

## Externe Bibliotheken

- Alle benötigten Bibliotheken sollten unter
  - `\\campdata\public\3rdparty_vs9` von den Praktikumsrechnern bzw.
  - `janus.informatik.tu-muenchen.de/home/shared/3rdparty_vs9` via SCP zu finden sein
- Alle benötigten Bibliotheken, `camp3rdparty.bat` und `setx.exe` runterladen
- `camp3rdparty.bat` ausführen damit die Pfade gesetzt werden

# Einführung in Qt

## Qt



- C++ GUI Bibliothek
- Plattformübergreifend
- Duales Lizenzsystem (GPL und kommerziell)
- Populäre Alternativen: wxWidgets, FLTK, GTK
- Gut dokumentiert und viele Beispiele: <http://doc.trolltech.com>
- Komplette Objektorientierung
- Große Auswahl an *Widgets* (diverse GUI Funktionalitäten)

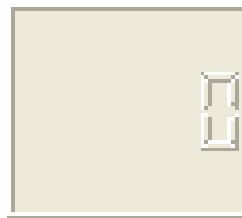
# Widgets

- Widgets sind visuelle Elemente einer graphischen Benutzerschnittstelle
- Grosse Auswahl an vordefinierten Widgets für GUIs
- Einfache Vererbung von vordefinierten Widgets um individuelle Funktionalität hinzuzufügen
- Können beliebig kombiniert werden
- Einfache Knöpfe und Textfelder bis zu komplexen Dialogen
  
- Widget Gallery  
<http://doc.trolltech.com/4.4/gallery.html>

## Widget Gallery – Display Widgets



- QProgressBar

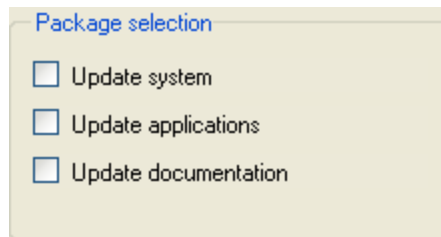


- QLCDNumber

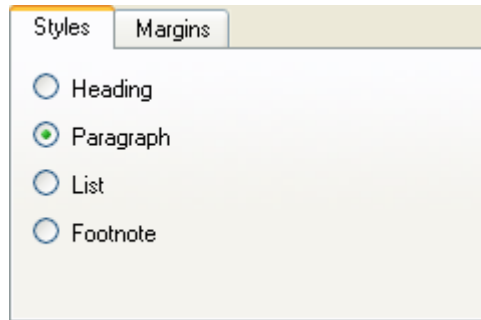
Text Label

- QLabel

# Widget Gallery – Containers

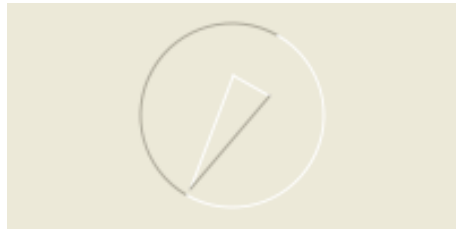
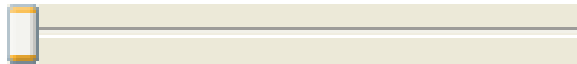


- QGroupBox



- QTabWidget

## Widget Gallery – Input Widgets



The **QTextEdit** class provides a widget that is used to edit and display both plain and rich text.

