

QtQuick Training Course



Module Qt Mobility

Objectives

1 Qt Mobility

What is Qt Mobility?

Setting up your environment

2 Mobility QML plugins

System Information

Location

Sensors

Objectives

3 Creating a demo

Starting point

System Information

Location

Running the demo

Topics

- 1 Qt Mobility
- 2 Mobility QML plugins
- 3 Creating a demo
- 4 Questions
- 5 Lab

What is Qt Mobility?

Locate services and network connections

Camera and microphone access

Multimedia features

Geolocation support

Phone status access

Qt Mobility QML API

List of QML plugins for QtMobility

Contacts

Location

Gallery

Sensors

Multimedia

System Information

Messaging

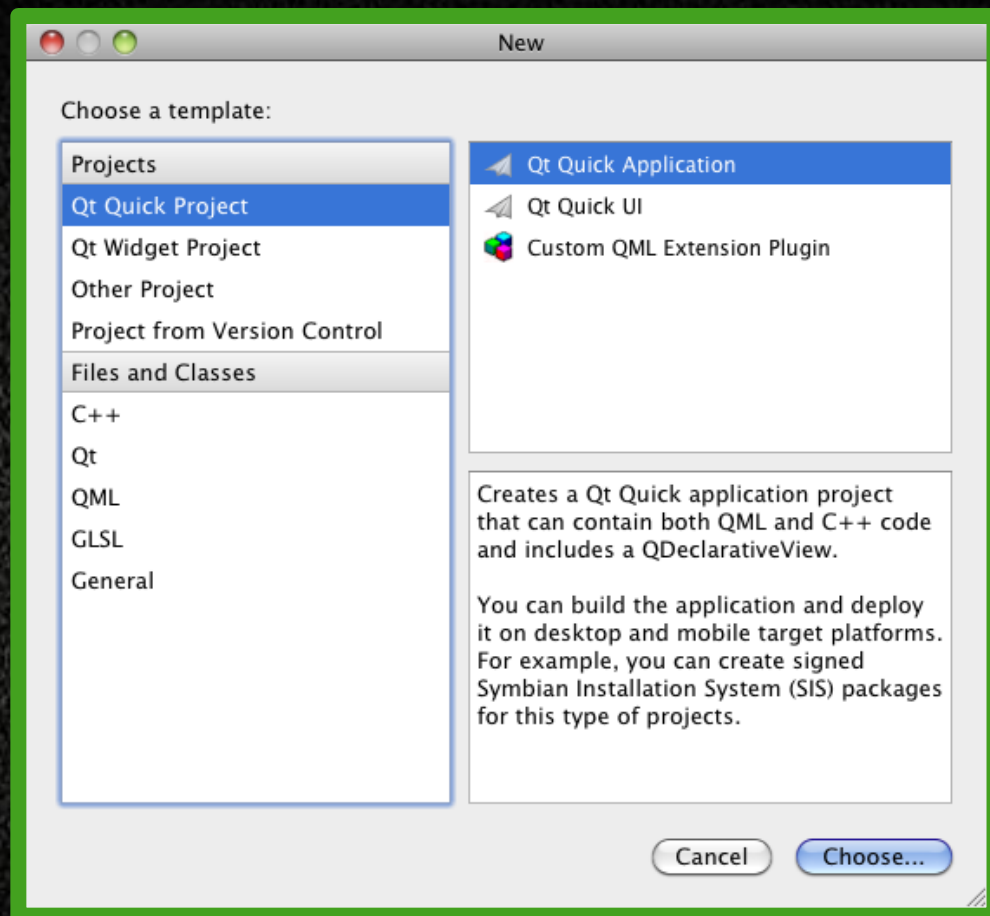
... and more

In this module, you will study **Location**, **Sensors** and **System Information**

For you to know more about the others go to: <http://doc.qt.nokia.com/qtmobility-1.2.0-beta1/qml-plugins.html>

