

# QFileDialog с Image Preview: метод хакера

В Qt3 была чудная возможность в диалоге открытия интегрировать виджет в котором осуществлять предпросмотра содержимого файла, в частности делать предпросмотр изображений. В Qt4 такую возможность убрали. Да, можно приделывать при помощи `ItemDelegate` кастомное отображение для иконок, и тем самым осуществлять предпросмотр, но не всегда это бывает удобным. Следующий вариант: писать свою реализацию класса для диалога открытия/сохранения. Я же решил попробовать хитрый способ...

Я никоим образом не претендую на правду в последней инстанции и не рекомендую относить данный метод к рекомендуемым практикам программирования, но как вариант решения с минимальными затратами он очень хорошо себя показывает.

Итак смысл: наследовать свой класс от **QFileDialog**, в конструкторе при помощи `findChild()` по имени искать контейнер в котором располагается виджеты для отображения иконок и быстрых ссылок, приводить его к нужному типу, создавать виджет для предпросмотра, добавлять его в контейнер, при смене файла обновлять содержимое.

Для того, что бы узнать какой контейнер используется (что бы произвести приведение типа) и его имя, нам потребуются исходники Qt, которые скачиваем с сайта Nokia: <http://qt.nokia.com>, распаковываем, находим файл `qfiledialog.ui` и загружаем его в Qt Designer. После пары минут изучения находим, что контейнер это компонент типа **QSplitter** и имя у него - `splitter`. Всё просто и лаконично.

Теперь создаём свой класс, наследуем его от **QFileDialog**

```
class MFileDialog : public QFileDialog
```

В теле класса обязательно ставим

```
Q_OBJECT
```

в секции `private`: создаем прототип функции `init()` - её будем вызывать из конструктора (если вы решитесь объявить их несколько), и указатель на экземпляр класса **QLabel** - его будем использовать для вывода миниатюры изображения:

```
private:  
    void init();  
    QLabel *_preview;
```

Да, я обычно использую для методов класса одну секцию `private/public/etc`, а для переменных - другую. Так получается нагляднее.

Далее, сразу вспоминаем, что нам нужно обновлять предпросмотр при смене файла, для этого создадим слот - обработчик сигнала `currentChanged(QString)`:

```
public slots:  
    void fileChanged(const QString &file);
```

Реализации методов `init()` и `fileChanged(QString &file)` в листинге ниже:

```
void MFileDialog::init()
{
    _preview = ;

    // HACK: original file dialog in Qt4 does not have preview functionality
    // Work well on 4.7.1
    QSplitter *splitter = findChild<QSplitter*>("splitter");
    if (splitter == )
    {
        return;
    }

    _preview = new QLabel();
    QRect geomerty = _preview->geometry();
    geomerty.setWidth(160); // Настройки геометрии, можно так же сделать что бы
рамочка отображалась или что-то вроде
    _preview->setGeometry(geomerty);

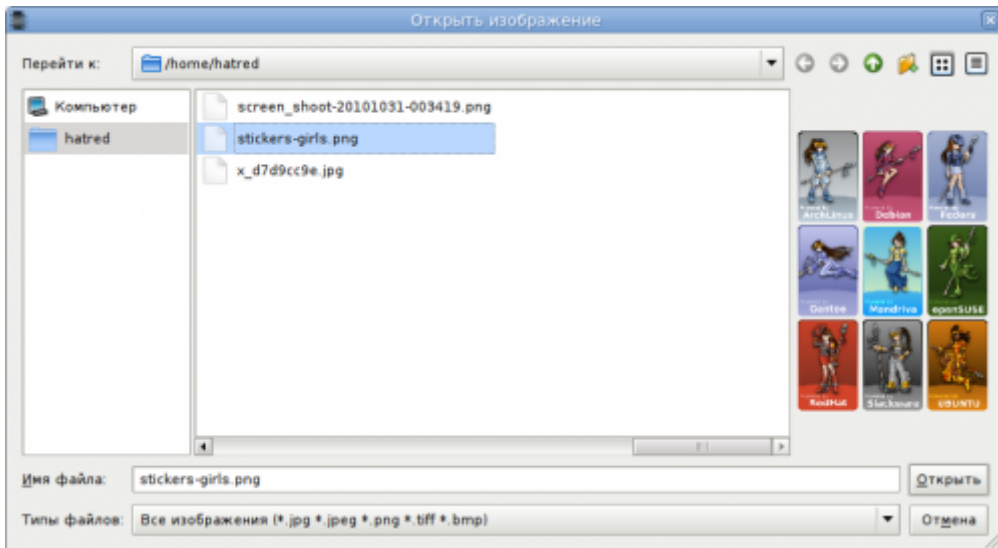
    splitter->addWidget(_preview); // Собсвенно, добавляем наш виджет в контейнер

    // подключаем обработчик сигнала currentChanged
    connect(this, SIGNAL(currentChanged(QString)),
            this, SLOT(fileChanged(QString)));
}

void MFileDialog::fileChanged(const QString &file)
{
    if (_preview == )
    {
        return;
    }

    QPixmap pix(file);
    if (pix.isNull())
    {
        _preview->setPixmap(QPixmap());
    }
    else
    {
        // Не забываем масштабировать изображение к размеру нашей области просмотра
        QSize size = _preview->size();
        pix = pix.scaled(size, Qt::KeepAspectRatio);
        _preview->setPixmap(pix);
    }
}
```

Не забываем в конструкторе вызвать `init()`, а в деструктор поместить `delete _preview`. И всё, примерно так это выглядит:



При этом, если даже не будет найден разделитель с таким именем, диалог не сломается - просто не будет отображаться окошко предпросмотра.

Какие улучшения тут можно сделать? Я думаю как минимум такие:

1. Установление ограничения на размер файла для которого будет автоматически делаться предпросмотр, ибо загрузка занимает время, а файлы могут быть очень большими (у меня, к примеру, есть такие которые не отрываются по причине нехватки памяти).
2. Вместо **QLabel** использовать **QToolButton** и если файл слишком большой обрабатывать нажатие и делать принудительный предпросмотр, либо какие-то другие действия.

PS Кому нужны исходники, пишите на мыло.

From: <https://htrd.su/wiki/> - **Hatred's Log Place**

Permanent link: [https://htrd.su/wiki/zhurnal/2010-11-01\\_21.40\\_qfiledialog\\_s\\_image\\_preview\\_metod\\_xakera](https://htrd.su/wiki/zhurnal/2010-11-01_21.40_qfiledialog_s_image_preview_metod_xakera)

Last update: **2011-10-31 11:00**

