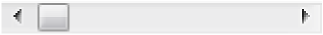
控件和布局

柴强

用户界面组件

- 界面使用控件建立

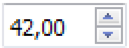
QLabel



QScrollBar

QLineEdit

QPushButton

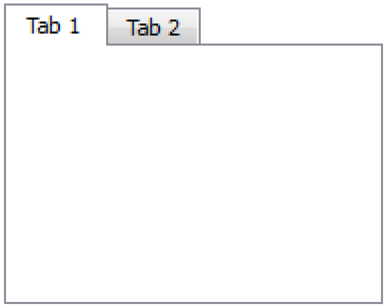


QDoubleSpinBox

控件的层级关系

- Container控件

QGroupBox



QTabWidget

控件的特点

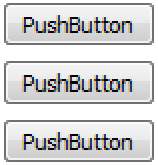
- 占据矩形空间
- 从输入设备接收事件
- 状态改变时发出信号
- 以层级的方式处理
- 可以包含其他控件

绝对布局

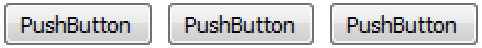
- 定义控件的大小和布局
  - move
  - resize
  - setGeometry
- 实例
  - QWidget *widget = new QWidget;
  - widget->resize(100,100);
  - widget->move(10,10);

布局

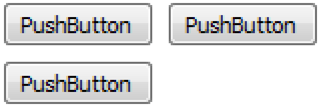
- 已有多种布局
- QHBoxLayout,QVBoxLayout,QGridLayout
- QFormLayout,QStackLayout



QVBoxLayout



QHBoxLayout



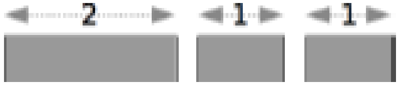
QGridLayout

布局实例

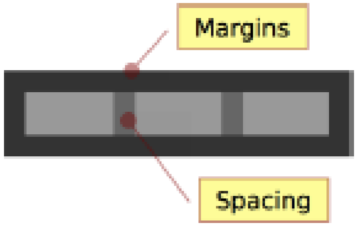
- $QTDIR/examples/layouts/basiclayout/
- 实例:ex-layout
- 实例:ex-gridlayout

布局

- **Stretch**
  - *Relative resize factor*
  - `QBoxLayout::addWidget(widget, stretch)`
  - `QBoxLayout::addStretch(stretch)`
  - `QGridLayout::setRowStretch(row, stretch)`
  - `QGridLayout::setColumnStretch(col, stretch)`



- **Contents Margins**
  - *Space reserved around the managed widgets.*
  - `QLayout::setContentsMargins(l,t,r,b)`
- **Spacing**
  - *Space reserved between widgets*
  - `QBoxLayout::addSpacing(size)`



实例

- so-layoutbuttons

处理文字的控件

- **QLabel**

  ```
  label = new QLabel("Text", parent);
  ```

  - setPixmap(pixmap) - as content

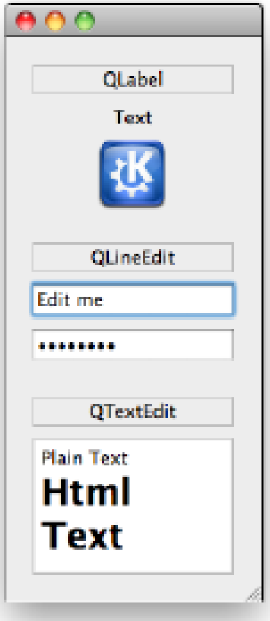
- **QLineEdit**

  ```
  line = new QLineEdit(parent);
  line->setText("Edit me");
  line->setEchoMode(QLineEdit::Password);
  connect(line, SIGNAL(textChanged(QString)) ...
  connect(line, SIGNAL(editingFinished()) ...
  ```

  - setInputMask(mask) - See Input Mask Documentation
  - setValidator(validator) - See Validator Documentation

