

# How to find out the correct location for softkey labels

## Overview

The Command Button Array (CBA) is usually located at the bottom of the screen in S60 devices. If a device supports switching between portrait and landscape orientations, the location of the CBA is not fixed (in relation to the screen). While the control pane in the S60 UI is always positioned correctly, scalable applications that draw their own custom labels for softkeys should be aware of the correct location for the CBA.

## Detailed description

In some phones (such as the Nokia E70, Nokia E90, Nokia N93, Nokia N95), the softkeys will be on the right or left side of the screen when in landscape mode. If an application needs to adjust its UI (softkey positions) based on orientation, the CBA location can be retrieved with `AknLayoutUtils::CbaLocation()`

## Solution

The return value of `AknLayoutUtils::CbaLocation()` is one of the following:

```
enum TAknCbaLocation
{
    EAknCbaLocationBottom, // landscape and portrait
    EAknCbaLocationRight,  // only landscape
    EAknCbaLocationLeft   // only landscape
};
```

When the CBA is located at the bottom, the positive (**OK/Options**) softkey is always on the left-hand side, and the negative (**Cancel/Back**) on the right-hand side. When the CBA is on either side of the screen, most devices have the positive softkey at the bottom. As an exception, the Nokia E90 Communicator (when open) has the positive softkey on top. There is no API for directly resolving the order of softkeys. However, the position of the softkey controls can be checked as follows:

```
TBool IsOptionsButtonOnTop()
{
    CEikButtonGroupContainer* cba = CEikButtonGroupContainer::Current();
    if( !cba )
    {
        return EFalse;
    }

    // Assumes that CBA has Options and Exit commands
    CCoeControl* options = cba->ControlOrNull( EAknSoftkeyOptions );
    CCoeControl* exit = cba->ControlOrNull( EAknSoftkeyExit );

    if( options && exit )
    {
        if( options->Position().iY < exit->Position().iY )
        {
            return ETrue;
        }
    }
    return EFalse;
}
```

