



The **CSearchListContainer** illustrates how to use **CAknSearchField** with list boxes to allow users to search the list items. To use this example modify the *GetArrayL()* method and add code to populate the text item array for the list box as well as add icons to the list box icon array. The text string format for the **CAknSingleGraphicStyleListBox** is `?1\Text ?`, where 1 is a zero based index for the icon array.

If you want to use a list box without the search box option you can use the [List box example](#) instead.

SearchListbox.cpp

```
CSearchListContainer* CSearchListContainer::NewL(void)
{
    CSearchListContainer* self = CSearchListContainer::NewLC();
    CleanupStack::Pop(self);
    return self;
}

CSearchListContainer* CSearchListContainer::NewLC(void)
{
    CSearchListContainer* self = new (ELeave) CSearchListContainer();
    CleanupStack::PushL(self);
    self->ConstructL();
    return self;
}

CSearchListContainer::~CSearchListContainer()
{
    delete iFindBox;
        iFindBox

    delete iListBox;
}

void CSearchListContainer::ConstructL()
{
    CreateWindowL();

    SetRect(CEikonEnv::Static()->EikAppUi()->ClientRect());

        MakeListBoxL        ();

        ActivateL        ();
    DrawNow();
}

void CSearchListContainer::MakeListBoxL()
{
    TInt MySetIndex

    if(iListBox)
    {
        = GetMySetFindIndexL();
    }

    delete iListBox;
        iFindBox

```

List_box_with_Search,_example

```

iListBox = new( ELeave ) CAknSingleGraphicStyleListBox();
iListBox->ConstructL(this, EAknListBoxSelectionList);

CArrayPtr<CGulIcon>* icons = new( ELeave ) CAknIconArray(10);
CleanupStack::PushL(icons);

iListBox->SetItemTextArray(GetArrayL(icons));
iListBox->Model()->SetOwnershipType(ELbmOwnsItemArray);

CleanupStack::Pop(icons);
iListBox->ColumnData()->SetIconArray(icons);
iListBox->ScrollBarFrameL( ETrue );
iListBox->ScrollBarFrame()->SetScrollBarVisibilityL(
CEikScrollBarFrame::EOff, CEikScrollBarFrame::EAuto );

iListBox->SetRect( Rect() );

delete iFindBox;
iFindBox = new( ELeave ) CAknSearchField(
CAknSearchField::ESearch);
CleanupStack::PushL(iFindBox);

iFindBox->SetRect( Rect() );

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

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

UpdateScrollBar();
DrawNow();
}

CAknSearchField* CSearchListContainer::CreateFindBoxL(CEikListBox* aListBox,
CTextListBoxModel* aModel, CAknSearchField::TSearchFieldStyle aStyle)
{
    CAknSearchField* findbox = NULL;

    if (aListBox && aModel)
    {
        // Gets pointer of CAknFilteredTextListBoxModel.
        CAknFilteredTextListBoxModel* model =
        STATIC_CAST( CAknFilteredTextListBoxModel*, aModel );
        // Creates FindBox.
        findbox = CAknSearchField::NewL( *this, aStyle, NULL,
KAKnExListFindBoxTextLength);
        CleanupStack::PushL(findbox);
        // Creates CAknListBoxFilterItems class.
        model->CreateFilterL( aListBox, findbox );
        //Filter can get by model->Filter();
        CleanupStack::Pop(findbox); // findbox
    }

    return findbox;
}

CDesCArray* CSearchListContainer::GetArrayL(CArrayPtr<CGulIcon>* aIcons)
{
    CDesCArray* MyArray = new(ELeave)CDesCArrayFlat(10);
}