- **QTextEdit**

  ```
  edit = new QTextEdit(parent);
  edit->setPlainText("Plain Text");
  edit->append("<h1>Html Text</h1>");
  connect(edit, SIGNAL(textChanged(QString)) ...
  ```



QLabel

Text

QLineEdit

Edit me

****

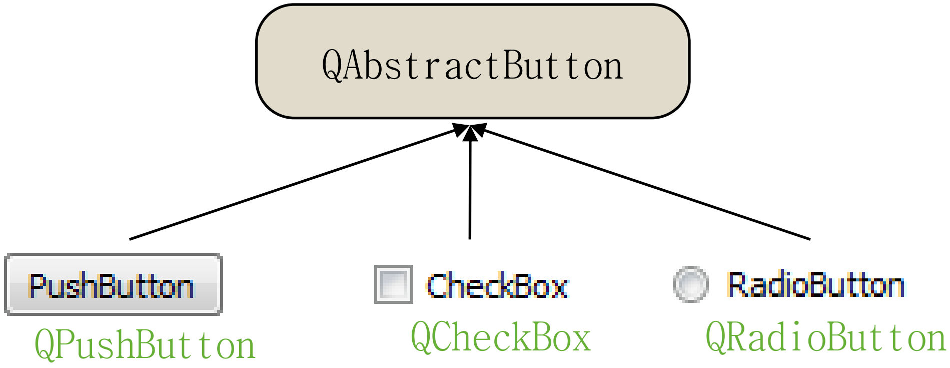
QTextEdit

Plain Text

Html Text

按钮

- 所有的按钮从QAbstractButton继承
- 信号
  - clicked()
  - toggled(bool)



按钮示例

- **QAbstractButton**
  - Abstract base class of buttons
- **QPushButton**

```
button = new QPushButton("Push Me", parent);
button->setIcon(QIcon("images/icon.png"));
connect(button, SIGNAL(clicked()) ...
```

  - setCheckable(bool) - toggle button
- **QRadioButton**

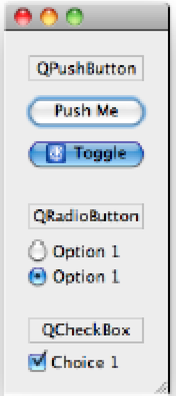
```
radio = new QRadionButton("Option 1", parent);
```

- **QCheckBox**

```
check = new QCheckBox("Choice 1", parent);
```

- **QButtonGroup** - non-visual button manager

```
group = new QButtonGroup(parent);
group->addButton(button); // add more buttons
group->setExclusive(true);
connect(group, SIGNAL(buttonClicked(QAbstractButton*)) ...
```

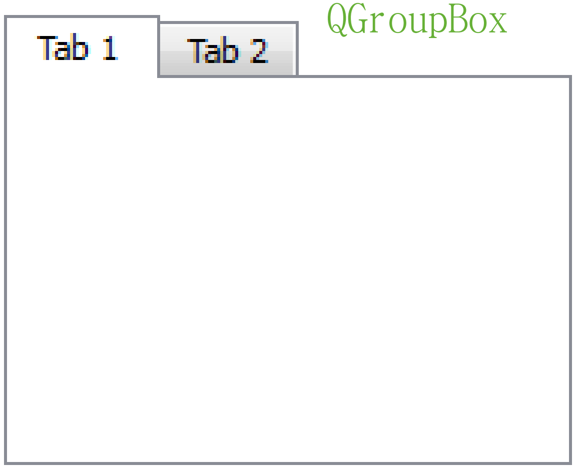


容器

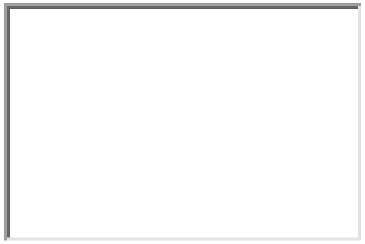
- 容器用来包含别的控件
  - QWidget可以被当做容器

QGroupBox

QGroupBox



```
QGroupBox *box = new QGroupBox();
QVBoxLayout *layout = new QVBoxLayout(box);
layout->addWidget(...);
...
```



