

Archived:Listening for screen orientation changes 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 (using C++ for the Qt app UI) is deprecated.

Overview

This code snippet demonstrates how to listen for screen or layout orientation changes. When using Qt layout managers, they handle the layout changes and you generally do not need to worry about them.

On custom UI applications that do not use layout managers, you have to handle screen orientation and size changes by implementing a virtual `QWidget::resizeEvent()`.

The event handler `resizeEvent()` of the base class `QWidget` is called whenever the user changes the screen mode. This event handler can be reimplemented in a subclass to receive widget resize events which are passed in the event parameter. `QMainWindow` is a subclass of `QWidget` and it contains `resizeEvent()` as well, so you can also get the resize event from `QMainWindow` if you implement it in your application.

This snippet can be self-signed.

Preconditions

- Install Qt SDK

Source

Solution 1: Requesting the new size from `QResizeEvent`

```
void QMyWidget::resizeEvent (QResizeEvent* event)
{
    QSize widgetSize = event->size();
    // TODO: You have new size of the screen
    // do your new layout

    // Call base class impl
    QWidget::resizeEvent(event);
}
```

Solution 2: Requesting the new size from `QMainWindow`'s central widget

```
void QMyMainWindow::resizeEvent (QResizeEvent* event)
{
    QSize widgetSize = centralWidget()->size();
    // TODO: You have new size of the screen
    // do your new layout

    // Call base class impl
    QMainWindow::resizeEvent(event);
}
```

See also

[Archived:Implementing a layout manager in Qt](#)

[Archived:Dynamic Layout handling with QWidget](#)

Postconditions

Layout orientation and size change are handled. You will get a notification when the user changes the screen mode.

