



<b>ID</b>		<b>Creation date</b>	27 December 2008
<b>Platform</b>	S60 3rd Edition, S60 5th Edition	<b>Tested on devices</b>	Emulator
<b>Category</b>	Qt for S60	<b>Subcategory</b>	UI

**Keywords (APIs, classes, methods, functions):** QMessageBox

## Introduction

The code given below shows means for informing the user or for asking the user a question and receiving an answer.

The QMessageBox class can be use for this purpose. It generate a message box that displays a primary text to alert the user to a situation.

The QMessageBox class provides a modal dialog for informing the user or for asking the user a question and receiving an answer. You can use QMessageBox to show information/warning message to user. QMessageBox has functionality to set title of message dialog. QMessageBox allows to show message with icon (allow both type of icon i.e. predefined and custom).

## Download Links

- Download and Install latest version [Qt for Symbian - Installation packages](#) which has links on how to install the latest version

## Various Function

- To set Window Title.

```
msgBox.setWindowTitle("HELP");
```

## How\_to\_create\_a\_message\_box\_in\_Qt\_for\_Symbian



- This provide the informative text that provides a fuller description for the message.

```
msgBox.setInformativeText("Do You Want any Help This???");
```



- Sets the modality of the message box to windowModality.This allow to set the window modality if its true then we can't focus any other window until the msgBox is being closed.

```
msgBox.setWindowModality(Qt::ApplicationModal);
```

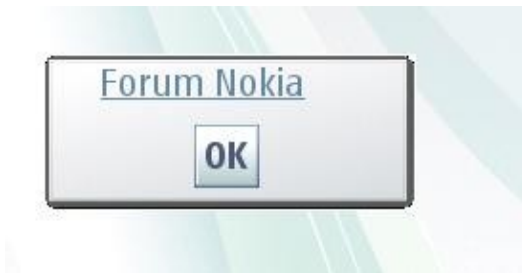
- This property holds collection of standard buttons in the message box.

```
msgBox.setStandardButtons(QMessageBox::Ok | QMessageBox::Save);
```



## Clickable URL in messageBox

```
msgbox->about(0, "Forum.nokia.com", "<a href='http://www.forum.nokia.com!'>Forum Nokia</a>");
```



## Code

```
#include <QtGui/QApplication>
#include <QMessageBox>
int main(int argc, char *argv[])
{
    QApplication app(argc, argv);
    QMessageBox msgBox;
    msgBox.setText("Hello Here");
    msgBox.exec();
    app.quit();
}
```

```
return 1;  
}
```

## Screen Shot

To read More about QMessageBox: [click here](#)



## Receiving Input from user

QMessageBox allows user to set predefined buttons (standard buttons), and then receive input when these button is clicked. **setStandardButtons()** allows to set these buttons, you can add more than one button at a time.

```
QMessageBox msgBox;  
    //Set text  
    msgBox.setText("Testing..");  
    //Set predefined icon, icon is show on left side of text.  
    msgBox.setIcon(QMessageBox::Information);  
    //set informative text  
    msgBox.setInformativeText("Just show information.");  
    //Add ok and cancel button.  
    msgBox.setStandardButtons(QMessageBox::Ok | QMessageBox::Cancel);  
    //Set focus of ok button  
    msgBox.setDefaultButton(QMessageBox::Ok);
```

## How\_to\_create\_a\_message\_box\_in\_Qt\_for\_Symbian

```
        //execute message box. method exec() return the button value of cliecke button
int ret = msgBox.exec();

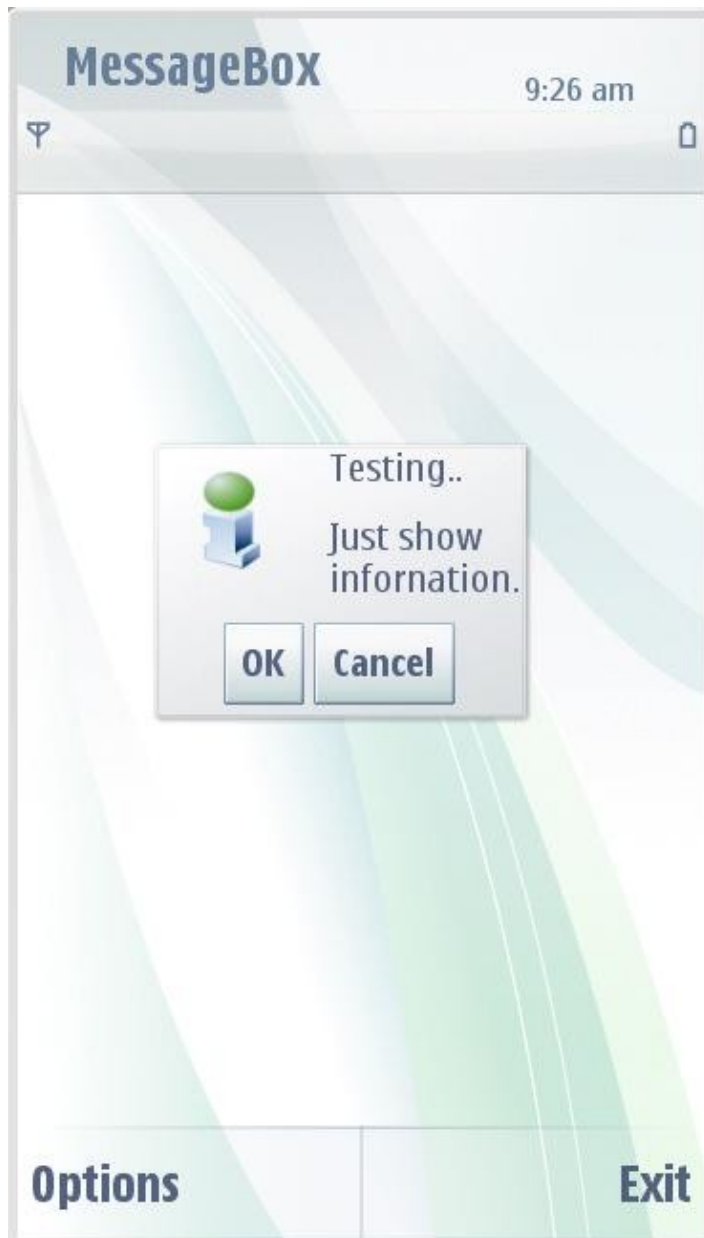
        //User get input from returned value (ret). you can handle it here.
switch (ret) {
    case QMessageBox::Save:
        // Save was clicked
        break;
    case QMessageBox::Ok:
        // ok was clicked
        break;
    default:
        // should never be reached
        break;
}
```

## Static methods of QMessageBox

QMessageBox has three static method, **warning()**, **information()** and **critical()** , which allow to create messagebox, but it does not allow to set custom icon, informative text and detailed text.

```
//Create warning message box
QMessageBox::warning(0, "Warning", "Warning message text");
//Create information message box
QMessageBox::information(0, "Information", "Information message text");
//Create critical message box
QMessageBox::critical(0, "Critical", "Critical message text");
```

## ScreenShots



## Code Example

[Code Example](#)

More about QMessageBox visit: <http://pepper.troll.no/s60prereleases/doc/qmessagebox.html>