Setting up your environment

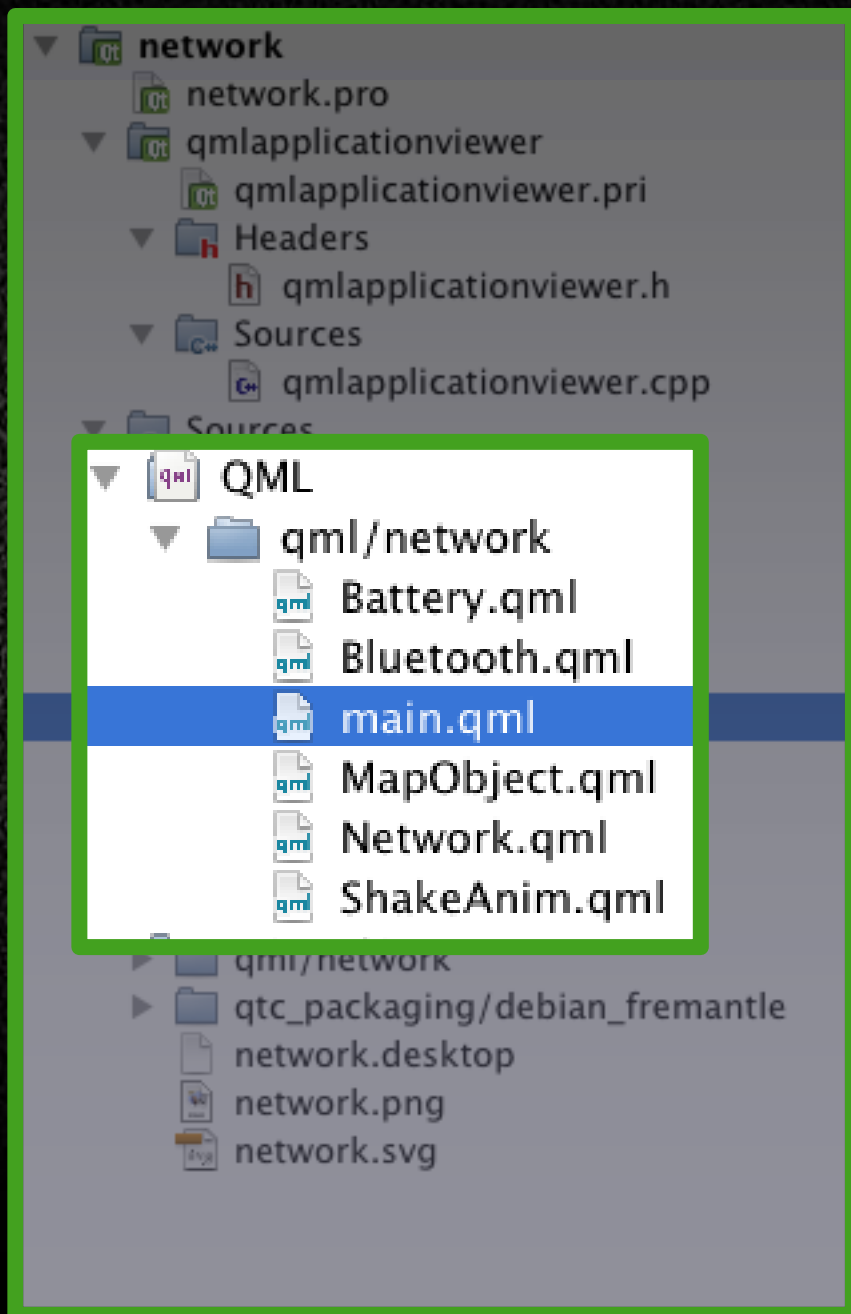
Qt Mobility is included on Qt SDK



Create a Qt Quick Application Project, if you want to use QtMobility API

Setting up your environment

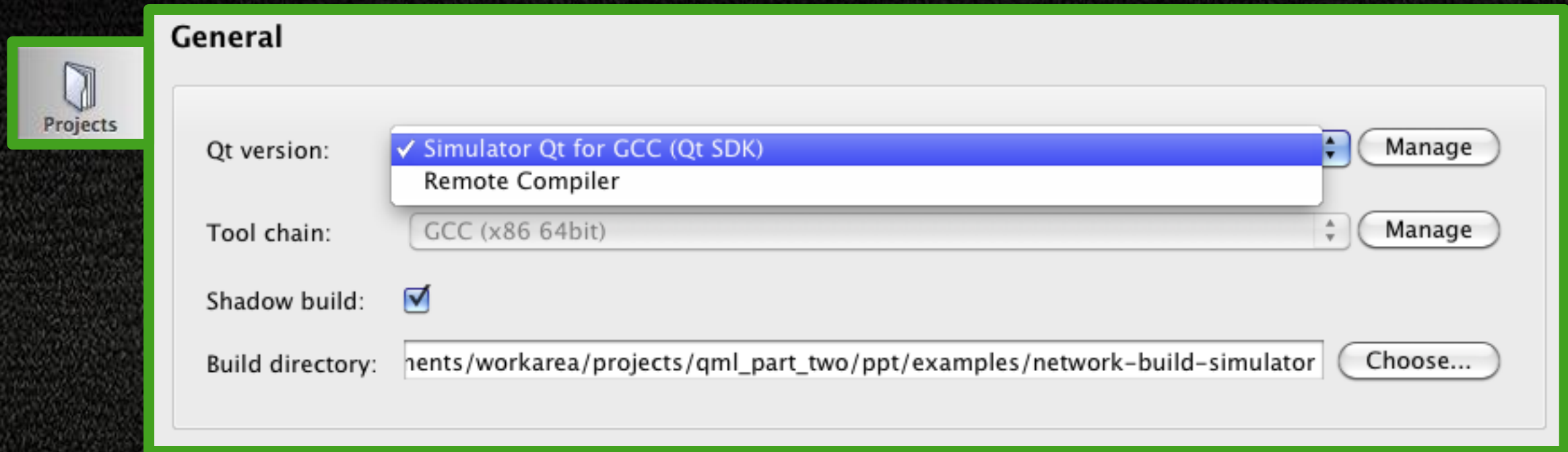
Custom QMLViewer using C++



The QML folder will contain the files you will have to change. The other folders and files are there to manage the application viewer

Setting up your environment

Make sure that your Qt Version is properly configured



You must configure your project to run on the Qt Simulator in order to test the Qt Mobility API