QFrame

容器实例

- **QGroupBox**

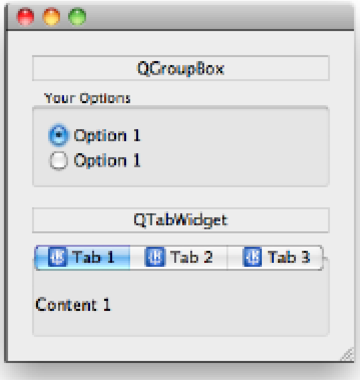
  ```
  box = new QGroupBox("Your Options", parent);
  // ... set layout and add widgets
  ```

  - setCheckable(bool) - checkbox in title

- **QTabWidget**

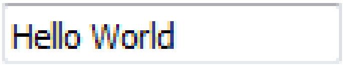
  ```
  tab = new QTabWidget(parent);
  tab->addWidget(widget, icon, "Tab 1");
  connect(tab, SIGNAL(currentChanged(int)) ...
  ```

  - setCurrentWidget(widget)
    - Displays page assoziated by widget
  - setTabPosition(position)
    - Defines where tabs are drawn
  - setTabsClosable(bool)
    - Adds close buttons



输入控件QLineEdit

- 使用QLineEdit实现单行输入
- 信号
  - textChanged
  - editingFinished
- 示例
  - lineedits

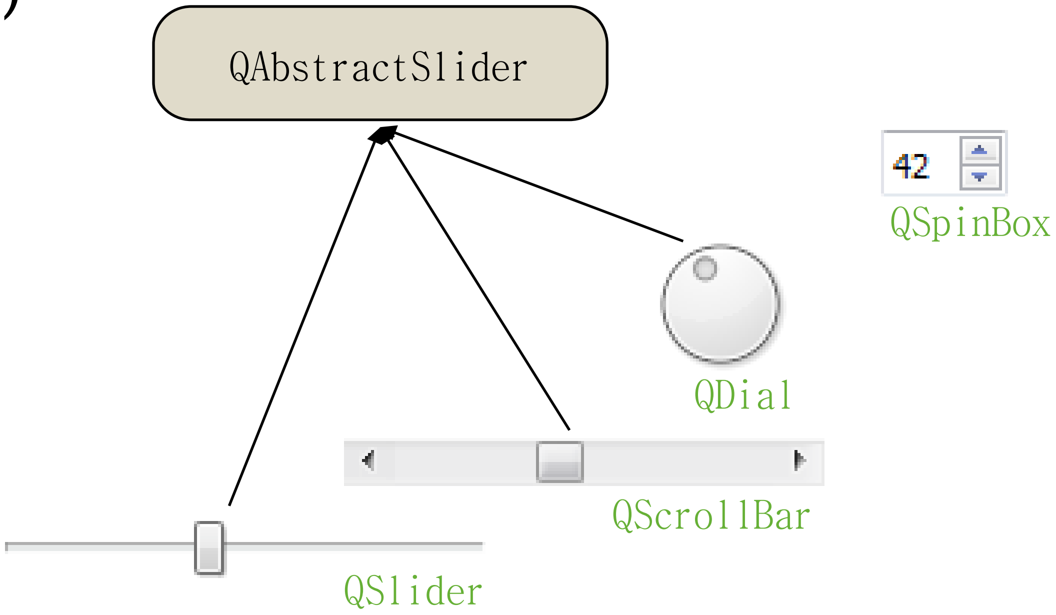


Hello World

QLineEdit

数据控件

- 滑动条
- 信号
  - valueChanged(int)



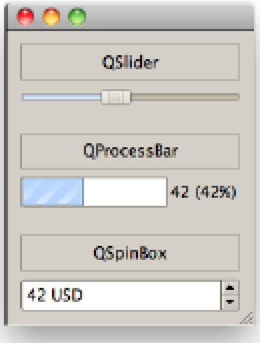
数据控件实例

- QSlider

```
slider = new QSlider(Qt::Horizontal, parent);
slider->setRange(0, 99);
slider->setValue(42);
connect(slider, SIGNAL(valueChanged(int)) ...
```

