MeeGo Qt Quick Query Dialog, Selection Dialog and MultiSelection Dialog Components

This article explains how to use QML Query Dialog, QML SelectionDialog and QML MultiSelectionDialog Element in Qt Quick Application for MeeGo Harmattan.

Introduction

This article explains how to use various type of QML Dialog Element like QML QueryDialog Element , QML SelectionDialog Element and QML MultiSelectionDialog Element in MeeGo Harmattan Qt Quick Component.

Article uses following MeeGo Harmattan Qt Quick Components

- QML PageStackWindow Element
- QML Page Element
- QML ListView Element
- QML Label Element
- QML Menu Elements
- QML ToolBar Element
- QML ToolBarLayout Element
- QML Dialog Element
- QML QueryDialog Element
- QML SelectionDialog Element
- QML MultiSelectionDialog Element
- QML Column Element
- QML ButtonRow Element

What is the Dialog

Dialogs are components for giving informations and asking for user inputs. They are shown on top of the whole screen or on top of an item, that is given with the visualParent property. Dialogs normally consist of three parts: a title, a content field (with a set of arbitrary components) and a button field. If another kind of dialog is needed, leave the title and buttons property empty and fill the content field.

Below is an example of a simple dialog:

```
// Create a simple "Hello World"-Dialog
 Dialog {
   id: myDialog
   title: Rectangle {
     id: titleField
     height: 2
     width: parent.width
     color: "red"
   }
   content:Item {
     id: name
     height: 50
     width: parent.width
     Text {
       id: text
       font.pixelSize: 22
       anchors.centerIn: parent
       color: "white"
       text: "Hello Dialog"
```


(C) Copyright Nokia 2014. All rights reserved.

```
      }
    }
    buttons: ButtonRow {
      style: ButtonStyle { }
        anchors.horizontalCenter: parent.horizontalCenter
        Button {text: "OK"; onClicked: myDialog.accept()}
      }
    }
```

QML QueryDialog Element

This dialog is used to ask a question from the user. It is composed of a title (which consists of a title text, an icon or both), a message and buttons for accepting and rejecting. You can check the MeeGo 1.2 Harmattan Developer Documentation for this QML QueryDialog Element here.

Here is the sample code to use QML QueryDialog Element in Qt Quick Application.

```
     QueryDialog {
         id: queryDialog
         icon: "image://theme/icon-l-contacts"
         titleText: "Query Dialog Sample"
         message: "Press accept or reject button"
         acceptButtonText: "Accept"
         rejectButtonText: "Reject"
         onAccepted: // some functionality
         onRejected: // some functionality
     }
```

You can Open Query Dialog via the open() call. Here we define Open() event on Button's OnClicked event.

```
    Button {
           text: "Query Dialog"
           width: 300
           onClicked: {
               queryDialog.open();
           }
    }
```

You can define what to do if user have clicked on "Accepted" or "Rejected" on QueryDialog's **onAccepted** and **onRejected** Properties respectively. Below is one simple example code.

```
         onAccepted: labelQueryResult.text = "Result: Accepted";
         onRejected: labelQueryResult.text = "Result: Rejected";
```

Screenshot of QueryDialog:

(C) Copyright Nokia 2014. All rights reserved.

QML SelectionDialog Element

The QML SelectionDialog is a dialog that allows the user to select a single item from several list elements. You can check the MeeGo 1.2 Harmattan Developer Documentation for this QML SelectionDialog Element here.

Here is the sample code to create QML SelectionDialog Element in Qt Quick Application.

```
// Create a selection dialog with a title and list elements to choose from.
 SelectionDialog {
     id: singleSelectionDialog
     titleText: "Single Selection Dialog Header"
     selectedIndex: 0
     model: ListModel {
         ListElement { name: "ListTitle #1" }
         ListElement { name: "ListTitle #2" }
         ListElement { name: "ListTitle #3" }
     }
   onSelectedIndexChanged:
         {
         // do something here on change of selected item
         }

 }
```

You can Open SingleSelection dialog via the open() call. Here we define Open() event on Button's OnClicked event.

```
Button {
     text: "SingleSelection"
     onClicked: {
         singleSelectionDialog.open();
     }
 }
```

You can get the selected item index by using selectedIndex property.

You can get the selectedIndex name by using.

```
singleSelectionDialog.model.get(singleSelectionDialog.selectedIndex).name
```

Screenshot of SelectionDialog:

(C) Copyright Nokia 2014. All rights reserved.