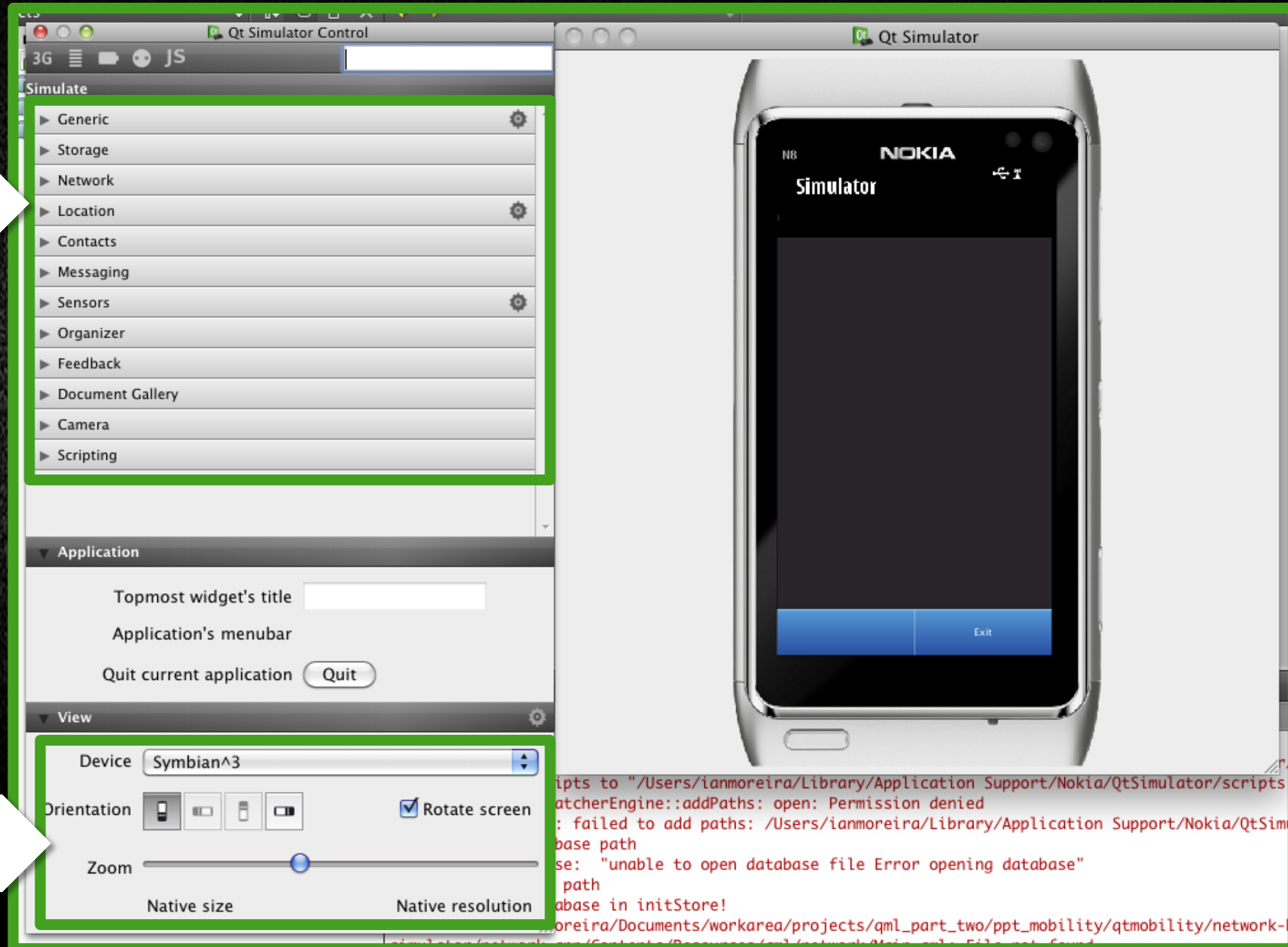
Setting up your environment

Understanding Qt Simulator

Qt Mobility API controller

This menu exposes all elements that you can change to test your mobility project

Device Visualization



Topics

- 1 Qt Mobility
- 2 Mobility QML plugins
- 3 Creating a demo
- 4 Questions
- 5 Lab

System Information



State and type of network connection



Wi-fi status



Bluetooth, FM radio, etc.



GPS connection



Battery status and power state

To know more about System Information:

<http://doc.qt.nokia.com/qtmobility/qml-systeminfo.html>

Location



Latitude, longitude and altitude coordinates



Positioning with speed and timestamp



World map display

To know more about Location:

<http://doc.qt.nokia.com/qtmobility/location-overview.html>

Sensors



Accelerometer, orientation, rotation and proximity sensors



Compass



Ambient light sensor

To know more about Sensors:

<http://doc.qt.nokia.com/qtmobility/sensors-api.html>

Topics

- 1 Qt Mobility
- 2 Mobility QML plugins
- 3 Creating a demo
- 4 Questions
- 5 Lab

Creating a demo

Starting point



See example: addon/module-009/examples/start/automotive.pro

Creating a demo

System information

Accessing

```
import QtQuick 1.0
import QtMobility.systeminfo 1.2
```

Import SystemInfo API

```
4  ▼ Image {
5      source: "images/status_bar.png"
6
7  ▼   NetworkInfo {
8      id: gsminfo
9      mode: NetworkInfo.GsmMode;
10  }
11
12 ▼   Text {
13     x: 10
14     text: gsminfo.networkName
15     color: "white"
16  }
17
```

Read network connection information
(gsminfo is referred to, afterwards)

Display the network name

See example: [addon/module-009/examples/systeminfo/qml/automotive/StatusBar.qml](#)

Creating a demo

System information

Preparing icons

```
Item {  
    height: childrenRect.height  
    width: childrenRect.width  
    property bool active: true  
  
    Image {  
        visible: active  
        source: "images/status_icon_bluetooth.png"  
    }  
}
```