**QTextEdit** is an advanced *WYSIWYG* viewer/editor that can display images, lists and tables.



- QComboBox
- QLineEdit
- QSlider
  
- QDial
  
- QTextEdit
  
- QScrollBar

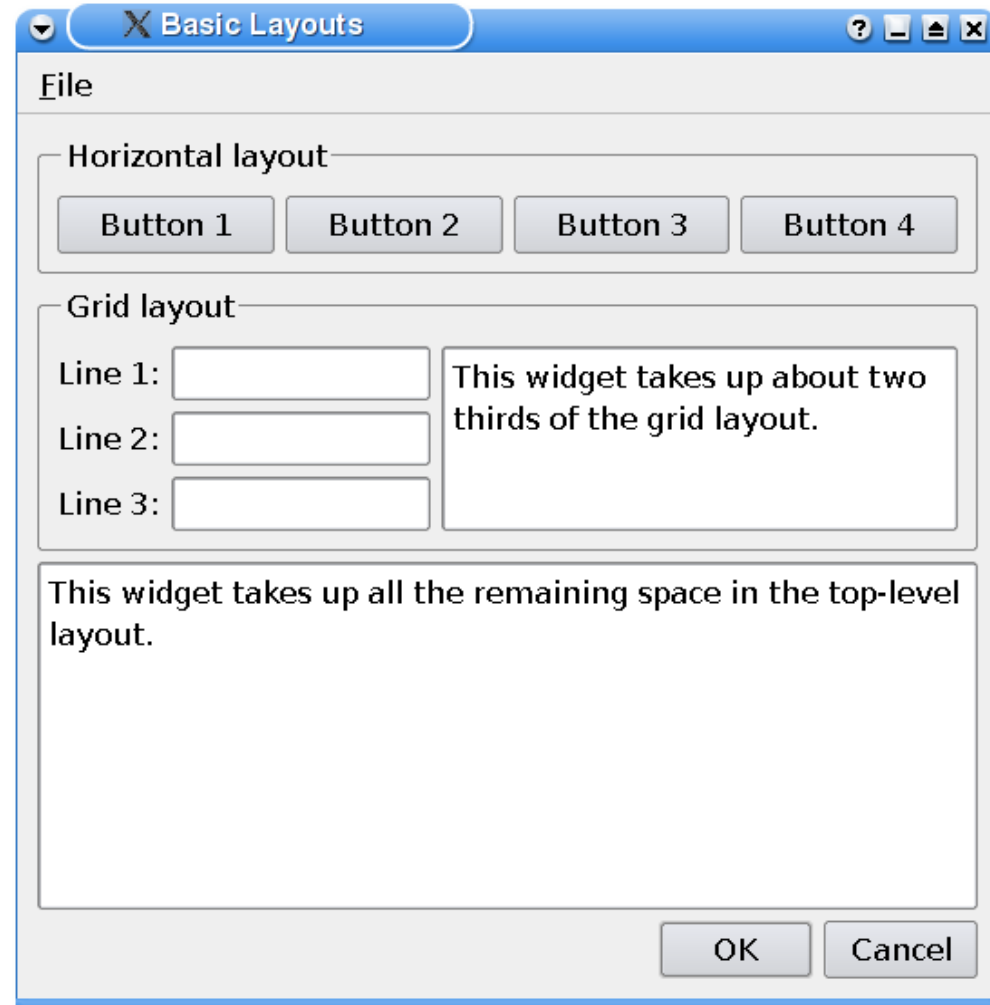
# Layout Manager

- QVBoxLayout
- QHBoxLayout
- QGridLayout

```
myLayout->addWidget(myWidget)
```

```
myLayout->addLayout(myInnerLayout)
```

```
myLayout->addSpacing(int space)
```



