

<b>ID</b>	...	<b>Creation date</b>	17 March 2009
<b>Platform</b>	S60 3rd Edition FP2, S60 5th Edition	<b>Tested on devices</b>	S60 Emulator
<b>Category</b>	Qt for S60	<b>Subcategory</b>	Application

**Keywords (APIs, classes, methods, functions):** QLineEdit, QStringList, QCompleter

## Overview

You can use QCompleter to provide auto completions in some of Qt widget, such as QLineEdit and QComboBox.

When the user starts typing a word, QCompleter suggests possible ways of completing the word, based on a word list. [QCompleter](#)

This example makes the use of following classes

[QCompleter](#)-Provides completions based on an item model

[QStringList](#)-Provides a list of strings

## Preconditions

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

## Various Function

- This property holds the case sensitivity of the matching.

```
completer->setCaseSensitivity(Qt::CaseInsensitive);
```

- This property holds how the completions of a specific words are provided to the user.

```
completer->setCompletionMode(QCompleter::InlineCompletion);  
completer->setCompletionMode(QCompleter::QCompleter::UnfilteredPopupCompletion);
```



- This property holds the item role to be used to query the contents of items for matching.

```
completer->setCompletionRole(3);
```

## Source Code

```
#include <QtGui/QApplication>
#include "qcomplimentwidget.h"
#include<QStringList>
#include<QLineEdit>
#include<QCompleter>
#include<QHBoxLayout>
#include<QWidget>
#include<QLabel>

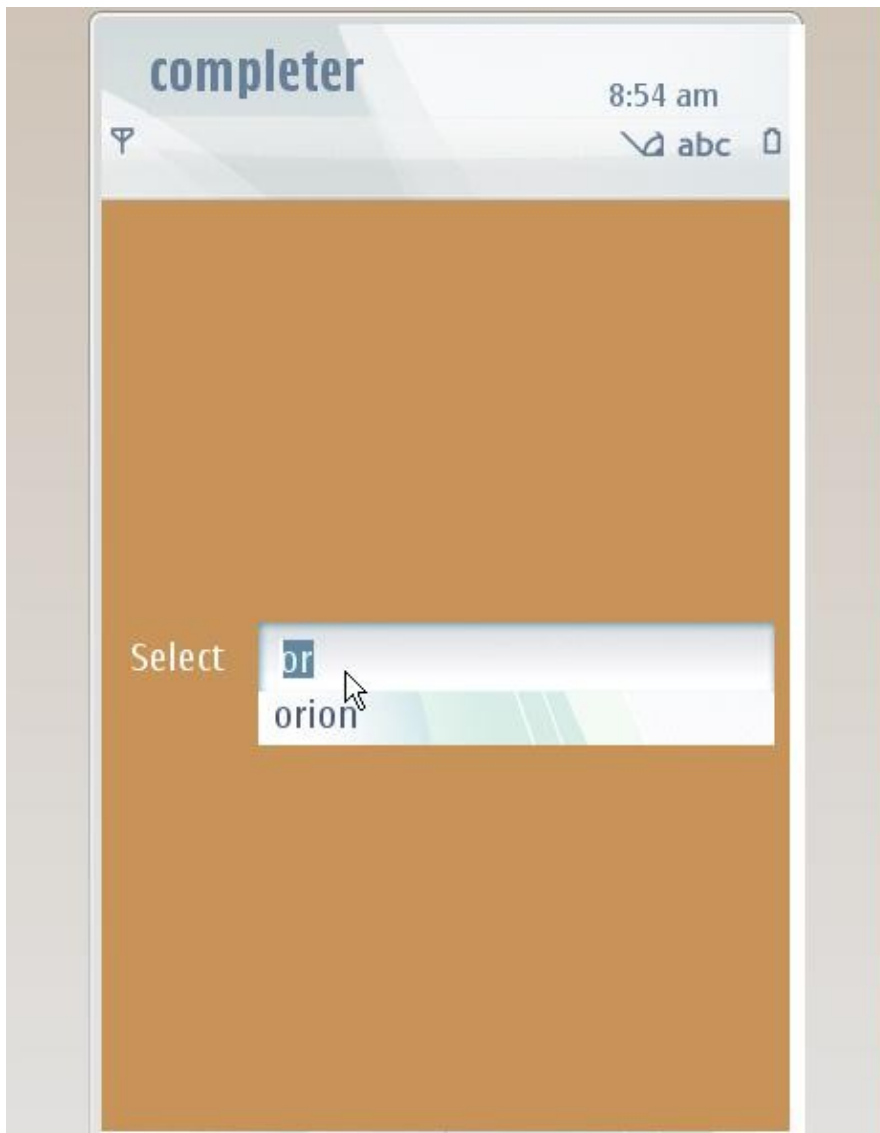
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget *win=new QWidget();
    QHBoxLayout *lay=new QHBoxLayout();
    QStringList wordList;
    wordList << "alpha" << "omega" << "omicron" << "zeta"<<"america"<<"orion"<<"amit"<<"Odssey";
    QLabel *lbl=new QLabel("Select");
    QLineEdit *lineEdit = new QLineEdit();
    lbl->setBuddy(lineEdit);
    QCompleter *completer = new QCompleter(wordList);
    completer->setCaseSensitivity(Qt::CaseInsensitive); //Make case insensitive selection
    //completer->setCompletionMode(QCompleter::InlineCompletion); //Used to enable in line search
```

## How\_To\_use\_QCompleter

```
//completer->setCompletionRole(3);  
lineEdit->setCompleter(completer);  
lay->addWidget(lbl);  
lay->addWidget(lineEdit);  
win->setLayout(lay);  
win->showMaximized();  
win->setStyleSheet("* { background-color:rgb(199,147,88); padding: 7px ; color:rgb(255,255,255);  
return a.exec();  
}
```

## Screenshot

More about QCompleter:[QCompleter](#)



More about QCompleter:[QCompleter](#)