Dynamic visibility

See example: [addon/module-009/examples/systeminfo/qml/automotive/Bluetooth.qml](#)

```
Item {  
    height: childrenRect.height  
    width: 27  
    property int signalStrength: 50  
  
    Item {  
        y: 2  
        width: 12 + (signalStrength * 15)/100  
        height: 11  
        clip: true  
        Image {  
            source: "images/status_icon_signal.png"  
        }  
    }  
}
```

Connection level indicator

See example: [addon/module-009/examples/systeminfo/qml/automotive/Network.qml](#)

Creating a demo

System information

Preparing icons

```
Item {  
    height: childrenRect.height  
    width: childrenRect.width  
    property int levelHeight  
  
    Image {  
        source: "images/status_icon_battery.png"  
        Rectangle {  
            anchors.bottom: parent.bottom  
            anchors.bottomMargin: 1  
            anchors.left: parent.left  
            anchors.leftMargin: 1  
            width: 7  
            height: (9 * levelHeight) / 100  
            color: "#1e4909"  
        }  
    }  
}
```

Battery level indicator

See example: [addon/module-009/examples/systeminfo/qml/automotive/Battery.qml](#)

```
Network {  
    id: network  
    signalStrength: gsminfo.networkSignalStrength  
}
```

Read connection level (from gsminfo)

See example: [addon/module-009/examples/systeminfo/qml/automotive/StatusBar.qml](#)

Creating a demo

System information

Reading

```
▼ Image {  
  source: "images/status_bar.png"  
  ▶ NetworkInfo { ... }  
  ▶ Text { ... }  
  ▶ Row { ... }  
  ▼ DeviceInfo {  
    id: deviceInfo  
  
    ▼ onBatteryLevelChanged: {  
      battery.levelHeight = deviceInfo.batteryLevel  
    }  
  
    ▼ onBluetoothStateChanged: {  
      bluetooth.visibility = deviceInfo.currentBluetoothPowerState  
    }  
  }  
  
  ▼ Component.onCompleted: {  
    battery.levelHeight = deviceInfo.batteryLevel  
    bluetooth.visibility = deviceInfo.currentBluetoothPowerState  
    deviceInfo.startBatteryLevelChanged();  
    deviceInfo.startBluetoothStateChanged();  
  }  
}
```

Read information

Collect information (ongoing)

See example: [addon/module-009/examples/systeminfo/qml/automotive/StatusBar.qml](#)

Creating a demo

Location

Accessing

```
import QtQuick 1.1
import QtMobility.location 1.2

▼ Rectangle {
    anchors.fill: parent
    color: "black"

    ▼ LandmarkModel {
        id: landmarkModelAll
        autoUpdate: true
        limit: 20
        importFile: "mylm.lmx"
    }

    ► Map { ... }
}
```

Import Location API

Withdraw points of interest from database

Display a map

See example: [addon/module-009/examples/automotive/qml/automotive/MapPage.qml](#)

Creating a demo

Location

Preparing baloon

```
import QtQuick 1.0
import QtMobility.location 1.2

▼ MapGroup {
    property alias name: text.text
    property Coordinate coordinate: null

    ▼ MapImage {
        offset.x: -194
        offset.y: -216
        source: "images/map_info_baloon.png"
        coordinate: parent.coordinate
    }

    ▼ MapText {
        id: text
        color: "#C00000"
        offset.x: -90
        offset.y: -175
        font.pixelSize: 20
        coordinate: parent.coordinate
    }

    ► MapText { ... }
    ► MapText { ... }
    ► MapText { ... }
    ► MapText { ... }
}
```



Import Location API

Group map objects

Display image on map
(anchored on a specific coordinate)

Display text on map
(anchored on a specific coordinate)

See example: [addon/module-009/examples/automotive/qml/automotive/Baloon.qml](#)

