# Skin usability

## Description

Skins provide a way of changing the appearance of the UI at run time. They allow you to customize the default background for applications and modify colors and icons.To give your application a consistent look and feel change skin dynamically. Not all controls need to provide the same level of support for skins. S60 UI controls offer different levels of support for skins, depending on their needs.

## Some important usability guideline are

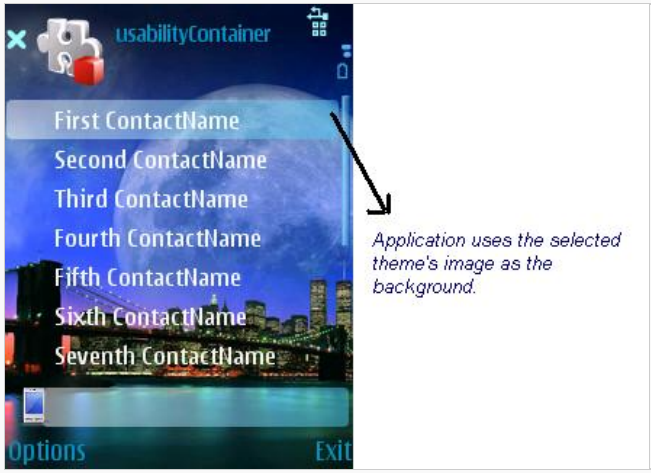
1. For control panes and status pane use compulsory skin-providing controls which always display skins and dynamically comply with skin changes.
2. For lists and grid controls use optionally skin-providing controls which do not display skins, but they can be enabled to do so if required
3. Always draw non-skin aware controls on the top of skinned background.

## How to use

The view background could either be:-

- The background image of the selected theme

Skins can be enabled for the application's main view, which would allow using the background image of the selected theme.



Theme image as background

For skin implementation on S60/Symbian C++ check links:-

- Enabling Skin
- Enable Skin Support in Views/Containers
- Drawing skins in container
- Custom image deployed by the application

From a design stand point, the UI designer can also think of creating custom skins/images to be used as the background of the application view. In this case it is important to make sure that the color combinations are not too loud, and standout well against the default themes on the device.

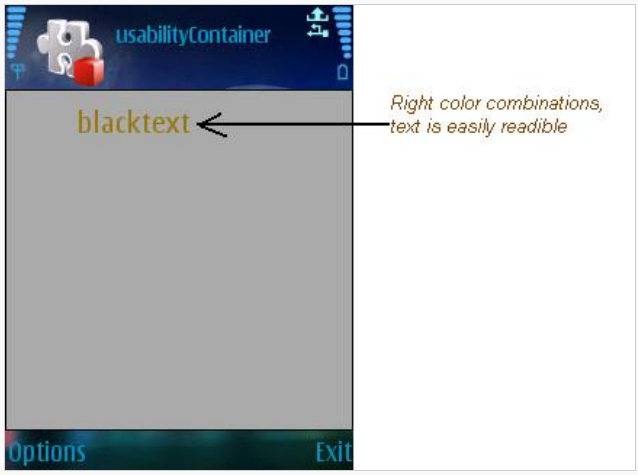

(C) Copyright Nokia 2013. All rights reserved.

Custom image as background

- Color pattern/animation implementation

The designer can also think about implementing the background using colors and animations. In this case it is important to make sure that the color combinations are harmonious and do not make it tough for the user to read the content being displayed.

Wrong color combination for background



Right color combination for background

In case of custom drawing/redrawing of the view background, care should be placed to avoid any flickering as it doesn't give a good experience to the end user. To avoid flickering, double buffering can be used as a possible mechanism, details from CS001049 - Avoid flickering with double buffering & CS001268 - Using double buffering in Java ME


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.