Carbide.c++ Tutorial 4 - Creating a Settings view

This Symbian C++ code example shows how to create a settings view using the Carbide.c++ UI Designer.

Overview

In tutorial 3 an example demonstration of, how to create a simple list box using View has been give. This tutorial is continuation of the tutorial 3. As in this tutorial we are going to design and create a view using UI designer. The view will show the use of control and editors which can be used in settings view.

The example has a view with one Text Editor control.

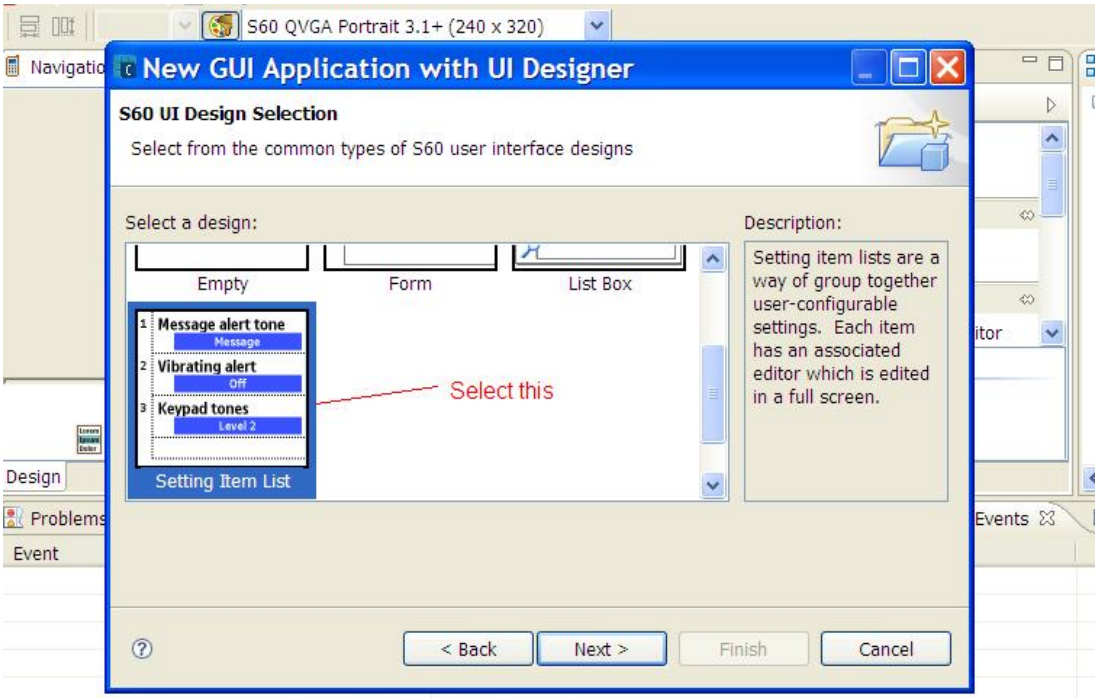
Prerequisites

Follow Tutorial one in order to understand basics of creating and running an application: Carbide.c++ Tutorial 1 - Helloworld

And follow Carbide.c++ Tutorial 3 - Using List view to see how to create a view based application using UI designer.