Creating a demo

Location

Preparing map

```
Map {
    id: map
    plugin : Plugin { name : "nokia" }
    anchors.fill: parent
    zoomLevel: 4

    center: Coordinate {
        latitude: -27.5
        longitude: 153
    }

    MapObjectView { ... }

    Balloon {
        id: balloon
        name: "Thomas"
        coordinate: Coordinate {
            latitude: -27.5
            longitude: 150
        }
    }

    MapMouseArea { ... }
}
```

Determine map service

Initial position

Populate map with MapObjects

Initial balloon

Handle mouse events

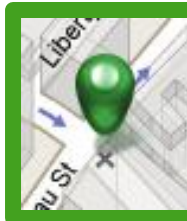
See example: [addon/module-009/examples/automotive/qml/automotive/MapPage.qml](#)

Creating a demo

Location

Populating map

```
MapObjectView {  
    model: landmarkModelAll  
    delegate: Component {  
        MapImage {  
            offset.x: -12  
            offset.y: -44  
            source: "images/pin_green.png"  
            coordinate: landmark.coordinate  
  
            MapMouseArea {  
                onClicked: {  
                    baloon.name = landmark.name  
                    baloon.coordinate = landmark.coordinate  
                }  
            }  
        }  
    }  
}
```



Determine model

Display image on coordinate
(points of interest)

Handle mouse events

See example: [addon/module-009/examples/automotive/qml/automotive/MapPage.qml](#)

Creating a demo

Location

Handling mouse events

```
MapMouseArea {
    property int lastX: -1
    property int lastY: -1

    onPressed : {
        lastX = mouse.x
        lastY = mouse.y
    }
    onReleased : {
        lastX = -1
        lastY = -1
    }
    onPositionChanged: {
        if (mouse.button == Qt.LeftButton) {
            if ((lastX != -1) && (lastY != -1)) {
                var dx = mouse.x - lastX
                var dy = mouse.y - lastY
                map.pan(-dx, -dy)
            }
            lastX = mouse.x
            lastY = mouse.y
        }
    }
    onDoubleClicked: {
        map.center = mouse.coordinate
        if (mouse.button == Qt.LeftButton)
            map.zoomLevel += 1
        else if (mouse.button == Qt.RightButton)
            map.zoomLevel -= 1
        lastX = -1
        lastY = -1
    }
}
```

Determine mouse position on click

Pan map on drag

Handle zoom on mouse events

See example: [addon/module-009/examples/automotive/qml/automotive/MapPage.qml](#)

Creating a demo

Running the demo



See video: [addon/module-009/videos/Automotive.mov](#)

Topics

- 1 Qt Mobility
- 2 Mobility QML plugins
- 3 Creating a demo
- 4 Questions
- 5 Lab

Questions

Where can you use Qt Mobility?

What plugin do you use for reading device's movements?

What's the use for the System Information plugin?

What's the use for the Location plugin?

