Switching views on Symbian

Overview

This snippet demonstrates how to switch views. The snippet uses the Avkon View-Switching Architecture class `CAknView`.

When and why do we need views?

- When your application has multiple screens that form complex navigational paths.
- When you want to save data on every view switching, to update the model with the newly entered or updated data.
- When you want to send data from one screen to another or to external applications.
- `CAknView` handles menu-commands, switching of views, sending keyboard/pointer events to the respective container class.
- Simple applications don't need views that are derived from `CAknView`.

The S60 view architecture is used in application development in the S60 platform. Each view has its own control stack. The view architecture uses the base class `CAknViewAppUi` for the UI controller. The UI controller manages one or more `CAknView` derived views. The view class uses `CCoeControl`-derived container control.

MMP file

The following libraries and capabilities are required:

```
CAPABILITY None

LIBRARY avkon.lib
LIBRARY eikcore.lib
LIBTATY eiksrv.lib
```

Header file

```
#include <aknview.h>
#include <aknviewappui.h>
```

Source file

Application app UI class `CAknViewAppUi` creates the default view on its construction.

```
void CSomeAppUi::ContructL()
    {
    BaseConstructL(EAknEnableSkin);
    // Create your view derived from CAknView
    iMyView = CMyView::NewL();
    // Add view into control stack
    AddViewL(iMyView);
    // Use this as a default view
    SetDefaultViewL(*iMyView);
    }
```

`CAknView` activates the control of the view. `DoActivateL()` is called by the View Server when a client requests that your view is activated.

```
void CSomeView::DoActivateL(const TVwsViewId& aPrevViewId,
      TUid aCustomMessageId,
      const TDesC8& aCustomMessage)
    {
    if (iSomeControl == NULL)
        {
        // Create control for the view
        iSomeControl = CSomeContainer::NewL(ClientRect());
        // Add it to control stack
        AppUi()->AddToStackL(*this, iSomeControl);
        }
    }

void CSomeView::DoDeactivate()
    {
    if (iSomeControl != NULL)
        {
        // Remove view from the control stack
        AppUi()->RemoveFromViewStack(*this, iSomeControl);
        // Delete view
        delete iSomeControl;
        iSomeControl = NULL;
        }
    }
```


(C) Copyright Nokia 2013. All rights reserved.

To switch to a view within your application, you can use the `AppUi` class method `ActivateLocalViewL()`. The new view is activated first, and the previous view is deactivated after that.

```
const TUid KSomeViewId = { 1 }; // UID of the some view
ActivateLocalViewL(KSomeViewId); // activate some view
```

Postconditions

Views can be activated and switched.

See also

- Getting a pointer to CAknViewAppUi or CAknView
- TSS000042 - How can I get a pointer from a container to a view?
- Archived:Sending package buffer between views in Symbian C++
- Archived:Sending custom message between views using Symbian C++
- Getting Eikon environment pointer

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.