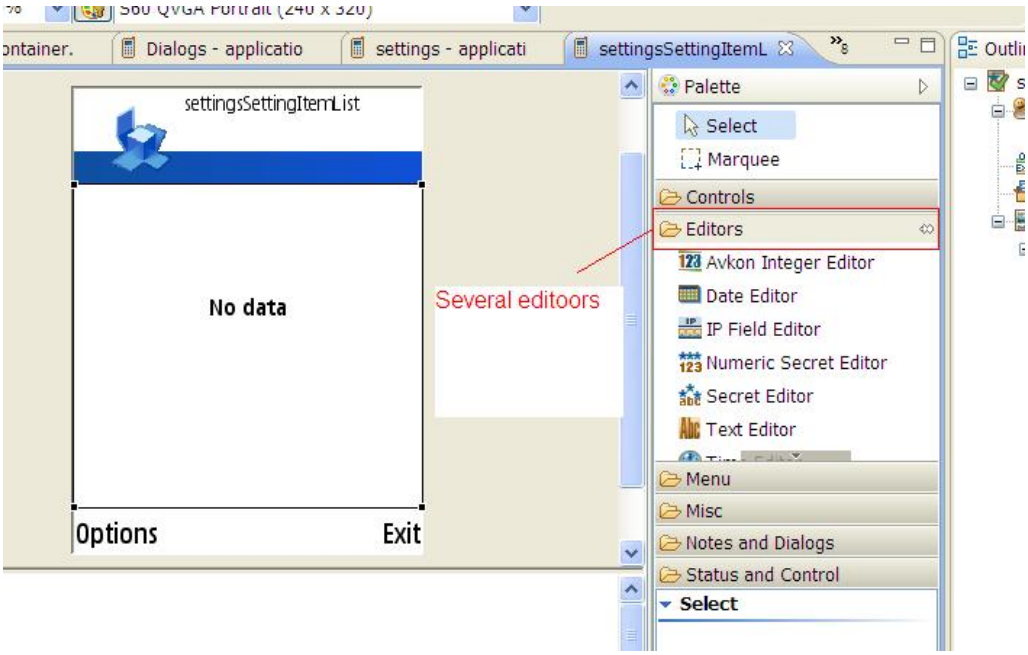
Step 1 - Start a new project

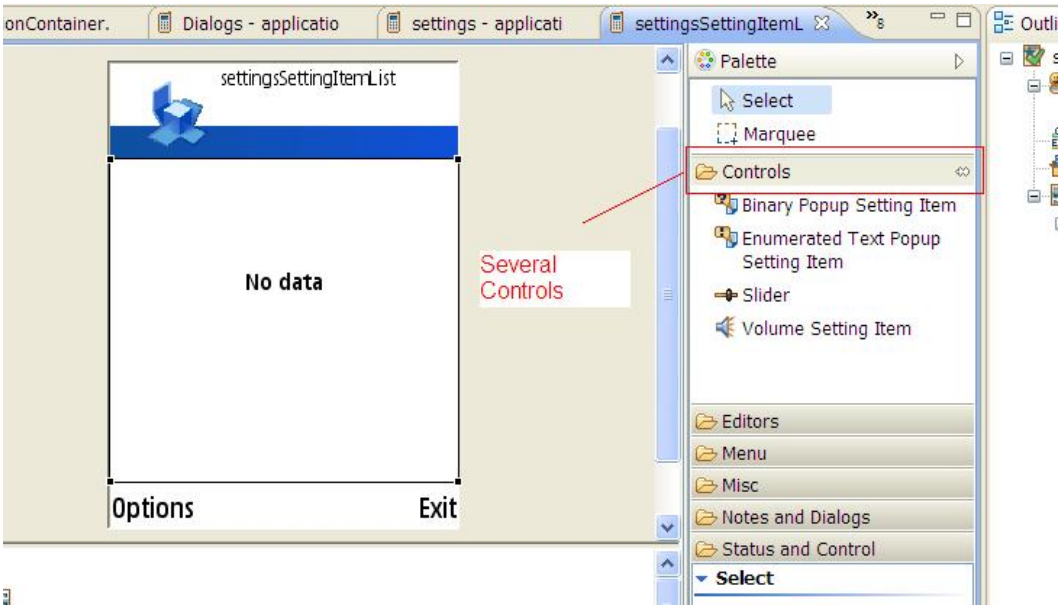
Start with a new project using Carbide.c++ IDE, call it "settings" (or whatever you like). Once you are done with all basic steps you will see this window. In this step you will need to select the settings view type.



After this step select all defaults and end up with the usual UI designer. In UI designer see various controls and Editors which can be added to settings view.

