

# Understanding Signals and Slot in Qt

This article shows how to use Qt's Signals and Slots mechanism.

## Introduction

The Signals and slots mechanism is fundamentally used to bind the object together without the object knowing any thing about each other.

Here slots are almost identical to C++ member functions which can be public, protected, or private. They can be directly invoked like any other C++ member functions. The difference is that a slot can also be connected to a signal, in which case it is automatically called each time the signal is emitted. We can use a connect statement to connect a signal and a slot. The connect() statement looks like this:

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

Here is the example of it.

```
QObject::connect(spinBox, SIGNAL(valueChanged(int)), slider, SLOT(setValue(int)));
```

In the above example a signal is emitted each time the value of the spinbox is changed and this signal is received by a slider and it called the corresponding function in the slot. Where sender and receiver are pointers to [QObject](#) and where signal and slot are function signatures.

There are many other possibilities for connecting a signal and a slot in a different way.

- One signal can be connected to many slots:

```
connect(slider, SIGNAL(valueChanged(int)), spinBox, SLOT(setValue(int)));  
connect(slider, SIGNAL(valueChanged(int)), this, SLOT(updateStatusBarIndicator(int)));
```

When the signal is emitted, the slots are called one after the other, in an unspecified order.

**For Example of signal asn slots visit:** [Archived:How to use QSpinBox and QSlider](#)

**For More details in Signals and Srote visit:** [Signals and Slots in Qt](#)

- Many signals can be connected to the same slot:

```
connect(sender0, SIGNAL(overflow()), receiver1, SLOT(handleMathError()));  
connect(sender1, SIGNAL(divisionByZero()), receiver1, SLOT(handleMathError()));
```

- A signal can be connected to another signal:

```
connect(sender1, SIGNAL(function1()), receiver, SIGNAL(function2()));
```

When the first signal is emitted, the second signal is emitted as well. Apart from that, signal–signal connections are indistinguishable from signal–slot connections. • Disconnect can be used to remove the connection.

```
disconnect(sender0, SIGNAL(overflow()), receiver1, SLOT(handleMathError()));
```

In practice disconnect statement are rarely used because Qt removes all the connection automatically when it delete the objects.

## Declaring your own slots

If you are going to declare your own slots then you have to define your slots in the header file:

```
public slots:  
    void slots();
```

To connect this slots we have to write

```
connect(button1, SIGNAL(clicked()), this, SLOT(slots()));
```

For more information about signals and slots see [Signals and Slots](#)