Qt信号与槽机制

信号与槽（Signal & Slot）是 [Qt](http://c.biancheng.net/qt/" \t "_blank) 编程的基础，也是 Qt 的一大创新。因为有了信号与槽的编程机制，在 Qt 中处理界面各个组件的交互操作时变得更加直观和简单。  
  
信号（Signal）就是在特定情况下被发射的事件，例如PushButton 最常见的信号就是鼠标单击时发射的 clicked() 信号，一个 ComboBox 最常见的信号是选择的列表项变化时发射的 CurrentIndexChanged() 信号。  
  
GUI 程序设计的主要内容就是对界面上各组件的信号的响应，只需要知道什么情况下发射哪些信号，合理地去响应和处理这些信号就可以了。  
  
槽（Slot）就是对信号响应的函数。槽就是一个函数，与一般的[C++](http://c.biancheng.net/cplus/" \t "_blank)函数是一样的，可以定义在类的任何部分（public、private 或 protected），可以具有任何参数，也可以被直接调用。槽函数与一般的函数不同的是：槽函数可以与一个信号关联，当信号被发射时，关联的槽函数被自动执行。  
  
信号与槽关联是用 connect() 函数实现的，其基本格式是：

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

connect() 是 QObject 类的一个静态函数，而 QObject 是所有 Qt 类的基类，在实际调用时可以忽略前面的限定符，所以可以直接写为：

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

其中，sender 是发射信号的对象的名称，signal() 是信号名称。信号可以看做是特殊的函数，需要带括号，有参数时还需要指明参数。receiver 是接收信号的对象名称，slot() 是槽函数的名称，需要带括号，有参数时还需要指明参数。  
  
SIGNAL 和 SLOT 是 Qt 的宏，用于指明信号和槽，并将它们的参数转换为相应的字符串。例如，在 samp2\_1（前面章节中的项目）的 ui\_widget.h 文件中，在 setupUi() 函数中有如下的语句：

QObject::connect(btnClose, SIGNAL(clicked()), Widget, SLOT(close()));

其作用就是将 btnClose 按钮的 clicked() 信号与窗体（Widget）的槽函数 close() 相关联，这样，当单击 btnClose 按钮（就是界面上的“Close”按钮）时，就会执行 Widget 的 close() 槽函数。  
  
关于信号与槽的使用，有以下一些规则需要注意：

* 一个信号可以连接多个槽，例如：

connect(spinNum, SIGNAL(valueChanged(int)), this, SLOT(addFun(int));  
connect(spinNum, SIGNAL(valueChanged(int)), this, SLOT(updateStatus(int));

这是当一个对象 spinNum 的数值发生变化时，所在窗体有两个槽进行响应，一个 addFun()用于计算，一个 updateStatus() 用于更新状态。  
  
当一个信号与多个槽函数关联时，槽函数按照建立连接时的顺序依次执行。  
  
当信号和槽函数带有参数时，在 connect()函数里，要写明参数的类型，但可以不写参数名称。

* 多个信号可以连接同一个槽，例如在 samp2\_2（前面章节中的项目）中，让三个选择颜色的 RadioButton的clicked() 信号关联到相同的一个自定义槽函数 setTextFontColor()。

connect(ui->rBtnBlue,SIGNAL(clicked()),this,SLOT(setTextFontColor()));  
connect(ui->rBtnRed,SIGNAL(clicked()),this,SLOT(setTextFontColor()));  
connect(ui->rBtnBlack,SIGNAL(clicked()),this,SLOT(setTextFontColor()));

这样，当任何一个 RadioButton 被单击时，都会执行 setTextFontColor() 函数。

* 一个信号可以连接另外一个信号，例如：

connect(spinNum, SIGNAL(valueChanged(int)), this, SIGNAL (refreshInfo(int));

这样，当一个信号发射时，也会发射另外一个信号，实现某些特殊的功能。

* 严格的情况下，信号与槽的参数个数和类型需要一致，至少信号的参数不能少于槽的参数。如果不匹配，会出现编译错误或运行错误。
* 在使用信号与槽的类中，必须在类的定义中加入宏 Q\_OBJECT。
* 当一个信号被发射时，与其关联的槽函数通常被立即执行，就像正常调用一个函数一样。只有当信号关联的所有槽函数执行完毕后，才会执行发射信号处后面的代码。

信号与槽机制是 Qt GUI 编程的基础，使用信号与槽机制可以比较容易地将信号与响应代码关联起来。