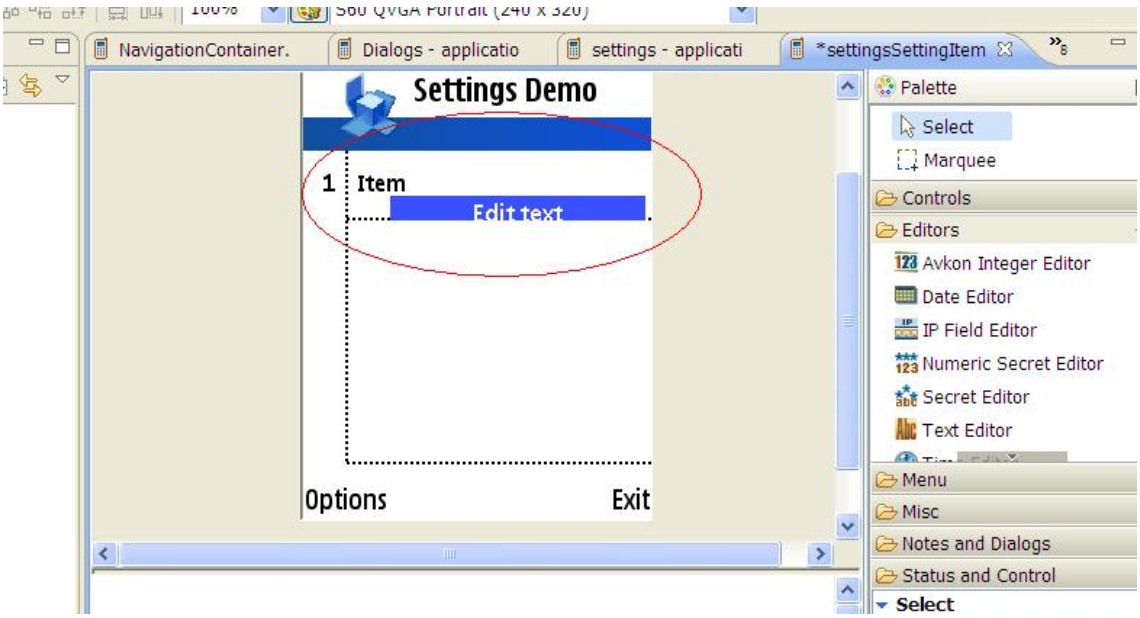
(C) Copyright Nokia 2013. All rights reserved.





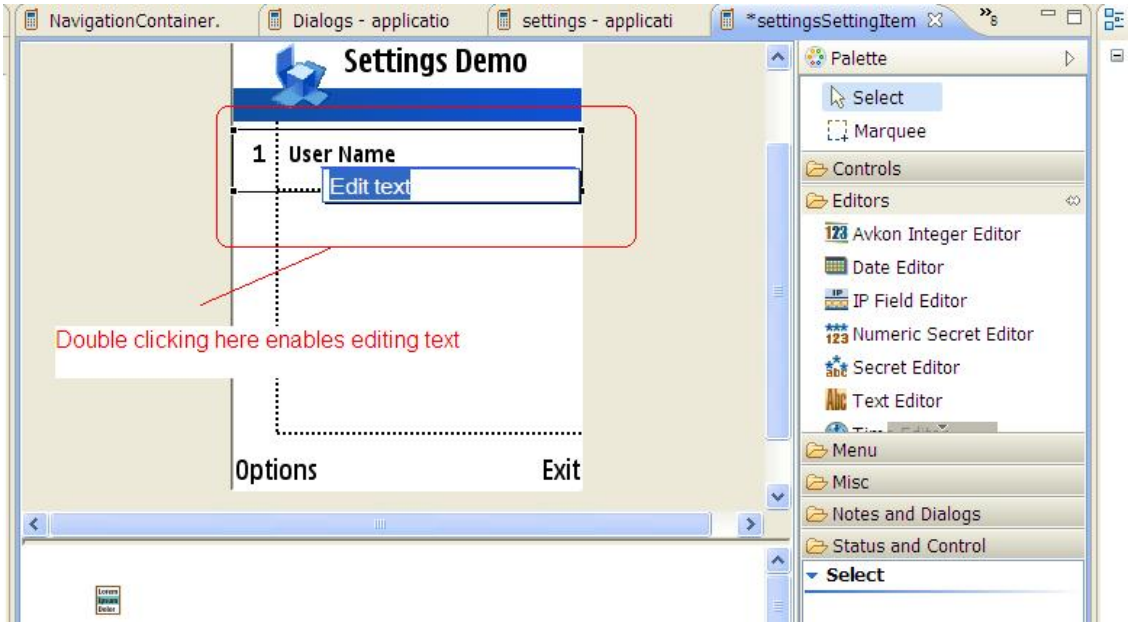
Step 2 - Adding a Text Editor

Now drag a Text Editor from "Editors" to view. The view will look as follows



(C) Copyright Nokia 2013. All rights reserved.

