

个人资料

[管理]



一去、二三里

 Qing

 微博

博客等级:

18

博客积分:

78

新

博客访问:

407,943

关注人气:

378

获赠金笔:

12

赠出金笔:

0

荣誉徽章:

3






相关博文


- [qobject_cast强制类型转换
堕落天使](#)
- [QT信号和槽以结构体为参数传递复
草帽_strawhat](#)
- [Qt之QTreeView（三）
一去、二三里](#)
- [Qt之QTreeView（一）
一去、二三里](#)
- [Qt之QTreeView（二）
一去、二三里](#)
- [一款VS开发利器-VisualAssistX
一去、二三里](#)
- [Qt之模型/视图
一去、二三里](#)
- [QWT编译、配置、使用（VS2010+Qt）
一去、二三里](#)
- [Qt之鼠标样式
一去、二三里](#)
- [获取当前系统的信息（CPU、物理内
一去、二三里](#)

更多>>

正文

字体大小: 大 中 小

Qt之信号与槽  (2013-07-26 10:29:07) [编辑][删除]

 转载 ▼

标签: qt 信号与槽 qt信号与槽 多信号 分类: Qt

在遇到多信号问题的时候,你是否经常会连接多个槽函数呢?如果你的答案是绝对的,那么你已经Out很久了。多信号连接多个槽,实现不同的槽就在潜意识的加大程序的开销!那么为什么不去链接同一个槽呢?

今天在写下这篇文章,感觉有些唐突,但是又不得不写!因为信号与槽是Qt里面的最基础而且是最重要的部分,有很多人问过我关于信号与槽的问题,就总结一下。Qt主要包括: Qt基础部分 (Qt入门、Qt对话框、Qt窗口、自定义窗口部件)、Qt中级 (布局管理、事件处理、二维绘图、容器、数据库、多线程、网络等)、Qt高级 (国际化、自定义样式、三维绘图、创建插件、嵌入式编程等)。

信号与槽的连接方式看起来会是这样的:

Qt5之前:

```
connect(sender, SIGNAL(signal), receiver, SLOT(slot));
```

Qt5开始:

```
connect(sender, &Sender::signal, receiver, &Receiver::slot);
```

前者:

sender和receiver是指向QObject的指针, signal和slot是不带参数的函数名。SIGNAL()宏和SLOT()宏会把他们的参数转换成相应的字符串。

后者:

- (1) 编译器, 检查信号与槽是否存在, 参数类型检查, Q_OBJECT宏是否存在
- (2) 信号可以和普通函数、类的普通成员函数、lambda函数连接 (不在局限于信号和槽函数)
- (3) 参数可以是typedef的或者使用不同的namespace specifier
- (4) 可以允许一些自动类型的转换 (即信号和槽函数类型不必完全匹配)

1、一个信号连接一个槽

```
connect(slider, &QSlider::valueChanged, spin_box, &QSpinBox::setValue);
```

2、一个信号连接多个槽

```
connect(slider, &QSlider::valueChanged, spin_box, &QSpinBox::setValue);
connect(slider, &QSlider::valueChanged, this, &QWidget::showValue);
```

3、多个信号连接同一个槽

```
connect(push_button, &QPushButton::clicked, this, &QWidget::show);
connect(tool_button, &QToolButton::clicked, this, &QWidget::show);
```

4、一个信号连接另一个信号

```
connect(push_button, &QPushButton::clicked, this, &QWidget::buttonClicked);
```

5、断开链接

```
disconnect(push_button); //断开push_button的所有连接
disconnect(push_button, &QPushButton::clicked, this, &QWidget::show); //断开此信号连接的槽
```

好了,这些都是最基本的应用。那么多个信号连接同一个槽的时候如何进行区分呢?

