

UI_Designer_Listbox_Scrollbar

I recommend you use the Carbide.c++ UI Designer to implement S60 applications as it really is a powerful tool that will improve your time to market. However you will need to get use to it to understand its limits.

One thing I was struggling with for instance was to get the scrollbar working properly on my list box views. In fact I could not find a way to enable the scrollbar through the UI designer itself so I had to add the code for creating the scroll bar myself.

Creating your scrollbar...

Now the code for creating the scrollbar is quite straight forward. It should look like that:

```
if ( iListBox->ScrollBarFrame()==NULL) //defensive
{
    iListBox->CreateScrollBarFrameL( ETrue );
    iListBox->ScrollBarFrame()->SetScrollBarVisibilityL( CEikScrollBarFrame::EAuto, CEikScrollBarFrame::EAuto );
}
```

...at the right time

The tricky part is to put it in the right place to avoid getting some flashing and blinking in place of the scrollbar frame upon view activation. That's a side effect I observed on E90 for instance. Turns out you have to create the scrollbar before you set the list box rectangle. So your code should look like that:

```
void CMyAppListBox::ConstructL(
    const TRect& aRect,
    const CCoeControl* aParent,
    MEikCommandObserver )
{
    if ( aParent == NULL )
    {
        CreateWindowL(
            aRect,
            aParent );
    }
    else
    {
        SetContainerWindow( aParent );
    }

    if ( !iFocusOnNULL )
        iCommandObserver = aCommandObserver;
    InitializeControlsL(
        aRect );

    //Make sure we create the scroll bar before SetRect
    if ( iListBox->ScrollBarFrame()==NULL) //defensive
    {
        iListBox->CreateScrollBarFrameL( ETrue );
        iListBox->ScrollBarFrame()->SetScrollBarVisibilityL( CEikScrollBarFrame::EAuto, CEikScrollBarFrame::EAuto );
    }

    SetRect( aRect );
    ActivateL(
        aRect );
    // [[[ begin generated region: do not modify [Post-ActivateL initializations]
    // ]]] end generated region [Post-ActivateL initializations]
}
```