## Hello World Button – Code Snippet

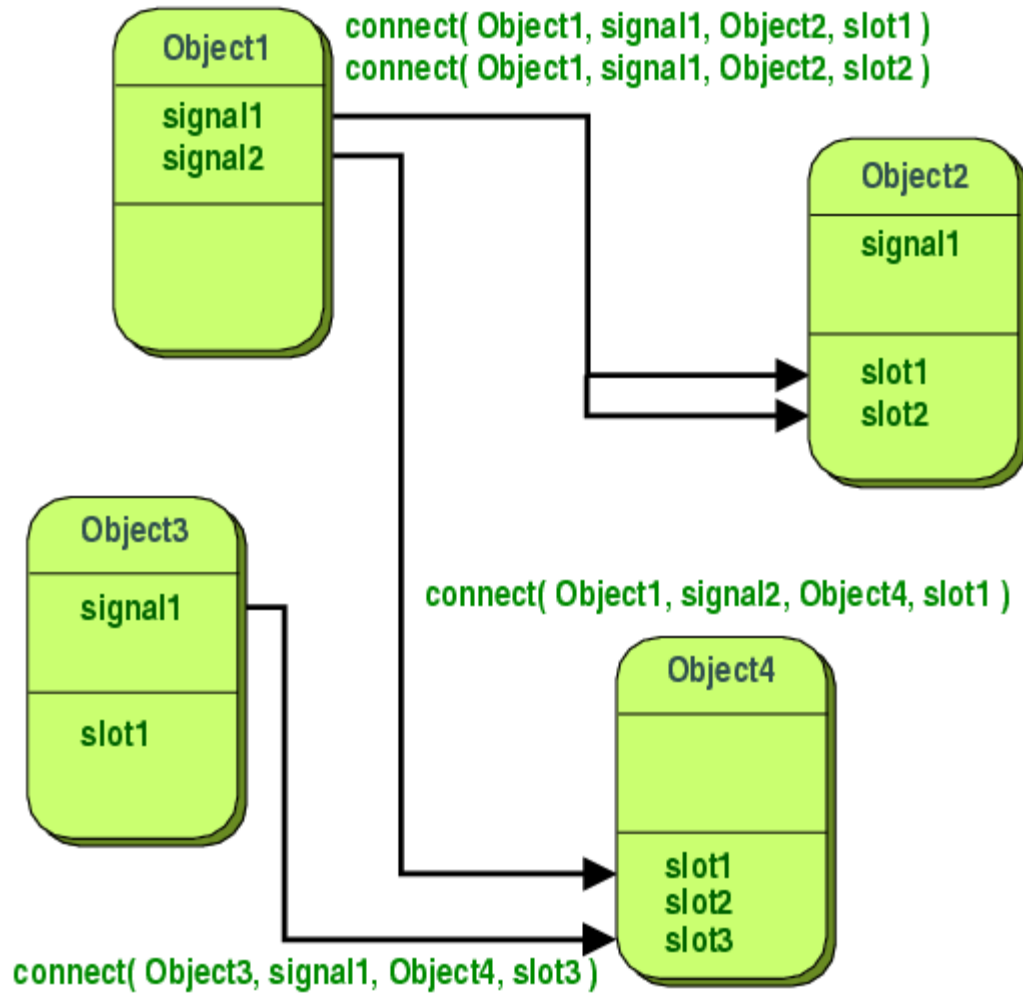
```
#include <QtGui/QApplication>
#include <QtGui/QPushButton>

int main(int argc, char *argv[])
{
    QApplication app(argc, argv);

    QPushButton hello("Hello world!");
    hello.resize(100, 30);

    hello.show();
    return app.exec();
}
```

# Signals and Slots

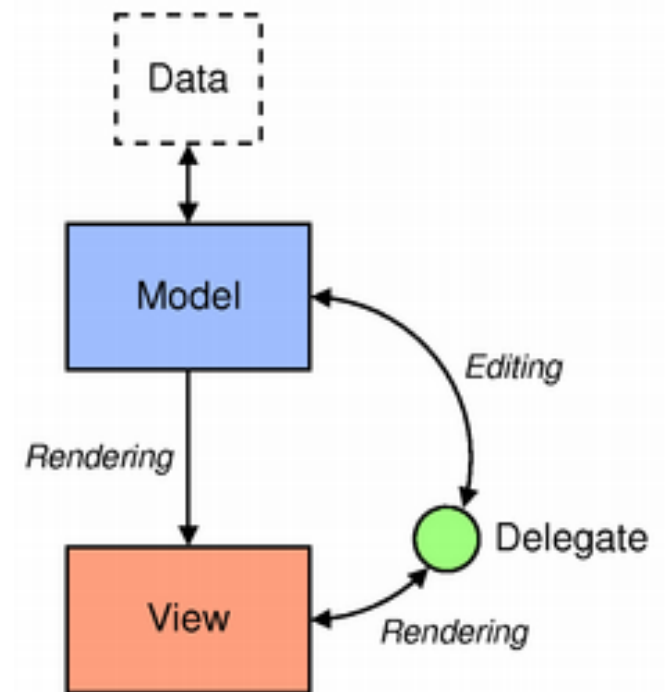


## Slots and Signals

- **Signals:** Widgets erzeugen Signale wenn bestimmte Aktionen ausgeführt werden (z.B. Button -> `clicked()`)
- **Slots:** Funktionen die von *Signals* (mittels `connect()`) oder auch normal aufgerufen werden können
- **Connect:** Verbindet ein *Signal* mit einem *Slot*
- **Beispiel:**  
`connect(button, SIGNAL(clicked()), QApplication, SLOT(quit()));`

# Model-View-Controller Konzept

- Mittlerweile teilweise umgesetzt: Model-View mit Delegates
- z.B. QStringListModel ist eine Liste von Strings
- Verschiedene Views (etwa QTableView, QListView) können verwendet werden



