# 第3篇 (基础) Qt登录对话框



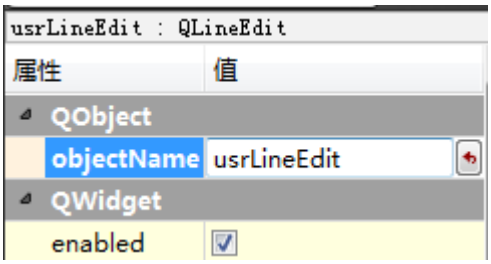
代码地址：https://github.com/Lornatang/QtStartQuicklyTutorial/tree/main/Example03

## 一、创建项目

1．新建Qt Gui应用，项目名称为`login`，类名和基类保持`MainWindow`和`QMainWindow`不变。

2．完成项目创建后，向项目中添加新的Qt设计师界面类，模板选择`Dialog without Buttons`，类名更改为`LoginDialog`。完成后向界面上添加两个标签`Label`、两个行编辑器`Line Edit`和两个按钮`Push Button`，设计界面如下图所示。

3. 这里在属性编辑器中将用户名后面的行编辑器的`object Name`属性更改为`usrLineEdit`，密码后面的行编辑器为`pwdLineEdit`，登录按钮为`loginBtn`，退出按钮为`exitBtn`。如下图所示。




4. 下面我们使用另外一种信号和槽的关联方法来设置退出按钮。在设计模式下面的信号和槽编辑器中，先点击左上角的绿色加号添加关联，然后选择发送者为`exitBtn`，信号为`clicked()`，接收者为`LoginDialog`，槽为`close()`。如下图所示。这样，当单击退出按钮时，就会关闭登录对话框。


5. 右击登录按钮，在弹出的菜单中选择“转到槽...”，然后选择`clicked()`信号并确定。转到相应的槽以后，添加函数调用：

```
void LoginDialog::on_loginBtn_clicked()
{
    accept();
}
```

6. 下面到`main.cpp`文件，更改内容如下：

```
#include "mainwindow.h"

int main(int argc, char* argv[]) {
    QApplication a(argc, argv);
    MainWindow w;
    LoginDialog dlg;
    if (dlg.exec() == QDialog::Accepted) {
        w.show();
        return a.exec();
```

```
    } else
        return 0;
}
```

