



Contents

- [1 Overview](#)
- [2 Source code: PopUpTextBoxMIDlet.java](#)
- [3 Source code: PopUpTextBox.java](#)
- [4 Source code: PopUpTextBoxCanvas.java](#)
- [5 Example application](#)
- [6 See also](#)

Overview

One of the Displayables in LCDUI is TextBox (extending Screen class), which allows user to enter and edit text. It is commonly used for entering relatively short texts, even single words. In any case TextBox has used the whole screen, which has made user experience bad. Now in S60 5th Edition new mode of pop-up TextBox is introduced. By using a JAD attribute "Nokia-UI-Enhancement" with value "PopUpTextBox" all the TextBox screens are shown as smaller dialogs, without obscuring the underlying screen.

```
Nokia-UI-Enhancement: PopUpTextBox
```

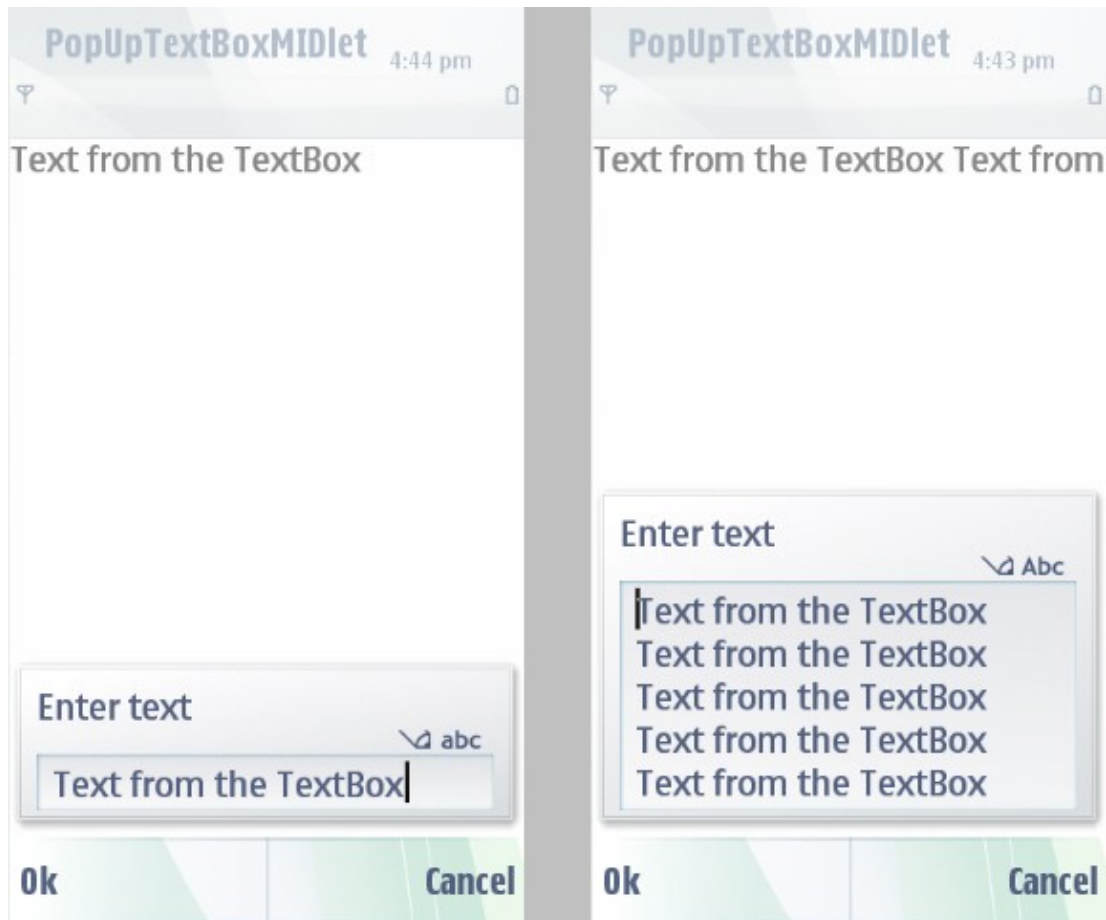
Pop-up TextBox does lack have some properties of "traditional" TextBox:

- Ticker is not visible
- Text input capacity indicator is not supported

An empty Pop-up TextBox has one line, but if needed, its size will grow. The exact maximum amount of visible lines depends on the screen size. In nHD screens (640x360 pixels) it is 5 rows of text. Inputting and editing text is possible by tapping on the TextBox.

The image below shows an empty pop-up TextBox on top of Canvas (in normal mode) and a pop-up TextBox with 5 rows of text

How_to_use_pop-up_TextBox_in_Java_ME



Here is a simple MIDlet demonstrating pop-up TextBox feature.