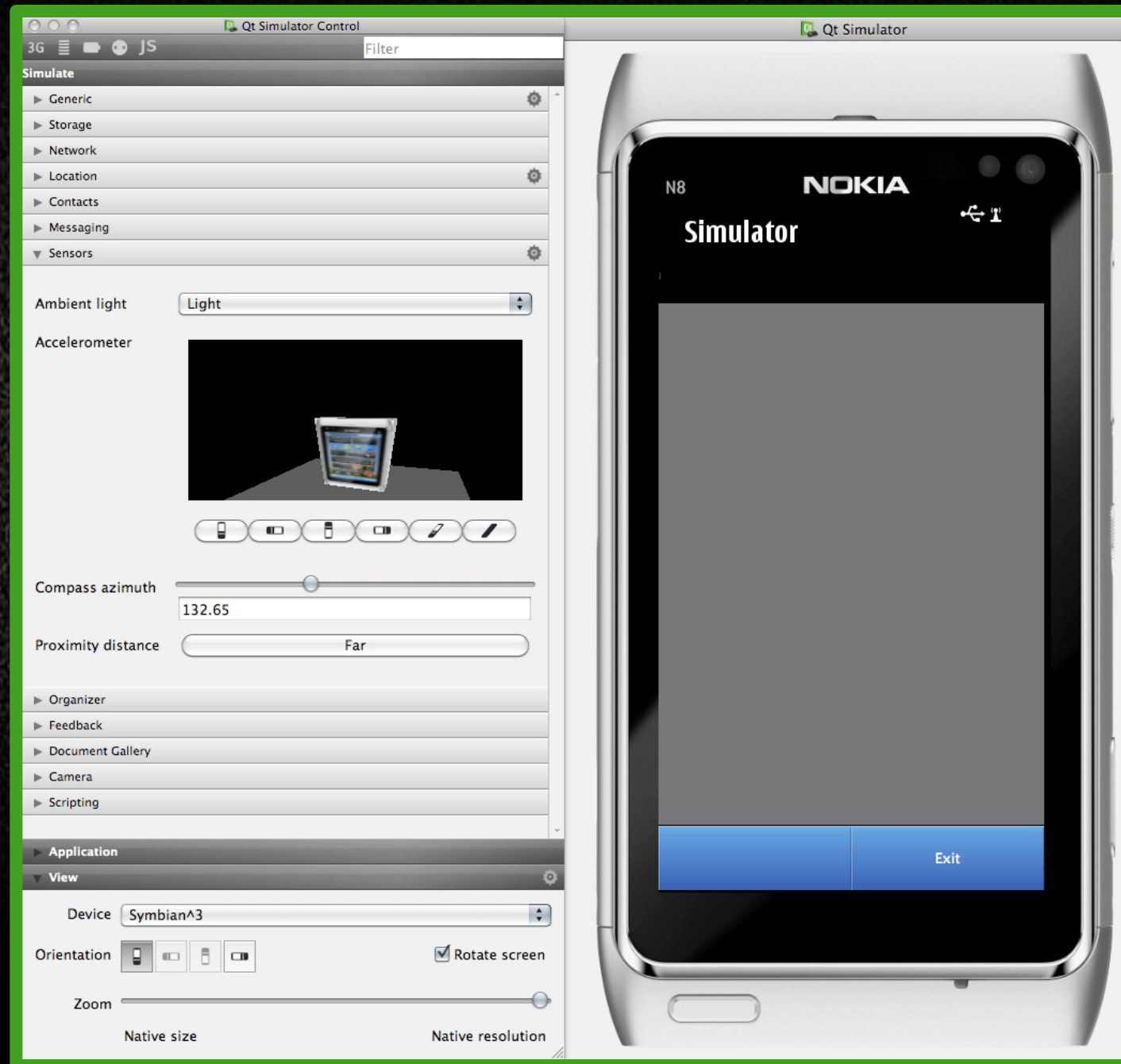
Where can you test the Mobility API?

Topics

- 1 Qt Mobility
- 2 Mobility QML plugins
- 3 Creating a demo
- 4 Questions
- 5 Lab

Lab

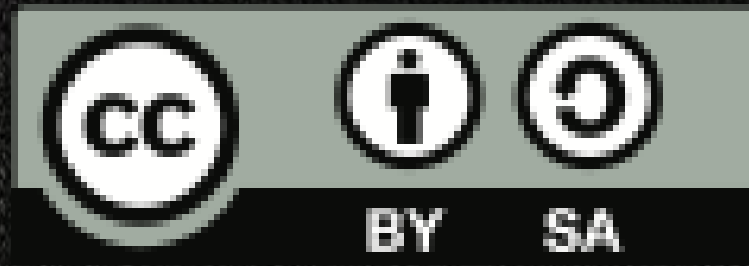
Use the documentation on <http://doc.qt.nokia.com/qtmobility/> to build a sensor that lights up when a hand gets close to it.



See video: [addon/module-009/videos/proximity.mov](#)
See lab: [labs/lab-9](#)

(c) 2011 Nokia Corporation and its Subsidiary(-ies).

The enclosed Qt Training Materials are provided under the Creative Commons Attribution ShareAlike 2.5 License Agreement.



The full license text is available here: <http://creativecommons.org/licenses/by-sa/2.5/legalcode>

Nokia, Qt and the Nokia and Qt logos are the registered trademarks of Nokia Corporation in Finland and other countries worldwide.