```

List_box_with_Search,_example

```

        CleanupSpace(MyArray);

// Append text items and icons in here...
//MyArray->AppendL(TextBuf);
//aIcons->AppendL(CGulIcon::NewL(Icon, IconMsk));

        CleanupSpace(MyArray);
return MyArray;
}

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

void CSearchListContainer::SizeChanged()
{
    if (iListBox)
    {
        if (iFindBox)
        {
            CAknColumnListBox* aknListBox = STATIC_CAST(CAknColumnListBox*,
iListBox);
            AknFindHandleFixedFindSizeChanged(this, aknListBox, iFindBox);
        }
        else
        {
            iListBox->SetRect(Rect()); // Sets rectangle of lstbox.
        }
    }
}

void CSearchListContainer::HandleResourceChange(TInt aType)
{
    TRect rect

    if ( aType==KEikDynamicLayoutVariantSwitch )
    {
        AknLayoutUtils::LayoutMetricsRect(AknLayoutUtils::EMainPane, rect);

        SetRect(rect);
    }

    CCoeClient::HandleResourceChange(aType);
}

TInt CSearchListContainer::GetSelectedIndexL(void)
{
    TInt i;

    if(iListBox)
    {
        TInt iCurBoxItemIndex();

```

List_box_with_Search,_example

```
CAknFilteredTextListBoxModel
STATIC_CAST(CAknFilteredTextListBoxModel*, iListBox->Model());

if(model && CurrItemInd >= 0)
{
    = model->Filter()->FilteredItemIndex(CurrItemInd);
}
}

return Ret;
}

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

switch (aKeyEvent.iCode)
    {
        case EKeyDevice3:
break;
default:
if(iListBox)
    {
if ( iFindBox )
        {
            TBool needRefresh( EFalse );

            // Offers the key event to find box.
            if ( AknFind::HandleFindOfferKeyEventL( aKeyEvent, aEventCode, this,
                                                    iListBox, iFindBox,
                                                    EFalse,
                                                    needRefresh ) ==
EKeyWasConsumed )
                {
                    if ( needRefresh )
                        {
                            SizeChanged ();
                            DrawNow ();
                        }

                    return EKeyWasConsumed;
                }
            }

        = iListBox->OfferKeyEventL(aKeyEvent, aEventCode);
    }
break;
}

return Ret;
}

void CSearchListContainer::Draw(const TRect& /*aRect*/) const
{
    CWindowGSystemGc();
    gc.    Clear(Rect());
}
}
```

List_box_with_Search,_example

```
CCoeControl* CSearchListContainer::ComponentControl( TInt aIndex) const
{
    if(iFindBox && aIndex)
        return iFindBox;
    else
        return iListBox;
}

TInt CSearchListContainer::CountComponentControls() const
{
    if(iListBox && iFindBox)
    {
        return 2;
    }
    else
    {
        return 0;
    }
}
```

SearchListbox.h

```
#include <coecntrl.h>
#include <aknlists.h>
#include <EIKLBX.H>
#include <aknsfld.h>

const TInt KAKnExListFindBoxTextLength = 20;

class CSearchListContainer : public CCoeControl
{
public:
    static CSearchListContainer* NewL(void);
    static CSearchListContainer* NewLC(void);
    ~CSearchListContainer();

public:
    TInt CountComponentControls();
    CCoeControl* ComponentControl( TInt aIndex) const;
    TKeyResponse OfferKeyEventL( TKeyEvent& aKeyEvent,
    TEventCode aEventCode);
    TInt GetSelectedIndexL();

private:
    void MakeListBoxL();
    CAknSearchField* CreateFindBoxL(CEikListBox* aListBox,
    CTextListBoxModel* aModel, CAknSearchField::TSearchFieldStyle aStyle);
    CDesCArrayL( CArrayPtr<CGulIcon>* aIcons);
    void UpdateScrollBar(CEikListBox* aListBox);
    virtual void SizeChanged();
    virtual void HandleResourceChange( TInt aType);
    void Draw(const TRect& aRect) const;
    void ConstructL();

private:
    CAknSingleGraphicStyle* iListBox ; iListBox
    CAknSearchField ; iFindBox
};
```

Links

[How to create a simple listbox](#)

[List box example](#)

[Querying selection with list](#)

--