## Benutzerdefinierte Widgets

- Klasse muss von `QObject` oder einer Unterklasse (z.B. `QWidget`) abgeleitet sein
- Das `QObject` macro muss in der Klassendefinition stehen
- *Slots* können in der *public slots*, *protected slots* oder *private slots* Sektion stehen
- Es muss der `moc` compiler vor dem Präprozessor verwendet werden
  - Cpp-Datei `mocen` (header-file kompilieren)
  - Entstandene Datei auch in das Projekt einfügen

## MyWidget – Code Snippet

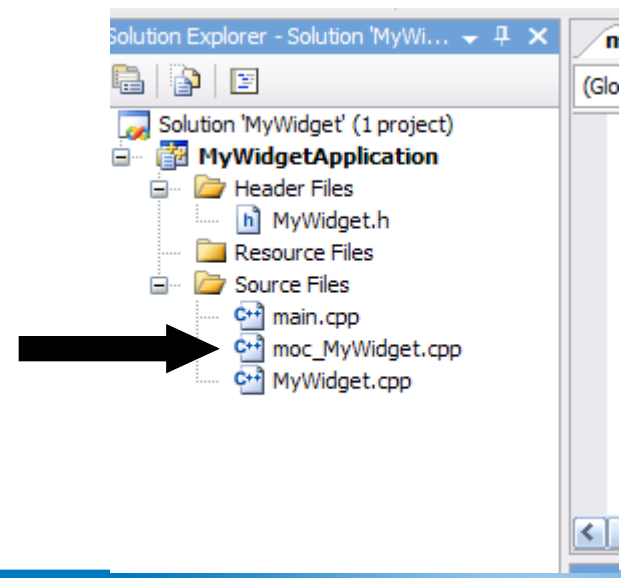
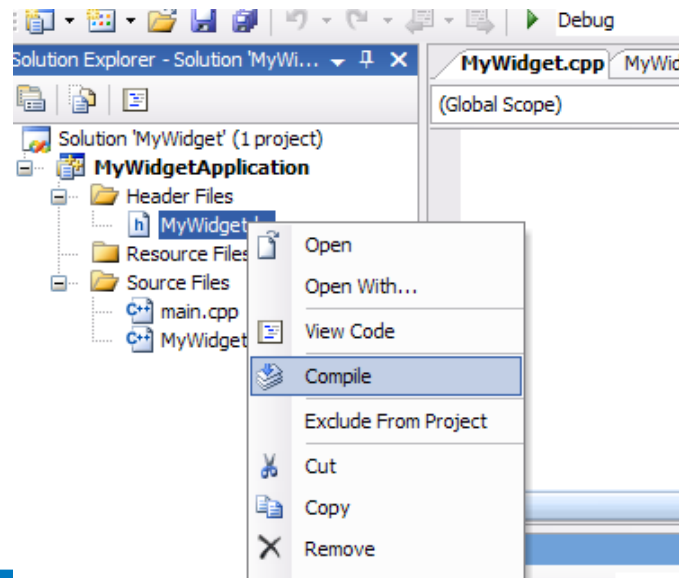
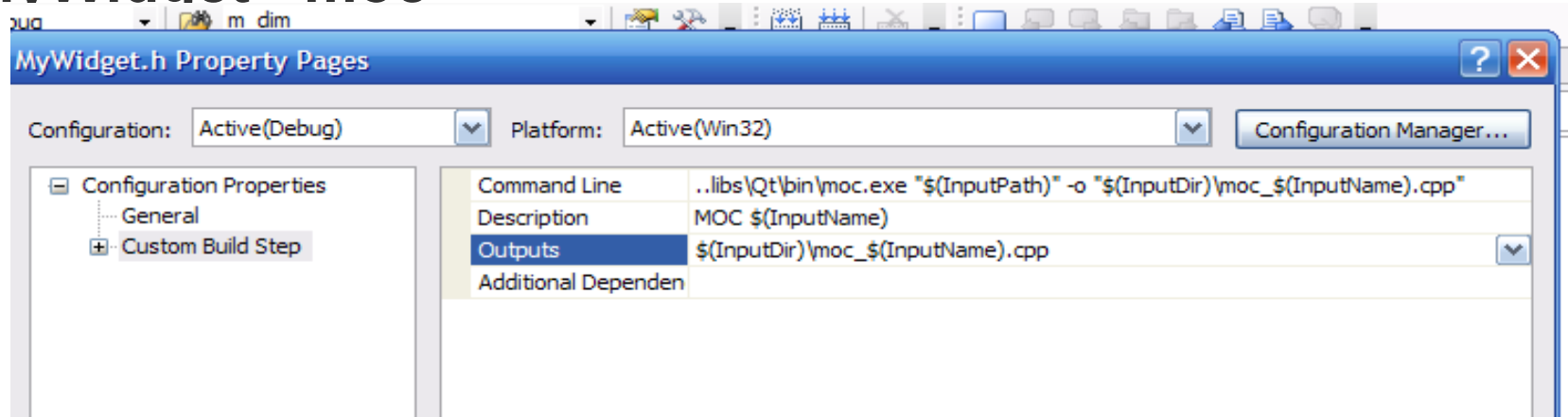
```
#include <QtGui/QWidget>

class MyWidget : public QWidget
{
    Q_OBJECT

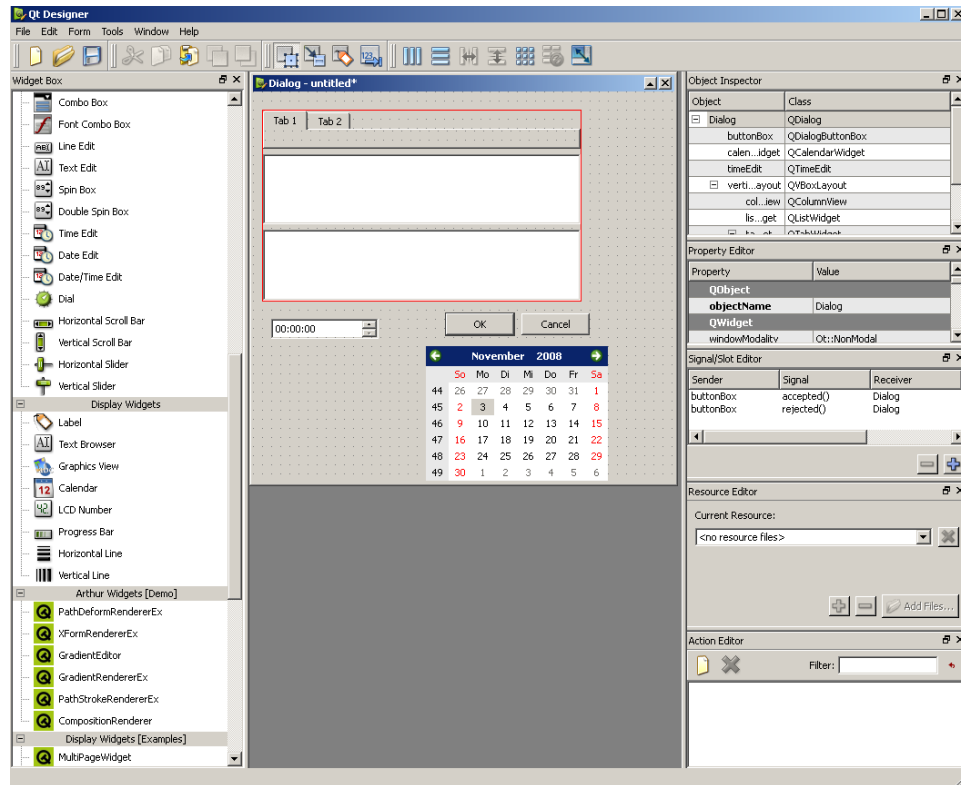
public:
    MyWidget(QWidget* parent = 0);
    ~MyWidget();
signals:
    positionChanged();
public slots:
    showContent();
private slots():
    ...;
}
```

MyWidget.h

# MyWidget - moc



# Qt Designer



<http://doc.trolltech.com/4.5/designer-using-a-ui-file.html>

## Bezug zu OpenGL - QGLWidget

- ermöglicht Integration von OpenGL Code zum Zeichnen von Objekten
- implementiert Methoden:

```
initializeGL() //initialisiert OpenGL
```

```
paintGL() //zeichnet OpenGL Objekte in Widget
```

```
resizeGL() //kann Größenänderungen handhaben
```

## Bezug zu OpenGL - QGLWidget

- weitere Methoden zu Mousefunktionen

```
mousePressEvent (MouseEvent* event);
```

```
mouseMoveEvent (MouseEvent* event);
```

```
mouseReleaseEvent (MouseEvent* event);
```

- weitere Methoden zu Tastaturfunktionen

```
keyPressEvent (KeyEvent* event);
```

```
keyReleaseEvent (KeyEvent* event);
```