Now remove the default text from Text Editor as follows

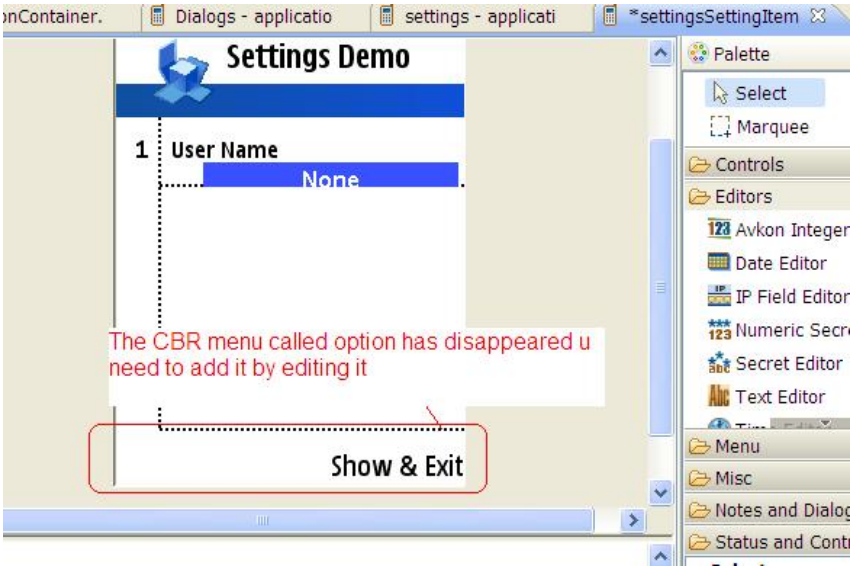


Step 3 - Configuring right softkey

Next step is to change the right softkey, which will trigger an action to show the entered "text" in settings (Text Editor). **This will show how you can read the entered settings. This will also give an idea how the overall code generated in background works.**

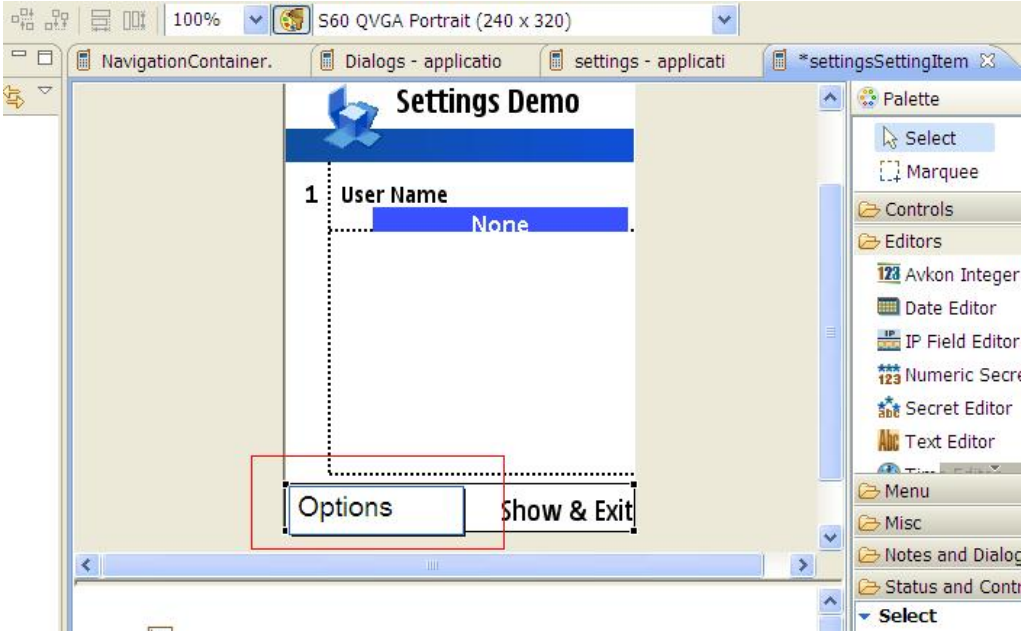
Double click on right softkey and edit it as shown in following image.

Important: When the right menu on CBR (i.e Menu Bar) or softkey is edited, the left softkey title disappears but this is nothing to worry. Simply retype the left softkey title by double clicking on empty space on right side on menu bar.



The menu now has both soft-keys with title.