7. 这时运行程序，按下退出按钮会退出程序，按下登录按钮会关闭登录对话框，并显示主窗口。

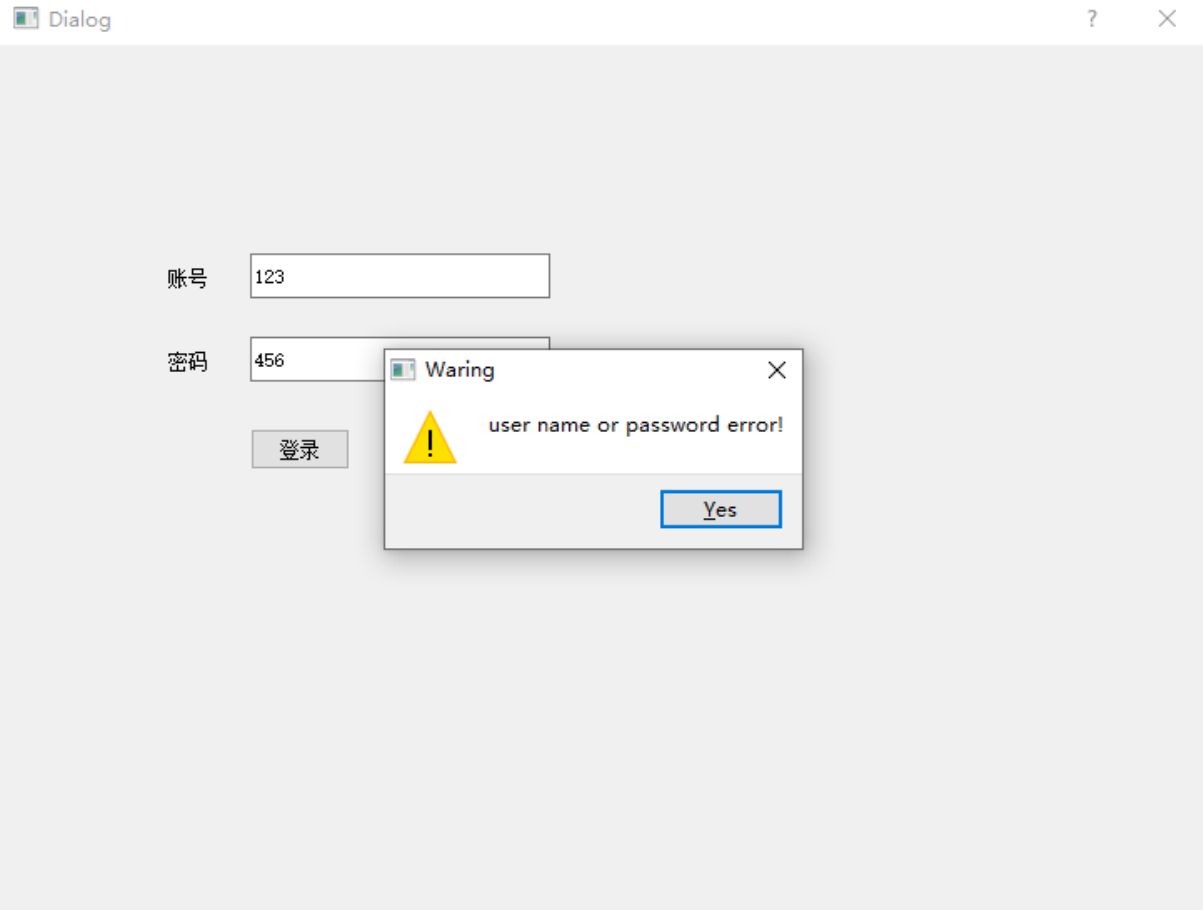
# 二、登录设置

1. 下面添加代码来实现使用用户名和密码登录，这里我们只是简单的将用户名和密码设置为了固定的字符串。到`logindialog.cpp`文件中将登录按钮的单击信号对应的槽的代码更改为：

```
void LoginDialog::on_loginBtn_clicked()
{
// 判断用户名和密码是否正确，
// 如果错误则弹出警告对话框
    if(ui->usrLineEdit->text() == tr("yafeilinux") &&
           ui->pwdLineEdit->text() == tr("123456"))
    {
       accept();
    } else {
       QMessageBox::warning(this, tr("Waring"),
                            tr("user name or password error!"),
                            QMessageBox::Yes);
    }
}
```

Qt中的`QMessageBox`类提供了多种常用的对话框类型，比如这里的警告对话框，还有提示对话框，问题对话框等。这里使用了静态函数来设置了一个警告对话框，这种方式很方便。其中的参数依次是：`this`表明父窗口是登录对话框；然后是窗口标题；然后是显示的内容；最后一个参数是显示的按钮，这里使用了一个Yes按钮。大家注意还要添加该类的头文件包含，即：`#include <QMessageBox>`。

2. 下面运行程序，如果输入用户名为`yafeilinux`，密码为`123456`，那么可以登录，如果输入其他的字符，则会弹出警告对话框，如下图所示。




3. 对于输入的密码，我们常见的是显示成小黑点的样式。下面点击`logindialog.ui`文件进入设计模式，然后选中界面上的密码行编辑器，在属性编辑器中将`echoMode`属性选择为`Password`。这时再次运行程序，可以看到密码显示已经改变了。如下图所示。


4. 当然，除了在属性编辑器中进行更改，也可以在`loginDialog`类的构造函数中使用`setEchoMode(QLineEdit::Password)`函数来设置。
5. 在行编辑器的属性栏中还可以设置占位符，就是没有输入信息前的一些提示语句。例如将密码行编辑器的`placeholderText`属性更改为“请输入密码”，将用户名行编辑器的更改为“请输入用户名”，运行效果如下图所示。


6. 对于行编辑器，还有一个问题就是，比如我们输入用户名，在前面添加了一个空格，这样也可以保证输入是正确的，这个可以使用`QString`类的`trimmed()`函数来实

现，它可以去除字符串前后的空白字符。下面将`logindialog.cpp`文件中登录按钮单击信号槽函数中的判断代码更改为：

```
if(ui->usrLineEdit->text().trimmed() == tr("123")
            && ui->pwdLineEdit->text() == tr("123"))
```

这时运行程序，已经实现相应的功能了。

7. 最后，当登录失败后，我们希望可以清空用户名和密码信息，并将光标定位到用户名输入框中。这个可以通过在判断用户名和密码错误后添加相应的代码来实现：

```
void LoginDialog::on_loginBtn_clicked()
{
    // 判断用户名和密码是否正确，如果错误则弹出警告对话框
    if(ui->usrLineEdit->text().trimmed() == tr("yafeilinux")
           && ui->pwdLineEdit->text() == tr("123456"))
    {
       accept();
    } else {
       QMessageBox::warning(this, tr("Waring"),
                            tr("user name or password error!"),
                            QMessageBox::Yes);
       // 清空内容并定位光标
       ui->usrLineEdit->clear();
       ui->pwdLineEdit->clear();
       ui->usrLineEdit->setFocus();
    }
}
```

下面运行程序，大家可以测试一下效果。