Source code: PopUpTextBoxMIDlet.java

```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Alert;
import javax.microedition.lcdui.AlertType;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;

public class PopUpTextBoxMIDlet extends MIDlet {
    private PopUpTextBoxCanvas canvas;
    protected String canvasText = "Text from the TextBox";

    public void startApp() {
        canvas = new PopUpTextBoxCanvas(this);
        Display.getDisplay(this).setCurrent(canvas);
    }

    public void pauseApp() {
    }

    public void destroyApp(boolean unconditional) {
    }

    //Finalizes the popup textbokx
}
```

How_to_use_pop-up_TextBox_in_Java_ME

```
protected void closeTextBox(boolean update) {
    //gets the written text
    if (update) canvasText = canvas.textbox.getString();
    //destroy Popup textbox reference
    if (canvas.textbox != null) canvas.textbox = null;
    //sets the canvas screen to visible
    Display.getDisplay(this).setCurrent(canvas);
}

protected void showError(String title, String text) {
    //Creates a new ERROR alert, with a title, a text and no images
    Alert alert = new Alert(title, text, null, AlertType.ERROR);
    alert.setTimeout(Alert.FOREVER); //sets the Alert to modal (the user must dismiss it)
    alert.getType().playSound(Display.getDisplay(this)); //plays sound related to ERROR alert
    Displayable current = Display.getDisplay(this).getCurrent(); //gets current displayable
    if (current instanceof Alert) {} //if displayable is an Alert. Do nothing!
    else Display.getDisplay(this).setCurrent(alert); //otherwise, sets Alert to visible
}
}
```

Source code: PopUpTextBox.java

```
import javax.microedition.lcdui.*;

public class PopUpTextBox extends TextBox implements CommandListener {
    //"Ok" command
    private Command okCommand;
    //"Cancel" command
    private Command cancelCommand;
    //Our MIDlet reference
    private PopUpTextBoxMIDlet midlet;

    public PopUpTextBox(String title, String text, int maxsize, int constraints, PopUpTextBoxMIDlet midlet) {
        super(title, text, maxsize, constraints);
        this.midlet = midlet;
        okCommand = new Command("Ok", Command.OK, 1); //Creates the "Ok" command
        cancelCommand = new Command("Cancel", Command.CANCEL, 1); //Creates the "Cancel" command
        this.addCommand(okCommand); //adds the "Ok" command to the TextBox
        this.addCommand(cancelCommand); //adds the "Cancel" command to the TextBox
        this.setCommandListener(this); //sets "this" class as the command listener
    }

    public void commandAction(Command c, Displayable d) {
        if (c == okCommand) {
            //Closes the textBox. If true
            midlet.closeTextBox(true);
        }
        if (c == cancelCommand) {
            midlet.closeTextBox(false);
        }
    }
}
```

Source code: PopUpTextBoxCanvas.java

```

import javax.microedition.lcdui.Canvas;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.Graphics;
import javax.microedition.lcdui.TextField;

public class PopUpTextBoxCanvas extends Canvas implements CommandListener {
    //MIDlet reference
    private PopUpTextBoxMIDlet midlet;
    //Enter command
    private Command enterCommand;
    //Exit command
    private Command exitCommand;
    //PopUpTextBox reference
    protected PopUpTextBox textbox;
    //The screen width
    private int width;
    //The screen height
    private int height;

    public PopUpTextBoxCanvas(PopUpTextBoxMIDlet midlet) {
        this.midlet = midlet;
        //Creates a new "Enter" Command: confirm the text
        enterCommand = new Command("Enter text", Command.SCREEN, 1);
        //Creates a new "Exit" Command: exit the application
        exitCommand = new Command("Exit", Command.EXIT, 1);
        this.addCommand(enterCommand); //adds the "Enter" command to the screen
        this.addCommand(exitCommand); //adds the "Exit" command to the screen
        this.setCommandListener(this); //sets "this" (PopUpTextBoxCanvas) as the command listener
    }

    public void paint(Graphics g) {
        g.setColor(255, 255,255);
        g.fillRect(0, 0, width, height);
        g.setColor(0, 0, 0);
        g.drawString(midlet.canvasText, 0, 0, Graphics.TOP|Graphics.LEFT);
    }

    protected void keyPressed(int keyCode) { }

    protected void keyReleased(int keyCode) { }

    protected void keyRepeated(int keyCode) { }

    protected void pointerDragged(int x, int y) { }

    protected void pointerPressed(int x, int y) { }

    protected void pointerReleased(int x, int y) { }

    protected void sizeChanged(int w, int h) {
        width = w;
        height = h;
        repaint();
    }

    public void commandAction(Command c, Displayable d) {

```

How_to_use_pop-up_TextBox_in_Java_ME

```
if (c == enterCommand) {
    //Creates a new PopUpTextBox
    textbox = new PopUpTextBox("Enter text", midlet.canvasText, 1000, TextField.ANY, midl
    //sets the popup textbox to visible
    Display.getDisplay(midlet).setCurrent(textbox);
}
if (c == exitCommand) {
    //destroy MIDlet: returns the resources to O.S (Operating System)
    midlet.notifyDestroyed();
}
}
}
```

Example application

- [PopUpTextBoxMIDlet.zip](#) containing PopUpTextBoxMIDlet.jad, PopUpTextBoxMIDlet.jar and the source code

See also

- [Pop-up TextBox](#)
- [JAD and JAR manifest attributes](#)