



The **CListBoxContainer** illustrates how to use list boxes in your application. The main function where the list box is made is named *MakeListBoxL()*. Inside it the list box is constructed as **CAknSingleLargeStyleListBox**. If you want to change it to something else, you can one of the following S60 list box types:

- CAknSingleStyleListBox
- CAknSingleNumberStyleListBox
- CAknSingleHeadingStyleListBox,
- CAknSingleGraphicStyleListBox
- CAknSingleGraphicHeadingStyleListBox
- CAknSingleNumberHeadingStyleListBox
- CAknSingleLargeStyleListBox
- CAknDoubleStyleListBox
- CAknDoubleNumberStyleListBox
- CAknDoubleLargeStyleListBox
- CAknDoubleGraphicStyleListBox
- CAknDoubleTimeStyleListBox

In *GetArrayL()* you need to construct the text strings for the list box according to the selected list box type. When using S60 specific list boxes, the icon size will be automatically set by the list box, so you don't need to set any size for the icons. Remember to add CListBoxContainer instance to control stack (with *AddToStackL()*) after creation to get notifications of key events in *OfferKeyEventL()*.

CListBoxContainer.cpp

```
#include "CListBoxContainer.h"
#include <akniconarray.h>
#include <eikclbd.h>

CListBoxContainer::CListBoxContainer(void)
{
}

CListBoxContainer::~CListBoxContainer()
{
    delete iMyListBox;
}

void CListBoxContainer::ConstructL()
{
    CreateWindowL
    (SEikEnv->EikAppUi()->ClientRect());

    ActivateL
    (DrawNow);
}

void CListBoxContainer::SizeChanged()
{
    TRAP_IGNORE(EstBoxL());
}

```

List_box_example

```
void CListBoxContainer::HandleResourceChange(TInt aType)
{
    TRect rect

if ( aType==KEikDynamicLayoutVariantSwitch )
{
    :ApplyMethodWithRect(AknLayoutUtils::EMainPane, rect);
    (rect); SetRect
}

    CCoEHandleResourceChange(aType);
}

void CListBoxContainer::MakeListBoxL()
{
    TInt MySetIndex

if(iMyListBox)
{
    = iMyListBox->GetCurrentItemIndex();
}

delete iMyListBox;
iMyListBox;

iMyListBox(ELeave ) CAknSingleLargeStyleListBox();
iMyListBox->ConstructL(this,EAknListBoxSelectionList);

CAknIconArray* icons =new( ELeave ) CAknIconArray(1);
CleanupStackPush(icons);

iMyListBox->SetItemTextArray(GetArrayL(icons));
iMyListBox->Model()->SetOwnershipType(ELbmOwnsItemArray);
//
// Inform the listbox of item additions.
// This should be done if you want your items displayed correctly
iMyListBox->HandleItemAdditionL();

CleanupStackPop(icons);
iMyListBox->Drawer()->ColumnData()->SetIconArray(icons);

iMyListBox->ScrollBarFrameL( ETrue );
iMyListBox->ScrollBarFrame()->SetScrollBarVisibilityL(
    CEikScrollBarFrame::EAuto);

iMyListBox->Rect();

iMyListBox->SetListEmptyTextL(KtxNoData);

iMyListBox->UpdateL();

TInt ItemsCount=iMyListBox->Model()->ItemTextArray()->MdcaCount();

if(ItemsCount > MySetIndex && MySetIndex >= 0)
    ->SetCurrentItemIndex(MySetIndex);
else if(ItemsCount > 0)
    ->SetCurrentItemIndex(0);

UpdateScrollBar();
iMyListBox->Draw();
}
```

List_box_example

```
CDesCArray* CListBoxContainer::GetArrayL(CArrayPtr<CGulIcon>* aIcon)
{
    CDesCArrayFlat myArray = new (ELeave)CDesCArrayFlat(1);
    CleanupStack(myArray);

    // 1) Append Text string to myArray in here as "1\tMytext",
    // where 1 is the zero based icon index
    //
    // Second and third icon indexes can also be specified in the form:
    // "1\tMytext\t2\t3" where 2 and 3 are also zero-based icon indexes
    //
    // 2) Remember to load and add icons in here. Listbox will
    // automatically resize iconsthus no need to set any size for them.
    // Tip: you may find AknIconUtils::CreateIconLC and
    // AknIconUtils::AvkonIconFileName() useful. In that case,
    // include akniconutils.h and avkon.mbg

    CleanupStack(myArray);
    return myArray;
}

void CListBoxContainer::UpdateScrollBar(CEikListBox* aListBox)
{
    if (aListBox)
    {
        TInt pos(aListBox->View()->CurrentItemIndex());
        if (aListBox->ScrollBarFrame())
        {
            aListBox->ScrollBarFrame()->MoveVertThumbTo(pos);
        }
    }
}

void CListBoxContainer::Draw(const TRect& /*aRect*/) const
{
    CWindowGSystemGc();
    ClearRect();
}

TKeyResponse CListBoxContainer::OfferKeyEventL(const TKeyEvent& aKeyEvent, TEventCode aEventCode)
{
    TKeyResponse Ret(EKeyWasNotConsumed);

    if (iMyListBox)
    {
        Ret = iMyListBox->OfferKeyEventL(aKeyEvent, aEventCode);
    }

    return Ret;
}

CCoeControl* CListBoxContainer::ComponentControl( TInt /*aIndex*/) const
{
    return iMyListBox;
}

TInt CListBoxContainer::CountComponentControls() const
{
    if (iMyListBox)
    {
        return 1;
    }
    else

```

```
return 0;
}
```

CListBoxContainer.h

```
#include <aknlists.h>
#include <gulicon.h>

_LIT(KtxNoData, "Empty list");

class CListBoxContainer : public CCoeControl
{
public:
    CListBoxContainer(void);
    void ConstructL(void);
    ~CListBoxContainer();
public:
    TKeyResponse OfferKeyEventL(KeyEvent& aKeyEvent, TEventCode aType);
    CCoeComponentControl( TInt aIndex) const;
    TInt CountComponentControls();
private:
    virtual void SizeChanged();
    virtual void HandleResourceChange(TInt aType);
    void Draw(const TRect& aRect) const;
        CDcsArrayL(CArrayPtr<CGulIcon>* aIcon);
    void MakeListBoxL(void);
    void UpdateScrollBar(CEikListBox* aListBox);
private:
    CAknSingleLargeStyleListBox;
};
```

Related Links

- [How to create a simple listbox](#)
- [Querying selection with list](#)
- [TSS000647 - Applying actions on list box items](#)
- [How to add Marquee effect in ListBox](#)
- [How to add ScrollBar in ListBox](#)
- [How to add additional data to the list box?](#)
- [Listbox creation using resource](#)