- QProgressBar

```
progress = new QProgressBar(parent);
progress->setRange(0, 99);
progress->setValue(42);
// format: %v for value; %p for percentage
progress->setFormat("%v (%p%)");
```



- QSpinBox

```
spin = new QSpinBox(parent);
spin->setRange(0, 99);
spin->setValue(42);
spin->setSuffix(" USD");
connect(spin, SIGNAL(valueChanged(int)) ...
```

ItemWidget

- **QComboBox**

  ```
  combo = new QComboBox(parent);
  combo->addItem("Option 1", data);
  connect(combo, SIGNAL(activated(int)) ...
  QVariant data = combo->itemData(index);
  ```

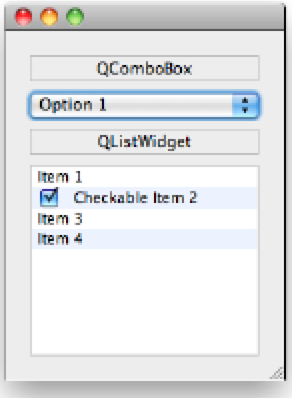
  - setCurrentIndex(index)

- **QListWidget**

  ```
  list = new QListWidget(parent);
  list->addItem("Item 1");
  // ownership of items with list
  item = new QListWidgetItem("Item 2", list);
  item->setCheckState(Qt::Checked);
  connect(list, SIGNAL(itemActivated(QListWidgetItem*)) ...
  ```

  - QListWidgetItem::setData(Qt::UserRole, data)

- *Other Item Widgets*: QTableWidget, QTreeWidget



数据项控件实例

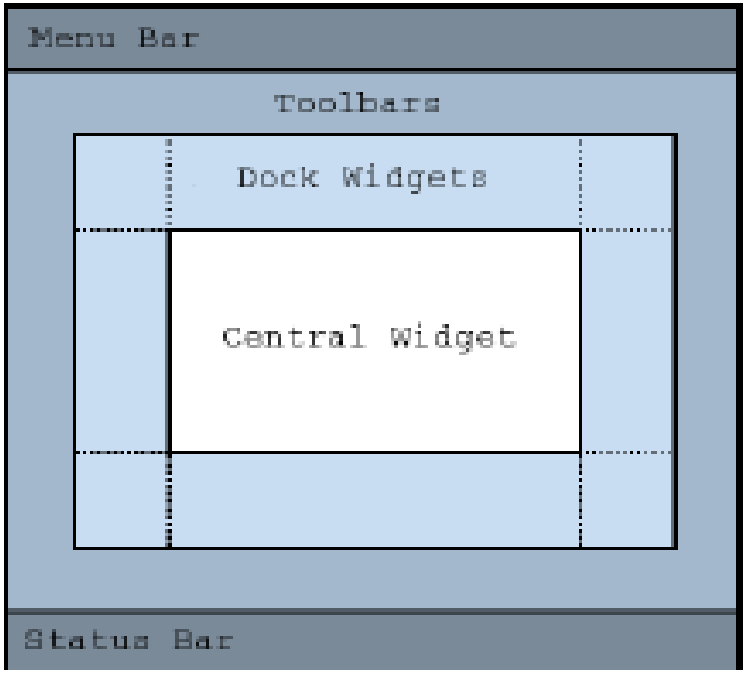
- ex-QListWidget
- ex-QTreeWidget
- ex-QTableWidget

