

个人资料



TJU355

访问： 100664次

积分： 1360

等级： **BLOG > 4**

排名： 第13482名

原创： 28篇    转载： 57篇

译文： 0篇    评论： 42条

文章搜索

文章分类

心情空间 (6)

嵌入式linux (39)

科研那些事 (8)

RTP 多媒体传输 (6)

多线程编程 (2)

人生智慧 (2)

QT (20)

操作系统 (2)

数据库 (0)

文章存档

2012年10月 (2)

2012年09月 (1)

2012年08月 (1)

2012年06月 (17)

2012年05月 (2)

展开

阅读排行

关于qt中的tr ( ) 函数 (14305)

QT 中添加资源图片 (5471)

浅谈 Qt 布局管理器 如何 (4324)

通信类重要学术会议以及 (4028)

QT 知识点总结 (3162)

H264 I帧 P帧 (3151)

QT 鼠标右键菜单 (2979)

从零开始掌握iOS8开发技术（Swift版）

那些年我们追过的Wrox精品红皮计算机图书

CSDN学院--学习礼包大派送

CSDN JOB

带你坐飞机回家过年

## 关于qt中的tr ( ) 函数

分类： QT

2012-02-12 20:40

14309人阅读

评论(4)

收藏

举报

qt

encoding

object

语言

c

windows

目录(?)

[+]

在论坛中漂，经常遇到有人遇到tr相关的问题。用tr的有两类人：

- (1)因为发现中文老出问题，然后搜索，发现很多人用tr，于是他也开始用tr
- (2)另一类人，确实是出于国际化的需要，将需要在界面上显示的文件都用tr包起来，这有分两种：
  - (2a) 用tr包住英文(最最推荐的用法，源码英文，然后提供英文到其他语言的翻译包)
  - (2b) 用tr包住中文(源码用中文，然后提供中文到其他语言的翻译包)

注意哦，如果你正在用tr包裹中文字符，却不属于(2b)，那么，这是个信号：

- 你在误用tr
- 你需要的是QString，而不是tr

如果你确实属于(2b)，请做好心理准备，你可能还会遇到很多困难,请考虑Qt国际化（源码含中文时）的点滴分析

tr 是做什么的？下面二者的区别是什么？

```
QString text1 = QObject::tr("hello"); QString text2 = QString("hello");
```

tr是用来实现国际化，如果你为这个程序提供了中文翻译包(其中hello被翻译成中文"你好")，那么text1的内容将是中文"你好"；如果你为程序提供且使用日文翻译包，那么text1的内容将是日文。

tr是经过多级函数调用才实现了翻译操作，是有代价的，所以不该用的时候最好不要用。

## 关注的对象

本文关注的是tr或translate中包含中文字符串的情况：

- QObject::tr()
- QApplication::translate()
- QTextCodec::setCodecForTr

这个问题本多少可说的。因为涉及到的编码问题和QString 与中文问题中是完全一样的，只不过一个是用的setCodecForCStrings一个用的是setCodecForTr。

简单回顾QString的中文问题

- QString 采用的unicode，在中文支持上不存在任何问题
- “我是中文” 这是传统的 const char \* 的窄字符串
- 当将窄字符串赋值到QString时，我们需要告诉它我们的窄串采用的何种编码(gbk?、utf-8?)
- 究竟何种编码主要取决于我们的源代码文件的编码(windows上一般是gbk，其他平台一般utf-8)

例子：

QT creator 设计创建工具 (2488)

Qt 地图浏览器：使用QGLWidget (2340)

ARM11硬件编解码例程，实现视频播放 (2234)

评论排行

关于qt中的tr（）函数

(4)

某同学和导师对话

(3)

Qt Creator添加资源

(3)

RTP实时传输协议学习

(2)

【十大堕落的表现】如果

(2)

学习Linux网络编程(1)

(2)

视频终于可以传了，happy

(2)

Qt 地图浏览器：使用QGLWidget

(2)

gcc -l参数和-L参数 讲得

(2)

终于实现了S3C6410 video capture

(2)

推荐文章

\* Nginx 中 HTTP 模块初始化

\* 3D语音天气球（源码分享）——完结篇

\* 2014年终总结——我的匆匆这一年

\* Linux系统中修改用户名的两种方案整理

\* C++、Java、JavaScript中的异常处理(Exception)

