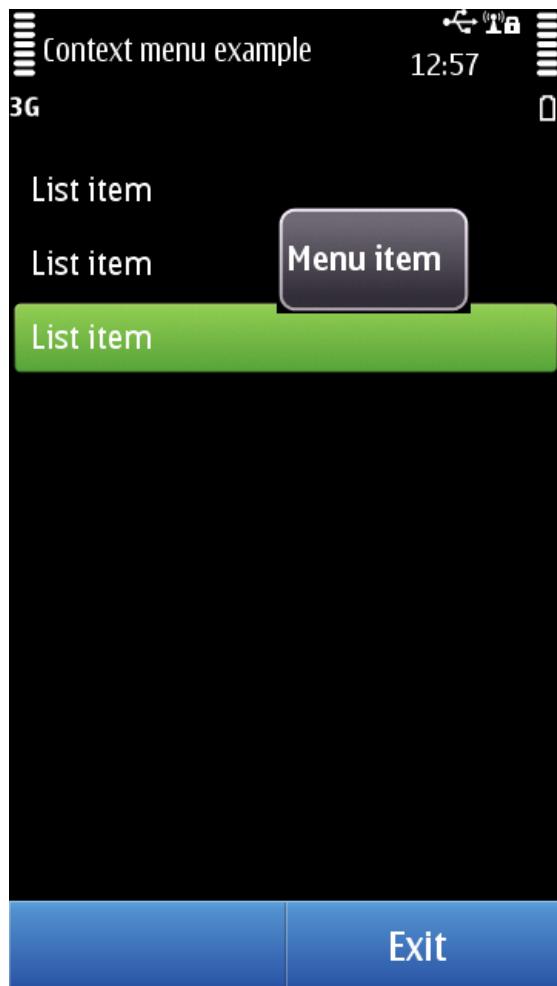


Archived:Context menus in Qt

 Archived: This article is **archived** because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user~~~~|write your reason here}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (based on [QWidget](#)) is deprecated.

This article shows how to define context menu for the [QWidget](#) using a [QListWidget](#).



Defining context menus

```
#include <QtGui>
class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = 0);
    ~MainWindow();

private:
    void createContextMenu();

private slots:
    void showContextMenu(const QPoint&);
```

```
private:  
    QMenu* m_contextMenu;  
    QListWidget* m_list;  
};
```

```
MainWindow::MainWindow(QWidget *parent)  
    : QMainWindow(parent)  
{  
    m_list = new QListWidget(this);  
    new QListWidgetItem("List item",m_list);  
    new QListWidgetItem("List item",m_list);  
    new QListWidgetItem("List item",m_list);  
    this->setCentralWidget(m_list);  
  
#ifdef Q_OS_SYMBIAN  
// Remove context menu from all widgets  
QWidgetList widgets = QApplication::allWidgets();  
QWidget* w = 0;  
foreach(w,widgets){  
    w->setContextMenuPolicy(Qt::NoContextMenu);  
}  
#endif  
  
#if defined Q_OS_SYMBIAN || defined Q_WS_MAEMO_5  
createContextMenu();  
#endif  
}  
  
MainWindow::~MainWindow()  
{  
}  
  
void MainWindow::createContextMenu()  
{  
    // Define context menu for the QListWidget  
    m_list->setContextMenuPolicy(Qt::CustomContextMenu);  
  
    // Connect to customContextMenuRequested signal  
    QObject::connect(m_list, SIGNAL(customContextMenuRequested(const QPoint&)),  
                     this, SLOT(showContextMenu(const QPoint&)));  
  
    // Context menu for QListWidget  
    m_contextMenu = new QMenu(this);  
  
    // Create one action item  
    QAction* menuItem = new QAction("Menu item",0);  
    //QObject::connect(menuItem, SIGNAL(triggered()), this, SLOT(someSlot()));  
    m_contextMenu->addAction(menuItem);  
}  
  
void MainWindow::showContextMenu(const QPoint& pos)  
{  
    // If statement is workaround for QTBUG-6256  
    if (pos != QPoint(0,0)) {
```

```
    // Execute context menu
    m_contextMenu->exec(pos);
}
}
```

QTBUG-6256

If a widget receives `setContextMenuPolicy(Qt::CustomContextMenu)` it gets an entry in the softkeys Options menu called "Actions", which when triggered emits `QWidget::customContextMenuRequested()`. However, the flag `Qt::CustomContextMenu` is not spec'd to cause that, `Qt::ActionsContextMenu` is. See more from [QTBUG-6256](#)

Postconditions

Holding stylus on QListWidget opens defined context menu.

See also

- [QWidget::contextMenuPolicy\(\)](#)
- [Qt::CustomContextMenu](#)
- [QTBUG-6256](#)