Working with Generic Button API - S60 Touch UI

Overview

Generic Button API is a new component introduced in S60 5th Edition Touch UI concept. In this article we will learn how to play with Generic Button API with different methods.

What is Generic Button API

- It is used for enabling touch support for buttons.
- It can be used in other UI container components. For e.g: Toolbar
- It is a floating component.
- It can have different states and action can be performed accordingly.
- To observe the buttons in your application, use the `MCoeControlObserver` observer interface.

Creating Generic Button API resource

```
...
#include <avkon.mbg>
//Button resource
RESOURCE AVKON_BUTTON r_first_button
    {
    flags  = KAknButtonTextInsideFrame;
    states =
        {
        AVKON_BUTTON_STATE
            {
            helptxt = "First";
            bmpfile = AVKON_BITMAP_FILE;
            bmpid =  EMbmAvkonQgn_note_query;
            bmpmask =  EMbmAvkonQgn_note_query_mask;
            },
```


(C) Copyright Nokia 2014. All rights reserved.

```
        AVKON_BUTTON_STATE
            {
            flags = KAknButtonStateHasLatchedFrame;
            bmpfile = AVKON_BITMAP_FILE;
            bmpid =  EMbmAvkonQgn_note_query;
            bmpmask =  EMbmAvkonQgn_note_query_mask;
            }
        };
    }
RESOURCE AVKON_BUTTON r_second_button
    {
    flags  = KAknButtonTextInsideFrame;
    states =
        {
        AVKON_BUTTON_STATE
            {
            txt = "NoIcon";
            helptxt = "Second";
            },
        AVKON_BUTTON_STATE
            {
            flags = KAknButtonStateHasLatchedFrame;
            txt = "Pressed!";
            helptxt = "Second";
            }
        };
    }
```

Implementing Generic Button API in the code

- We will inherit our class from `MCoeControlObserver` interface and implement its pure virtual method `HandleControlEventL()` to receive and handle events for Buttons.
- Declare objects of `CAknButton` and use it to set observer for this class and handle events on Buttons.

UsingButtonsAPIAppView.h

```
...
#include <aknbutton.h>
class CUsingButtonsAPIAppView : public CCoeControl, public MCoeControlObserver
{
    ....
    ....
        //From MCoeControlObserver
        virtual void HandleControlEventL(CCoeControl* aControl,TCoeEvent aEventType);

        TInt CountComponentControls() const;
        CCoeControl* ComponentControl(TInt aIndex) const;
        void CreateButtonUsingResourceL();
        void CreateButtonRuntimeL();

    private:
        CAknButton* iFirstButton;
        CAknButton* iSecondButton;
};
```


(C) Copyright Nokia 2014. All rights reserved.

UsingButtonsAPIAppView.cpp

- Call to create Buttons from `ConstructL()` as shown below.

```
// ----------------------------------------------------------------------------
// CUsingButtonsAPIAppView::ConstructL()
// Symbian 2nd phase constructor can leave.
// ----------------------------------------------------------------------------
//
void CUsingButtonsAPIAppView::ConstructL(const TRect& aRect)
{
    // Create a window for this application view
    CreateWindowL();

    // Set the windows size
    SetRect(aRect);

    CreateButtonUsingResourceL();
    // CreateButtonRuntimeL();

    // Activate the window, which makes it ready to be drawn
    ActivateL();
}
```

- Create generic buttons and set the rectangle for them.
- Set "this" as an observer for buttons and eventually we will receive callbacks in `HandleControlEventL()`

```
void CUsingButtonsAPIAppView::CreateButtonUsingResourceL()
{
    iFirstButton = CAknButton::NewL();
    iFirstButton->ConstructFromResourceL(R_FIRST_BUTTON);
    iFirstButton->SetContainerWindowL(*this);
    iFirstButton->SetRect(TRect ( TPoint(20,180), TSize(150,100) ) );
    iFirstButton->SetObserver(this);
    iFirstButton->MakeVisible(ETrue);
    iFirstButton->ActivateL();

    iSecondButton = CAknButton::NewL();
    iSecondButton->ConstructFromResourceL(R_SECOND_BUTTON);
    iSecondButton->SetContainerWindowL(*this);
    iSecondButton->SetRect(TRect ( TPoint(160,180), TSize(150,100) ) );
    iSecondButton->SetObserver(this);
    iSecondButton->MakeVisible(ETrue);
    iSecondButton->ActivateL();
}
```