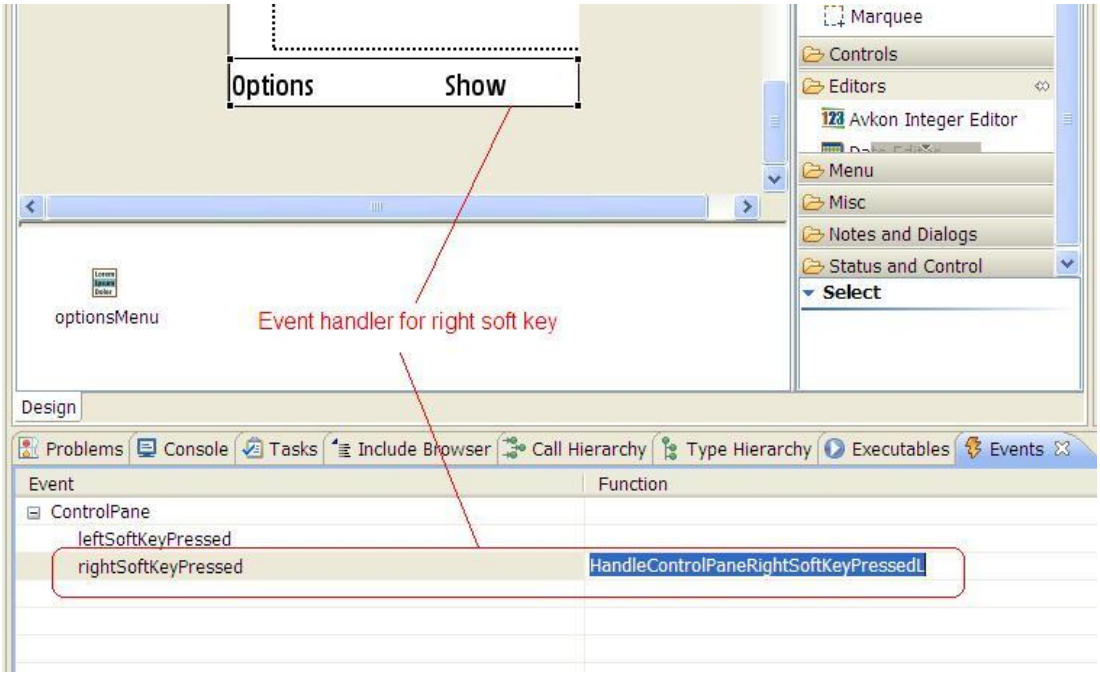
(C) Copyright Nokia 2013. All rights reserved.



Step 4 - Handling the softkey event

Now we have an application with all necessary UI designed with UI designer. Lets add the code to make it working. Add the event handler for the right softkey from menu bar. Generating the event handler code has been explained in *Tutorial 1*.

Select the events section, make sure that the menu bar is selected. Events section will show two events one for right softkey and second for left softkey. Select the right softkey menu event and double click in the empty space in the next column.



Type down the code shown in the following image.

(C) Copyright Nokia 2013. All rights reserved.

