

Възможности на Qt 4.7 и Qt Quick

Николай Николов

Qt кот



- Мултиплатформена библиотека за разработка на конзолни и графични приложения.
- Основен език C++
- Портове: Python, Ruby, PHP, Perl, Ada, Java и др.
- Лиценз: GPL 3.0, LGPL 2.1 и комерсиален лиценз.
- Използва се в редица големи проекти:

Autodesk, Google Earth, KDE, Adobe Photoshop Album, European Space Agency, Skype, VLC, VirtualBox и др.



Скоро се очаква поддръжка и на:

- iPhone OS
- Android
- webOS
- Qt за Wayland

- Qt Designer
- Qt Creator
- Qt Assistant
- Qt Linguist
- qmake
- Qt Simulator
- Visual Studio Add-In
- Qt Eclipse Integration

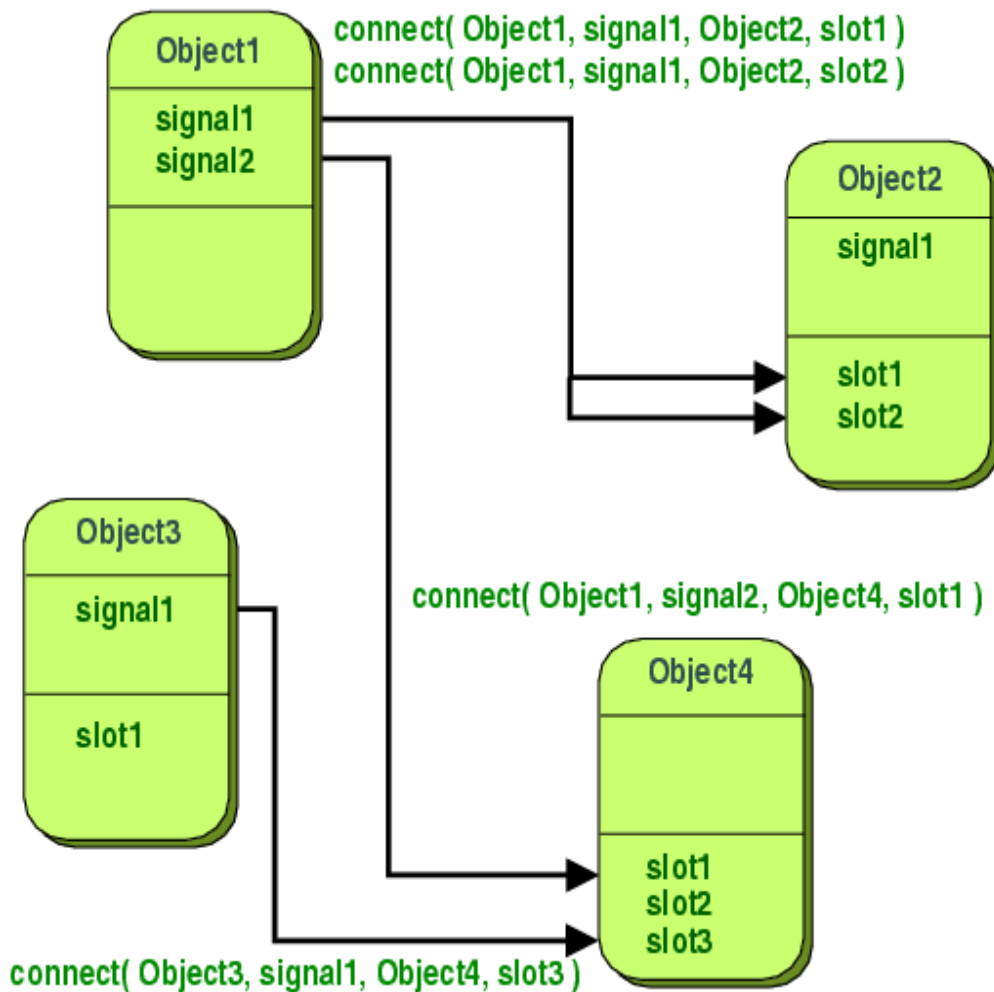
- QtCore
- QtGui
- QtMultimedia
- QtNetwork
- QtOpenGL
- QtOpenVG
- QtScript
- QtScriptTools
- QtSql
- QtSvg
- QtWebKit
- QtXml
- QtXmlPatterns
- Phonon
- Qt3Support
- QtDeclarative
- QtDBus



Code less.
Create more.
Deploy everywhere.

- QObject
- Meta Object Compiler
- Signal/Slots
- Property
- други

```
connect( pObject, SIGNAL( some_signal() ), pOtherObject, SLOT( do_something_cool() ) )
```



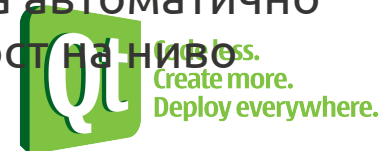
Code less.
Create more.
Deploy everywhere.

Qt Quick е колекция от технологии, които са разработени за да помогнат на разработчиците и дизайнерите да създават интуитивни, модерни и приятно изглеждащи интерфейси, каквито все повече се използват в мобилните телефони, плейъри и други мобилни устройства.

Qt Quick се състои от богат избор от графични елементи, декларативен език за описване на интерфейса и интерпретатор. Използва се колекция от C++ API-та, за да се интегрират тези елементи към стандартните Qt приложения.

Потребителският интерфейс и неговото държание се описва с QML, разширение на JavaScript, което позволява на разработчиците и дизайнерите да използват декларативен синтаксис. QML елементите са богат набор от графични и поведенчески блокове, които могат да се комбинират в QML документи, за да изградят компоненти, вариращи от прости бутони до цели приложения с достъп до Интернет.

QML подобрява интеграцията между JavaScript и съществуващата в Qt QObject-базирана система от типове, добавя поддръжка за автоматично свързване на свойствата и предоставя мрежова прозрачност на ниво програмен език.



Button.qml

```
import Qt 4.7

Rectangle {
    property alias text: textItem.text

    width: 100; height: 30
    border.width: 1
    radius: 5
    smooth: true

    gradient: Gradient {
        GradientStop { position: 0.0; color:
"darkGray" }
        GradientStop { position: 0.5; color: "black" }
        GradientStop { position: 1.0; color:
"darkGray" }
    }

    Text {
        id: textItem
        anchors.centerIn: parent
        font.pointSize: 20
        color: "white"
    }
}
```

application.qml

```
import Qt 4.7

Column {
    spacing: 10

    Button { text: "Apple" }
    Button { text: "Orange" }
    Button { text: "Pear" }
    Button { text: "Grape" }
}
```



Code less.
Create more.
Deploy everywhere.

- Property binding
- Подреждане
- State
- Transition
- Animation
- Behavior

- OpenGL модул
- QML 3D
- QtComponents
- scenegraph