何使用 utf8 进行接口开发

（一）显式转换

上面说到过，“pb9 和 pb10 转 utf8 的代码也不一样，因此 pb9 要是升级 pb10，这部分代码要重写”。Pbidea 里提供了转换函数，确保代码一致性。

由于 utf8 只能保存在 blob 类型里面，所以，pbidea 里许多 blob 类型的参数是传 utf8 字符串，而不是直接使用 blob(…)函数转换！！！

```
Blob lb_utf8

String ls_text

Uo_crypto c

Ls_text = “接口开发”

C = create uo_crypto
```

Lb_utf8 = c.toUtf8(ls_text) //string 转 utf8

Messagebox(“”,c.fromUtf8(lb_utf8)) //utf8 转 string

上面的代码，在 PB9 和 PB10 上不需要做任何改变，结果一致。

在进行加密、解密、签名时，通常我们的数据需要显式转换。

例如 base64

Uo_crypto c

C = create uo_crypto

Blob lb_base64

Lb_base64 = c. Base64Encode (c.toUtf8(“文本”))

例如 取 sha256

Uo_crypto c

C = create uo_crypto

Blob lb_sha256

Lb_sha256 = c.hash(c.toUtf8(“文本”),“sha256”)

例如 aes/128/ecb 加密

Uo_crypto c

C = create uo_crypto

Blob lb_aes

lb_aes = c .crypto(c.ToUtf8(“文本”),key,1,“aes_128_ecb” ,“”, 1)

（二）隐式转换

Pbidea 里许多内部可以直接自动转成 utf8 编码。

1.uo_json 内部实际处理编码是 utf8

Uo_json js;js = create uo_json

……

Lb_json = Js.toUtf8() 可得到 utf8 编码数据

2.uo_httpclient uo_curl

作为 http 接口，可以默认直接使用 utf8。

（1）凡是 blob 类型的参数，都传 utf8 类型的 blob 数据。

（2）凡是 string 类型的参数，如果其后有 boolean toUTF8 参数，把这个参数设置为 true，内部会自动 转换为 utf8 类型

（3）subroutine SetUtf8(boolean utf8) 函数，可以设置默认使用 utf8，所有 string 类型参数都会默认被转换成 utf8 字符集。

（4） 当接口默认是 utf8 字符集时，uo_response.data 默认也是 utf8 字符集的字符串，

Uo_crypto c; c = create uo_crypto

Messagebox(“”,c.fromUtf8(uo_httpclient.response.data)) 即可得到 string 字符串。

在接口开发中，涉及到加密、解密、签名、httpclient，当你测试有问题时，当你的结果与在线结果或 postman 不一致时，你要思考一下：你使用 utf8 了吗？

大自在 QQ:781770213 2023/1/9