ID		Creation date	May 29, 2009
Platform	S60 3rd Edition, FP1, FP2 S60 5th Edition	Tested on devices	
Category	Qt for Symbian	Subcategory	

Keywords (APIs, classes, methods, functions): QPainterPath ,QRegion

Overview

This code snippet shows how to display widgets in rounded rectangle borders with transparency effect.

This can be done by two simple steps

1. Clip the widget to a rounded rectangle 2. Mask the widget with the clipped region

For this we need to implement QWidget::paintEvent() of your widget

Note: In order to use this code, you need to have Qt for S60 installed on your platform.

Preconditions

- Download and Install latest version Qt for Symbian - Installation packages which has links on how to install the latest version

Code

```
void MyRoundedWidget::paintEvent(QPaintEvent *aPaintEvent)
{

        qreal(0.75);
int roundness(12);
  QRect widget_rect = this->rect();

        QPainter(painter);
        painter.
        setRenderHint(QPainter::Antialiasing);
  painter.setPen(Qt::red);

  // clip
```

```
        QPainterPath rounded_rect;
        rounded_rect.addRoundRect(1, 1, widget_rect.width() - 2, widget_rect.height() - 2, roundness, rou
        painter.setClipPath(rounded_rect);

// get clipping region
  QRegion maskregion = painter.clipRegion();

  // mask the widget
  setMask(maskregion);
  painter.setOpacity(opacity);

  // fill path with color
  painter.fillPath(rounded_rect,QBrush(Qt::black));

  // restore painter
  painter.restore();

}
```

First create a painter path of a rounded rectangle

```
QPainterPath rounded_rect;
        rounded_rect.addRoundRect(1, 1, widget_rect.width() - 2, widget_rect.height() - 2, roundness, rou
```

Clip the widget to the created painter path which is a rounded rectangle. By clipping we are directing the painter to draw in the clipped region

```
painter.setClipPath(rounded_rect);
```

Now mask the widget to the clipped region after which it shows the rectangle with rounded edges only

```
// get clipping region
  QRegion maskregion = painter.clipRegion();

  // mask the widget
  setMask(maskregion);
```

If you dont want transparency effect set the opacity to 1.0

```
qreal opacity(1.0);
```