QML MultiSelectionDialog Element

The MultiSelectionDialog is a dialog that allows the user to pick multiple items from available options. You can check the MeeGo 1.2 Harmattan Developer Documentation for this QML MultiSelectionDialog Element here.

Here is the sample code to create QML MultiSelectionDialog Element in Qt Quick Application.

```
// Create a multi-selection dialog with a title and list elements to choose from.
 MultiSelectionDialog {
     id: multiSelectionDialog
     titleText: "Multi Selection Dialog Header"
     selectedIndexes: [1, 3, 5]
     model: ListModel {
         ListElement { name: "ListTitle #1" }
         ListElement { name: "ListTitle #2" }
         ListElement { name: "ListTitle #3" }
         ListElement { name: "ListTitle #4" }
         ListElement { name: "ListTitle #5" }
     }
     acceptButtonText: "OK"
 }
```

You can Open MultiSelection Dialog via the open() call. Here we define Open() event on Button's OnClicked event.

```
Button {
     text: "MultiSelection"
     onClicked: {
         multiSelectionDialog.open();
     }
 }
```

You can get name of the selected ListElements from the model via the selectedIndexes property.

You can get the selectedIndex name by using.

```
multiSelectionDialog.model.get(multiSelectionDialog.selectedIndexes[index]).name
```

Sample Exampe Code

I have created sample code for how to use various Dialogs in Qt Quick Harmattan. Here is the source code.

MainPage.qml (Page)

```
import QtQuick 1.1
import com.nokia.meego 1.0
```


(C) Copyright Nokia 2014. All rights reserved.

```
Page {
    id: mainPage
    tools: commanTools
    QueryDialog {
        id: queryDialog
        icon: "image://theme/icon-l-contacts"
        titleText: "Query Dialog Sample"
        message: "Press accept or reject button"
        acceptButtonText: "Accept"
        rejectButtonText: "Reject"
        onAccepted: labelQueryResult.text = "Result: Accepted";
        onRejected: labelQueryResult.text = "Result: Rejected";
    }
    SelectionDialog {
        id: singleSelectionDialog
        titleText: "Single Selection Dialog Header"
        selectedIndex: 0
        model: ListModel {
            ListElement { name: "ListElement #1" }
            ListElement { name: "ListElement #2" }
            ListElement { name: "ListElement #3" }
            ListElement { name: "ListElement #4" }
            ListElement { name: "ListElement #5" }
            ListElement { name: "ListElement #6" }
            ListElement { name: "ListElement #7" }
            ListElement { name: "ListElement #8" }
            ListElement { name: "ListElement #9" }
            ListElement { name: "ListElement #10" }
            ListElement { name: "ListElement #11" }
            ListElement { name: "ListElement #12" }
            ListElement { name: "ListElement #14" }
            ListElement { name: "ListElement #15" }
            ListElement { name: "ListElement #16" }
            ListElement { name: "ListElement #17" }
            ListElement { name: "ListElement #18" }
            ListElement { name: "ListElement #19" }
            ListElement { name: "ListElement #20" }
        }
    }
    Column {
        id: dialogs
        spacing: 24
        Row {
            spacing: 32
            Button {
                text: "Query Dialog"
                width: 300
                onClicked: {
                    queryDialog.open();
                }
            }
            Label {
                id: labelQueryResult
                text: "Result: N/A"
            }
        }
```


(C) Copyright Nokia 2014. All rights reserved.

```
        Row {
            spacing: 32
            Button {
                text: "Single Selection Dialog"
                width: 400
                onClicked: {
                    singleSelectionDialog.open();
                }
            }
            Label {
                text:
singleSelectionDialog.model.get(singleSelectionDialog.selectedIndex).name
            }
        }
      }
}
```

Main.qml (PageStackWindow)

```
import QtQuick 1.1
import com.nokia.meego 1.0
PageStackWindow {
    id: appWindow
    initialPage: mainPage
    MainPage {
        id: mainPage
    }
    ToolBarLayout {
        id: commanTools
        ToolIcon { iconId: "toolbar-back"; onClicked: { colorMenu.close();
pageStack.pop(); }  }
        ToolIcon { iconId: "toolbar-view-menu" ; onClicked: colorMenu.open(); }
    }
    Menu {
        id: myMenu
        visualParent: pageStack
        MenuLayout {
            MenuItem { text: qsTr("Sample menu item") }
        }
    }
}
```

Screenshots

(C) Copyright Nokia 2014. All rights reserved.

Source code

You can find the source code of this Code Example on File:MeeGo Qt Quick Query Dialog and Selection Dialog.zip.

(C) Copyright Nokia 2014. All rights reserved.