最新评论

关于qt中的tr（）函数

zq4132: 非常感谢分享，谢谢。正在学习中

期待第1111次访客

zhyl724: 我要apad

ARM11硬件编解码例程,实现视频播放

zhanglu1985: 东西不错！谢谢支持！

一些常用的SCI论文句式 (1)

bendanban: 好文啊，博主盼你更新啊~

RTP学习（四）：自己安装嵌入式RTP

行知E人: #include

关于qt中的tr（）函数

dinger555: 解释的牛逼

浅谈 Qt 布局管理器 如何让控件排列整齐

行潇: 请问，用代码怎么实现好呢？

Qt Creator添加资源

xd2008456: 楼组的解答很是详细，真心谢谢楼组分享

终于实现了S3C6410 video capture

zhjlance1: 版主能写个文档么？小弟完全不会弄，不知道如何下手，理论倒是有一大堆

关于qt中的tr（）函数

木容峰: 学习

QString s1 = “我是中文”; QString s2(“我是中文”); QString s3; s3 = “我是中文”

• s1、s2 用的是QString的构造函数QString（const char \* str）

• s3 用的是QString的赋值操作符 QString & operator=（const char \* str）

如果不指定编码，s1,s2,s3将全部都是(国内大多数人所称的)乱码。因为QString将这些const char \*按照latin1来解释的，而不是用户期待的gbk或utf8。

QTextCodec::setCodecForCStrings(QTextCodec::codecForName(“GB2312”));QTextCodec::setCodecForCStrings(QTextCodec::codecForName(“UTF-8”))

这两条语句中的一条可以解决问题，至于如何选择，此处不再重复。

## QObject::tr

说实话，在tr中使用中文不是个好主意。不过既然总有人用（无论是(1)还是(2b)），而且总有人遇到问题，所以还是简单整理一下吧。

相比QCoreApplication::translate，大家用tr应该用的很多了，尽管不少人不清楚tr究竟是做什么的^\_^

tr(“我是中文”);

这调用的是下面这个函数(至少我们可这么认为是)。

QString QObject::tr（const char \* sourceText, const char \* disambiguation = 0, int n = -1）

与QString("我是中文")完全一样，你必须告诉tr这个窄字符串是何种编码？你不告诉它，它就用latin1。于是所谓的乱码问题就出来了。

如何告诉tr你写的这几个汉字在磁盘中保存的是何种编码呢？这正是

QTextCodec::setCodecForTr(QTextCodec::codecForName(“GB2312”));  
QTextCodec::setCodecForTr(QTextCodec::codecForName(“UTF-8”));

所做的。这两个选择的原则，由于和前文完全一样，此处也不再重复。

如果你的编码采用的utf8，可以直接使用trUtf8而不必设置setCodecForTr。

如果你只关心乱码问题，到此为止就可以了(下面不再关注编码)。如果想对tr进一步了解，不妨。。继续。。

## QCoreApplication::translate

我们知道tr是用于实现程序的国际化(或者说多语言翻译)，看Qt相关资料的话，我们知道实现该功能的还有下面这个函数：

QString QCoreApplication::translate（const char \* context, const char \* sourceText, const char \* disambiguation, Encoding encoding, int n）

其实，这个才是真正进行翻译操作的函数，前面我们提到的tr最终是调用这个函数来实现翻译功能的(稍后我们会看tr是如何调用translate的)。

对tr和这个函数，manual中都有比较详尽的解释。我们这儿简单看一二个参数：

- context 上下文，一般就是需要翻译的字符串所在的类的名字
- sourceText 需要翻译的字符串。(我们关注的编码其实就是它的编码)
- disambiguation 消除歧义用的。(比如我们的类内出现两处"close"，一处含义是关闭，另一处含义是亲密的。显然需要让翻译人员知道这点区别)
- encoding 指定编码。它有两个值
  - CodecForTr 使用setCodecForTr()设置的编码来解释 sourceText
  - UnicodeUTF8 使用utf8编码来解释 sourceText
  - 其实这两个分别对应tr和trUtf8

- n 处理单复数(对中文来说，不存在这个问题)