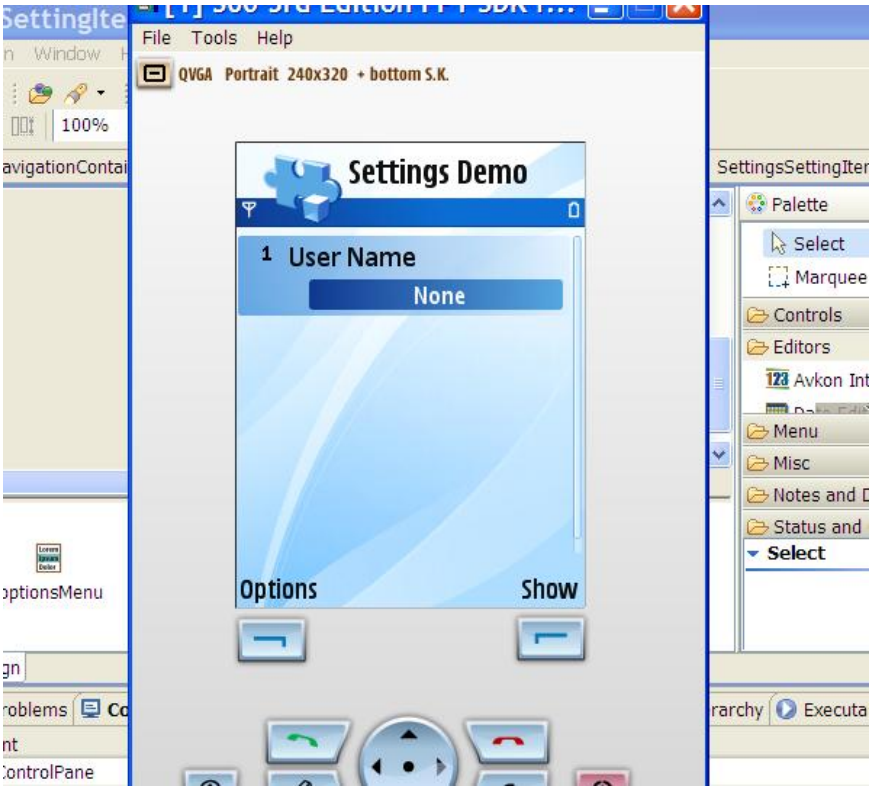
```
 * Handle the rightSoftKeyPressed event.
 * @return ETrue if the command was handled, EFalse if not
 */
TBool CsettingsSettingItemListView::HandleControlPaneRightSoftKeyPressedL( TInt aCom
    {
    // TODO: implement rightSoftKeyPressed event handler
    TBuf<10> text;
    text.Copy(this->iSettings->Edit1());
    iEikonEnv->AlertWin(text);

    return ETrue;
    }
```

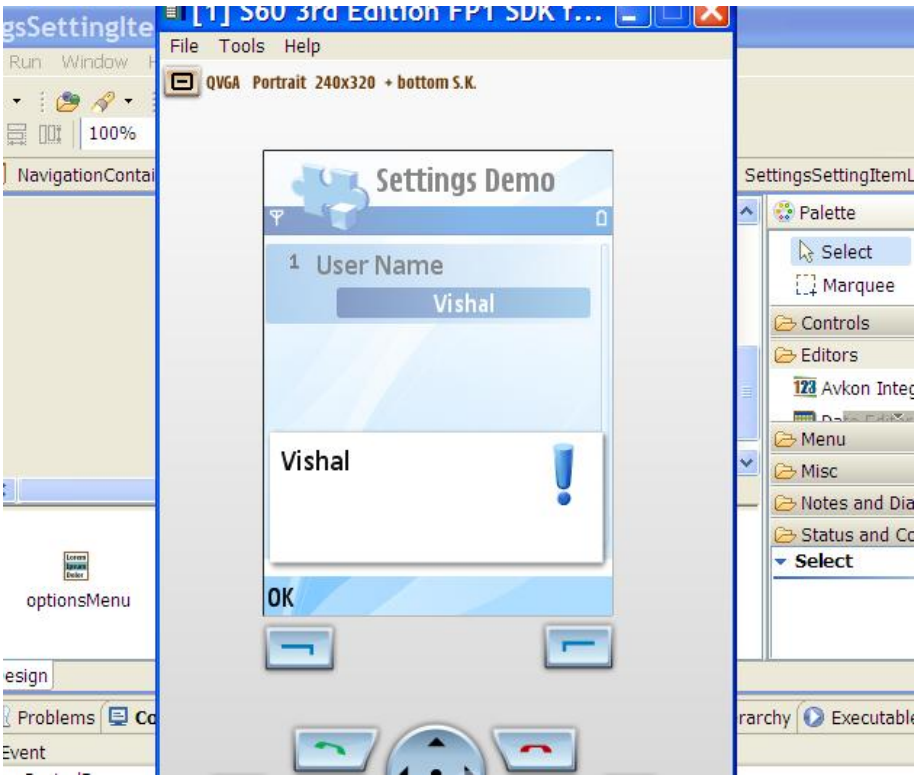
Problems | Console | Tasks | Include Browser | Call Hierarchy | Type Hierarchy | Executables | Events

Step 5 - Running the application

Build the code and run the application in emulator. Following will be the output.



(C) Copyright Nokia 2013. All rights reserved.



Success ! we are done with a basic application which shows how to use settings and related controls/editors.

Conclusion

Several editors and controls are available, which can be used with settings view. This tutorial show how to design the basic settings view with a simple editor.

Source code

File:Settings.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.