Archived:How to use QStatusBar

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 create a QWidget based app with a status bar.

Overview

The QMainWindow class allows the creation of a main window which provides a horizontal bar suitable for presenting status information. This bar lies on the bottom of window and can be achieved by calling the function `statusBar()`. This function creates and returns an object of type QStatusBar, if it does not exist.

The *QStatusBar* class allows addition of temporary or permanent messages and addition of QLabel, QProgressBar or QToolButton widgets.

Preconditions

To use `QStatusBar` class we need to include `QStatusBar` header file:

```
#include <QStatusBar>
```

Example

Considering that we have a `QMainWindow`, let's add a temporary message (for ten seconds):

```
QStatusBar *status;
status = statusBar();
status->showMessage("this is the status bar", 10000);
```

Optionally, let's add a permanent widget (*QLabel* in this case):

```
QStatusBar *status;
QLabel *label = new QLabel(this);
label->setText("this the status bar");
status = statusBar();
status->addPermanentWidget(label, 200);
status->show();
```

Postconditions

On Maemo, this will display the following image when run:

External links

- QMainWindow
- QStatusBar

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.