## tr与translate

这两个函数的说明，一个在QObject的manual，另一个在QCoreApplication的manual中。

介绍一下tr与translate的关系。前面提到了，tr调用的是translate。如果仅仅这样一说，没有证据，还真难以让大家相信。好吧，继续

### tr 在何处定义

你可能说：这不废话吗，manual中写得明白的，它是QObject的静态成员函数。而且还有源码为证：

```
//来自 src/corelib/kernel/qobject.h #ifdef
qdoc static QString tr(const char *sourceText,const char *comment = 0, int n =
-1); static QString trUtf8(const char *sourceText, constchar *comment = 0, int n = -1); #endif
```

嘿嘿，差点就被骗了，发现没：它们被预处理语句包住了。

这说明了什么呢？说明了这段代码仅仅是用来生成Qt那漂亮的文档的(qdoc3从代码中抽取信息，生成一系列的html格式的manual)。

啊，也就是说，这是假的。那么真正的定义呢？？在一个大家都很熟悉的地方,猜猜看？

这就是

Q\_OBJECT

该宏的定义在src/corelib/kernel/qobjectdefs.h中

```
#define Q_OBJECT \ public: \ Q_OBJECT_CHECK \ static const QMetaObiect staticMetaObject;
virtual void staticMetaObject::tr(const char *, const char *, int, const char *) const; \ virtual void
staticMetaObject::trUtf8(const char *, const char *, int, const char *) const; \ virtual void
acall(QMetaObject::Call, int, const void **); \ private:
```

### 其中的宏QT\_TR\_FUNCTIONS

```
# define QT_TR_FUNCTIONS \ static inline QString tr(const char *s, const char *c = 0) \ { return
staticMetaObject.tr(s, c); } \ static inline QString trUtf8(const char *s, const char *c = 0) \ {
return staticMetaObject.trUtf8(s, c); } \ static inline QString tr(const char *s, const char *c, int
n) \ { return staticMetaObject.tr(s, c, n); } \ static inline QString trUtf8(const char *s, const
char *c, int n) \ { return staticMetaObject.trUtf8(s, c, n); }
```

现在看到：tr调用的是 staticMetaObject对象的tr函数，staticMetaObject 的定义在moc生成的 xxx.moc 或 moc\_xxx.cpp 文件内(你随时可以验证的)。

staticMetaObject 是一个 QMetaObject 类的实例，我们继续看一下该类的源码：

```
/*! \internal */ QString QMetaObject::tr(const char *s, const char *c) const { returnQCoreApplica
tion::translate(d.stringdata, s, c, QCoreApplication::CodecForTr); } /*! \internal */ QString QMeta
Object::trUtf8(const char *s, const char *c) const { returnQCoreApplication::translate(d.stringdata,
s, c, QCoreApplication::UnicodeUTF8);
}
}
```

上一篇 QT 的信号与槽机制介绍

下一篇 linux学习笔记——各种问题记录

### 主题推荐

国际化      源代码      源码      语言      乱码

### 猜你在找

宏Q\_OBJECT

Qt网络编程TCP/IP二

更多相关资源：[qt](#)

Qt学习布局管理器QLayout类

Qt的皮肤设计Style Sheet

Qt入门—QLineEditInputMask

请检查是否是磁盘空间不足路径无效或权限不够

Qt导出Excel的简单实现

QT QWidget设置窗体透明度方法汇总

用qwt绘制AD波形转载

Qt学习停靠窗体QDockWidget类



查看评论

4楼 [zq4132](#) 2014-12-22 11:42发表



非常感谢分享，谢谢。正在学习中

3楼 [dinger555](#) 2014-02-18 14:42发表



解释的牛逼

2楼 [木容峰](#) 2013-12-27 04:25发表



学习

1楼 [sunnybeach](#) 2013-11-20 09:20发表



终于找到一篇透彻解释tr（）的文章了

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop AWS 移动游戏 Java Android iOS Swift 智能硬件 Docker OpenStack
- VPN Spark ERP IE10 Eclipse CRM JavaScript 数据库 Ubuntu NFC WAP jQuery
- BI HTML5 Spring Apache .NET API HTML SDK IIS Fedora XML LBS Unity
- Splashtop UML components Windows Mobile Rails QEMU KDE Cassandra CloudStack
- FTC coremail OPhone CouchBase 云计算 iOS6 Rackspace Web App SpringSide Maemo
- Compuware 大数据 aptech Perl Tornado Ruby Hibernate ThinkPHP HBase Pure Solr
- Angular Cloud Foundry Redis Scala Django Bootstrap