- We will handle callback events on buttons as shown below in the `HandleControlEventL`

```
void CUsingButtonsAPIAppView::HandleControlEventL( CCoeControl* aControl,
                                                    TCoeEvent aEventType )
{
    switch ( aEventType )
    {
        case EEventStateChanged:
```


(C) Copyright Nokia 2014. All rights reserved.

```
        {
            if(aControl == iFirstButton)
            {
                if(iSecondButton->State()->Flags()==KAknButtonStateHasLatchedFrame)
                    iSecondButton->SetCurrentState(KAknButtonTextInsideFrame,ETrue);
            }
            else if(aControl ==iSecondButton)
            {
                if(iFirstButton->State()->Flags()==KAknButtonStateHasLatchedFrame)
                    iFirstButton->SetCurrentState(KAknButtonTextInsideFrame,ETrue);
            }
        }   break;

        default:
             break;
    }
}
```

- Definition for the following functions to return button controls.

```
TInt CUsingButtonsAPIAppView::CountComponentControls() const
{
    return 2; // return number of controls inside this container
}

CCoeControl* CUsingButtonsAPIAppView::ComponentControl(TInt aIndex) const
{
    switch ( aIndex )
    {
        case 0:
            return iFirstButton;
        case 1:
            return iSecondButton;
        default:
            return NULL;
    }
}
```

- Make sure to delete objects on buttons in the destructor of the class.

```
// ----------------------------------------------------------------------------
// CUsingButtonsAPIAppView::~CUsingButtonsAPIAppView()
// Destructor.
// ----------------------------------------------------------------------------
//
CUsingButtonsAPIAppView::~CUsingButtonsAPIAppView()
{
    delete iFirstButton;
    delete iSecondButton;
    ....
    ....
}
```

Creating Generic Buttons on runtime without using resource

(C) Copyright Nokia 2014. All rights reserved.

```
void CUsingButtonsAPIAppView::CreateButtonRuntimeL()
{
    _LIT(KFirstButtonText,"First");
    _LIT(KFirstButtonHelpText,"Help First Button");
    iFirstButton =
CAknButton::NewL(NULL,NULL,NULL,NULL,KFirstButtonText,KFirstButtonHelpText,0,0);
    iFirstButton->SetContainerWindowL(*this);
    iFirstButton->SetRect(TRect ( TPoint(20,180), TSize(150,100) ) );
    iFirstButton->SetObserver(this);
    iFirstButton->MakeVisible(ETrue);
    iFirstButton->ActivateL();

    _LIT(KSecondButtonText,"Second");
    _LIT(KSecondButtonHelpText,"Help Second Button");
    iSecondButton =
CAknButton::NewL(NULL,NULL,NULL,NULL,KSecondButtonText,KSecondButtonHelpText,0,0);
    iSecondButton->SetContainerWindowL(*this);
    iSecondButton->SetRect(TRect ( TPoint(160,180), TSize(150,100) ) );
    iSecondButton->SetObserver(this);
    iSecondButton->MakeVisible(ETrue);
    iSecondButton->ActivateL();
}
```

Useful functions

CAknButton

- AddStateL()
- SetCurrentState()
- StateIndex()

CAknButtonState

- SetDimmed()
- CreateButtonIconL()
- SetTextL()
- SetHelpTextL()
- SetFlags()
- SetIcon()
- Text()
- HelpText()
- Flags()

MCoeControlObserver

- HandleControlEventL()

Keywords

Headers

- #include <CAknButton.h>

Classes

- CAknButton
- CAknButtonState
- MCoeControlObserver

(C) Copyright Nokia 2014. All rights reserved.

Libraries

- eikcoctl.lib

Example Application

- Generic Button.zip

Related links

- A tour to the S60 Touch UI components
- Working with Toolbar API - S60 Touch UI
- Working with LongTapDetector API - S60 Touch UI
- Working with Stylus Pop-up Menu API - S60 Touch UI
- Working with Tactile Feedback Client API - S60 Touch UI
- Working with Adaptive Search feature - S60 Touch UI
- Working with ChoiceList API - S60 Touch UI
- Working with SingleStyleTreeList with Hierarchical Lists API - S60 Touch UI
- Working with SingleColumnStyleTreeList with Hierarchical Lists API - S60 Touch UI

Reference list

- S60 5th edition SDK help
- S60 5th Edition C++ Developer's Library v1.0

(C) Copyright Nokia 2014. All rights reserved.