方法一:

```
typedef enum{
    BUTTON_1,
    BUTTON_2,
    BUTTON_3,
    BUTTON_4
}BUTTON;
```

```
push_button_1->setObjectName(QString::number(BUTTON_1, 10));
```

- 2550mAh电
- 日本雾霾之战：民众与政府的博弈
- 南非猎豹冒死捕杀豪猪被扎满嘴刺
- 别光顾着把枪口对准柴静
- 马来西亚男子与眼镜王蛇接吻(图)
- 【早评】降息利好将会在后市中逐渐
- 哪些人能柴静的“穹顶之下”赚
- “深圳机场撞人事件”不是一个人
- 政采剔除国外品牌：早该说不了！
- 【DIY】做一枚宫灯迎元宵



[查看更多>>](#)

谁看过这篇博文

	flwwater	3月2日
	只会放大...	2月28日
	mule	2月27日
	todaysky	2月27日
	邪风	2月27日
	Ling	2月27日
	NOHACKCC-...	2月26日
	袁磊	2月26日
	自然麻雀	2月26日
	右右youyou	2月24日
	浪到西	2月24日
	362692213	2月17日

```
push_button_2->setObjectName(QString::number(BUTTON_2, 10));
tool_button_1->setObjectName(QString::number(BUTTON_3, 10));
tool_button_2->setObjectName(QString::number(BUTTON_4, 10));
connect(push_button_1, &QPushButton::clicked, this, &MyWidget::changeButton);
connect(push_button_2, &QPushButton::clicked, this, &MyWidget::changeButton);
connect(tool_button_1, &QToolButton::clicked, this, &MyWidget::changeButton);
connect(tool_button_2, &QToolButton::clicked, this, &MyWidget::changeButton);
```

```
void MyWidget::changeButton()
{
    QObject *object = QObject::sender();
    QPushButton *push_button = qobject_cast(object);
    QToolButton *tool_button = qobject_cast<</span>QToolButton  *>(object);
    int index;
    if(push_button)
    {
        QString object_name = push_button->objectName();
        index = object_name.toInt();
    }
    else if(tool_button )
    {
        QString object_name = tool_button->objectName();
        index = object_name.toInt();
    }

    QString information = QString("");
    switch(index)
    {
    case BUTTON_1:
        information = QString("clicked 1");
        break;

    case BUTTON_2:
        information = QString("clicked 2");
        break;

    case BUTTON_3:
        information = QString("clicked 3");
        break;

    case BUTTON_4:
        information = QString("clicked 4");
        break;

    default:
        information = QString("which is clicked?");
        break;
    }
    QMessageBox::information(NULL, QString("Title"), information);
}
```

当然，setObjectName不是专门用来干这事的，也可以使用text进行区分或者其它方法，这里介绍的只是一种思路而已！

方法二：

QSignalMapper类可以简单的理解为信号的翻译和转发器， 它可以把一个无参数的信号翻译成带int参数、QString参数、QObject*参数或者QWidget*参数的信号，并将之转发。

```
QSignalMapper *signal_mapper = new QSignalMapper(this);
connect(push_button_1, &QPushButton::clicked, signal_mapper, &QSignalMapper::map);
connect(push_button_2, &QPushButton::clicked, signal_mapper, &QSignalMapper::map);
connect(tool_button_1, &QToolButton::clicked, signal_mapper, &QSignalMapper::map);
connect(tool_button_2, &QToolButton::clicked, signal_mapper, &QSignalMapper::map);

signal_mapper->setMapping(push_button_1, QString::number(BUTTON_1, 10));
signal_mapper->setMapping(push_button_2, QString::number(BUTTON_2, 10));
signal_mapper->setMapping(tool_button_1, QString::number(BUTTON_3, 10));
signal_mapper->setMapping(tool_button_2, QString::number(BUTTON_4, 10));
connect(signal_mapper, &QSignalMapper::mapped, this, &MyWidget::changeButton);
```

```
void MyWidget::changeButton(QString text)
{
    int index = text.toInt();
    QString information = QString("");
    switch(index)
    {
    case BUTTON_1:
        information = QString("clicked 1");
        break;

    case BUTTON_2:
        information = QString("clicked 2");
        break;

    case BUTTON_3:
        information = QString("clicked 3");
        break;

    case BUTTON_4:
        information = QString("clicked 4");
        break;

    default:
        information = QString("which is clicked?");
        break;
    }
    QMessageBox::information(NULL, QString("Title"), information);
}
```