顶层窗口

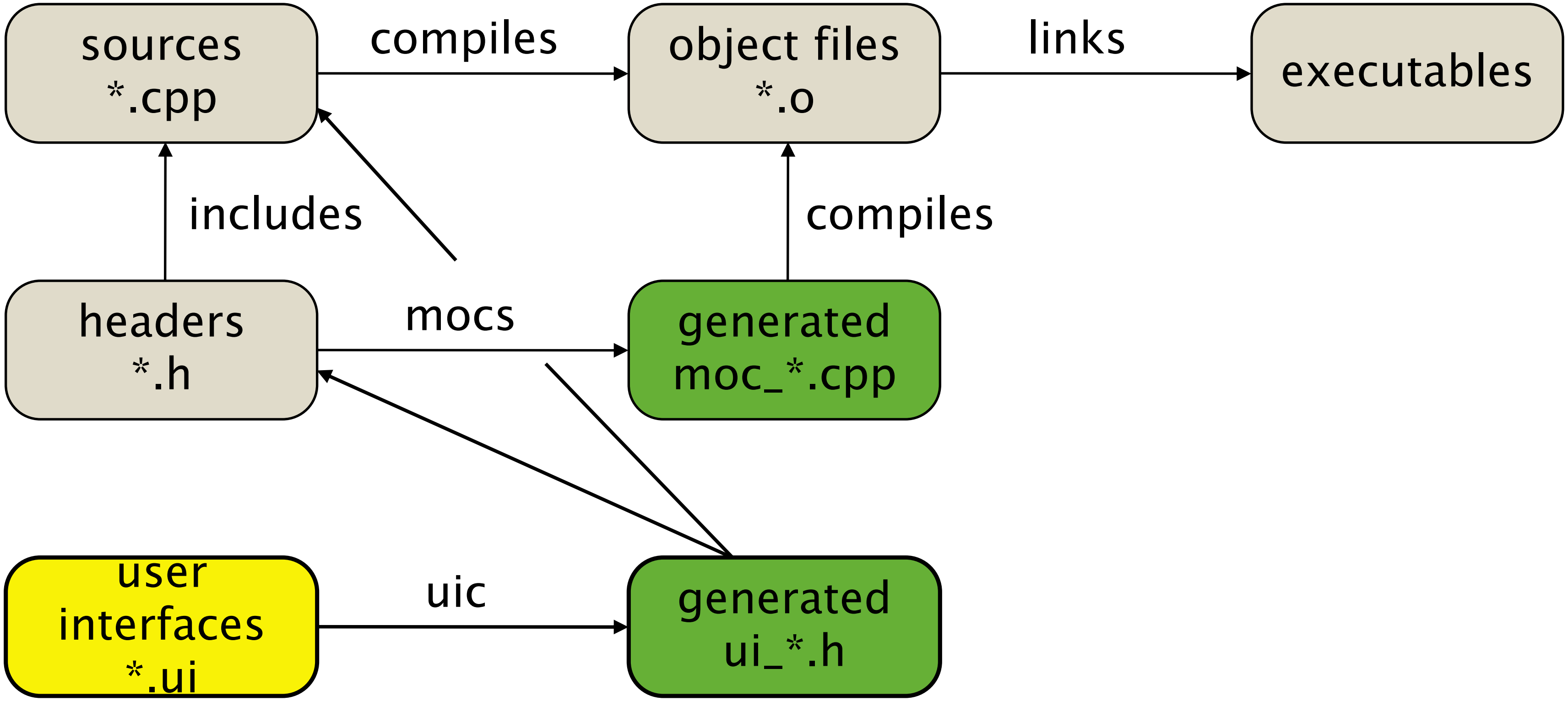
- 没有父亲的控件自动变为窗口
  - QWidget
  - QDialog
  - QMainWindow