执行顺序

同一信号连接多个槽呢，槽函数执行没有绝对的先后顺序。

如：

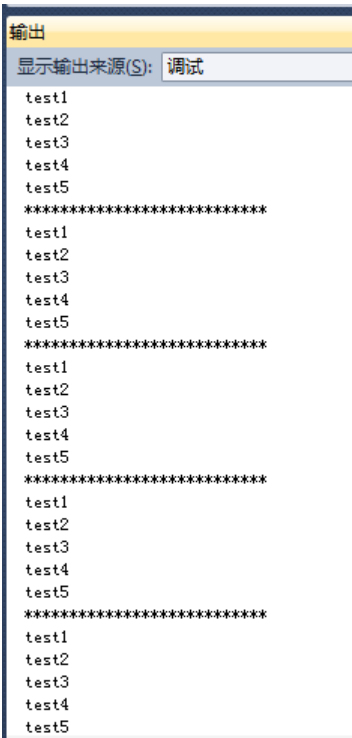
```
connect(slider, &QSlider::valueChanged, spin_box, &QSpinBox::setValue);
connect(slider, &QSlider::valueChanged, this, &QWidget::showValue);
```

在Qt5之前，并不是setValue一定会比showValue先执行。

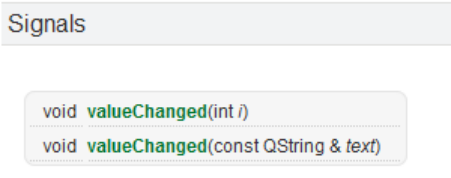
但在Qt5中，文档中这样介绍：

A signal can be connected to many slots and signals. Many signals can be connected to one slot. If a signal is connected to several slots, the slots are activated in the same order in which the connections were made, when the signal is emitted. (一个信号连接多个槽，信号发射后，会按照链接顺序执行)。

经过简单测试的确如此：



重载函数连接：
关于QSpinBox的信号：



自定义槽函数：

```
private slots:
    void changeValue(int value);
```

connect(spin_box, &QSpinBox::valueChanged, this, &ListView::changeValue);
信号与槽连接看上去很正确，但是会出现如下错误：
l:\listview.cpp(27): error C2664: "QObject::connect(const QObject *, const char *, const QObject *, const char *, Qt::ConnectionType)": 不能将参数 2 从 "overloaded-function" 转换为 "const char *" 上下文不允许重载函数的歧义
意思就是说不能够明确的找出到底调用的是哪个信号（因为只有函数名称，并无详细参数说明），所以需加上参数说明，调用static_cast进行转换。
connect(spin_box, static_cast(&QSpinBox::valueChanged), this, &ListView::changeValue);

总结就到这里，都是很常用的东西，编程过程中多注意细节部分，多总结就好了。很多东西文档里都说的很清楚，write less, do more。。。

注：
技术在于交流、沟通，转载请注明出处并保持作品的完整性。
作者： ☆奋斗ing♥孩子` 原文： http://blog.sina.com.cn/s/blog_a6fb6cc90101epbg.html。



后一篇：Qt之等待提示框三（QLabel进行多图片切换）

评论

重要提示：警惕虚假中奖信息

[发评论]

realfans

早期版本，同一信号连接多个槽呢，槽函数执行没有绝对的先后顺序。好像从4.6开始（不确认），一个信号连接到多个槽，信号发射后，会按连接的顺序执行。见Qt帮助文档：
If a signal is connected to several slots, the slots are activated in the same order as the order the connection was made, wh [查看更多](#)

2013-11-25 15:44 回复(1)

孟旭昕



2014-11-17 22:30 回复(0)

bw7ibfp9k7

大神 学习了

2014-12-11 15:31 回复(1)

发评论

一去、二三里：

☐ 分享到微博

☐ 匿名评论

验证码： [请点击后输入验证码](#) [收听验证码](#)

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

< 前一篇

Qt之再谈阴影边框

后一篇 >

Qt之等待提示框三（QLabel进行多图片切换）