顶层窗口QMainWindow示例

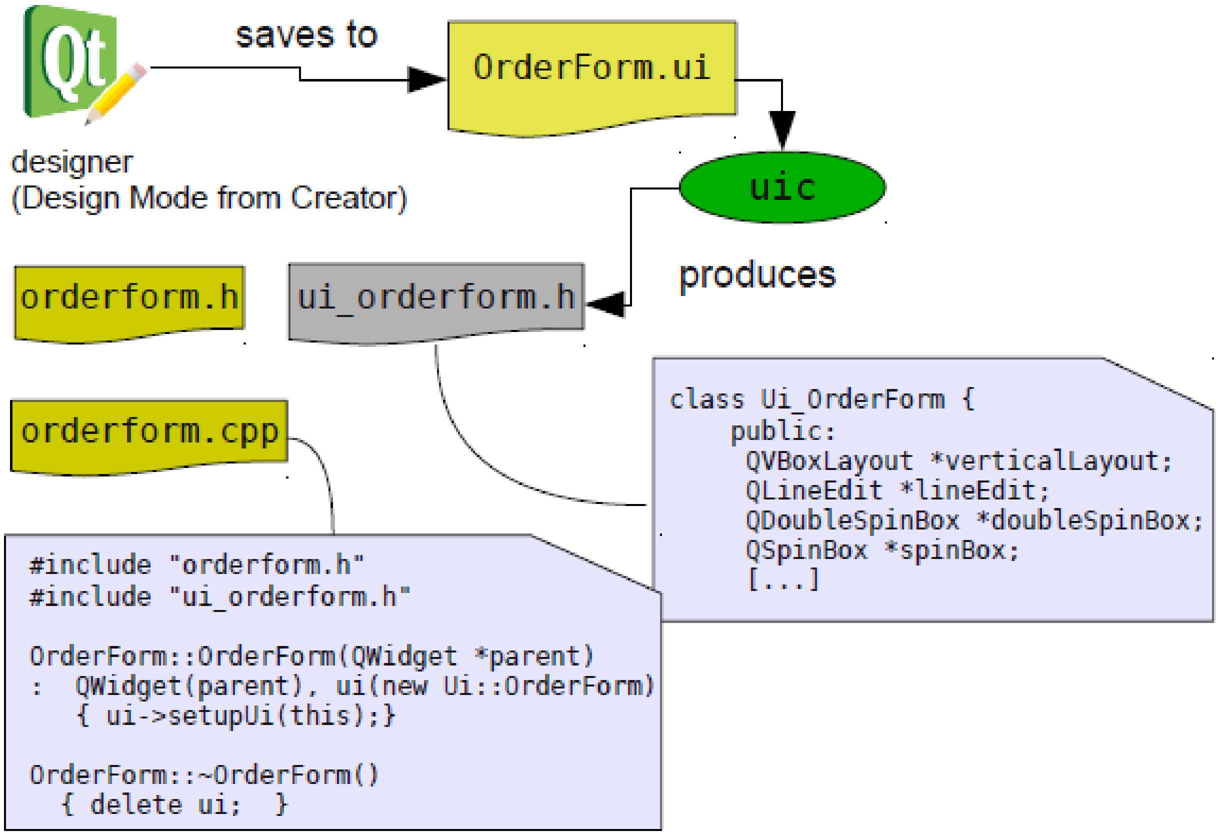
- QMenuBar
- QToolBar
- QStatusBar
- CentralWidget
- 示例
  - ex-mainwindow



QtDesigner


sources
*.cpp
compiles
object files
*.o
links
executables
includes
compiles
headers
*.h
mocs
generated
moc_*.cpp
user
interfaces
*.ui
uic
generated
ui_*.h

From .ui to C++



designer
(Design Mode from Creator)

saves to

OrderForm.ui

uic

produces

orderform.h

ui_orderform.h

orderform.cpp

```
class Ui_OrderForm {
    public:
     QVBoxLayout *verticalLayout;
     QLineEdit *lineEdit;
     QDoubleSpinBox *doubleSpinBox;
     QSpinBox *spinBox;
     [...]
```

```
#include "orderform.h"
#include "ui_orderform.h"

OrderForm::OrderForm(QWidget *parent)
:  QWidget(parent), ui(new Ui::OrderForm)
   { ui->setupUi(this);}

OrderForm::~OrderForm()
  { delete ui;  }
```

QtDesign示例

- orderform
- optionsdialog

对话框

- 自定义对话框从QDialog继承
- 创建布局
- 使用QDialogButtonBox处理对话框按钮
- 重写accept/reject
- 示例
  - $QTDIR/examples/dialog/extension

通用Dialog

- QFileDialog
- QMessageBox
- QProgressDialog
- QErrorMessage
- 示例
  - $QTDIR/examples/dialogs/standarddialogs

Q&A