Using a Symbian custom control in a dialog

Overview

This example extends the Custom Control example How to create a custom control in Symbian C++ to add the `CMyControl` custom control into `CAknDialog`.

MMP file

Add the following extra library because `CAknDialog` is used.

```
LIBRARY eikdlg.lib
```

Creating CMyDialog

Define a custom dialog that implements the `CEikDialog::CreateCustomControlL()` method.

The method creates a control line in the dialog. The line can thereafter be accessed through the identifier control ID. The control type is created by the Eikon control factory. If the value of control type is not known to the Eikon control factory, the construction of the control must be handled by `CreateCustomControlL()`.

`CMyDialog` dialog resource in `multiviews.rss`

```
RESOURCE DIALOG r_dialog
    {
    flags = EAknDialogGenericFullScreen;
    buttons = R_AVKON_SOFTKEYS_OK_BACK;
    items =
        {
        DLG_LINE
            {
            // CMyControl custom control type (defined in multiviews.hrh)
            type = KMyCustomCtl;
            // CMyControl custom control id (defined in multiviews.hrh)
            id = KMyCustomCtlId;
            control = CUSTOMCONTROL
                {
                txt = STRING_r_custom_control_dialog;
                };
            }
        };
    }
```

`CMyDialog` dialog header that defines the needed `CreateCustomControlL()`

```
#include <akndialog.h>

class CMyDialog : public CAknDialog
    {
    public:
        static TInt RunDlgLD();
        void PreLayoutDynInitL();

    private:
```


(C) Copyright Nokia 2014. All rights reserved.

```
        SEikControlInfo CreateCustomControlL(TInt aControlType);
    };
```

`CMyDialog` dialog implementation.

```
#include "cmydialog.h"
#include "cmycontrol.h"
#include <MultiViews.rsg>
#include "MultiViews.hrh"

TInt CMyDialog::RunDlgLD()
    {
    CMyDialog* dlg = new (ELeave) CMyDialog;
    return dlg->ExecuteLD(R_DIALOG);
    }

void CMyDialog::PreLayoutDynInitL()
    {
    CMyControl* control = (CMyControl*)Control(KMyCustomCtlId);
    // TODO: tune components if needed
    }

SEikControlInfo CMyDialog::CreateCustomControlL(TInt aControlType)
    {
    SEikControlInfo controlInfo;
    controlInfo.iControl = NULL ;
    controlInfo.iTrailerTextId = 0 ;
    controlInfo.iFlags = 0 ;

    switch (aControlType)
        {
        // CMyControl custom control type (defined in multiviews.hrh)
        case KMyCustomCtl:
            {
            controlInfo.iControl = new(ELeave)CMyControl();
            break;
            }
        default:
            break;
        }
    return controlInfo;
    }
```

New enumeration values into `multiviews.hrh`

```
// CMyControl custom control type in resource file
enum {KMyCustomCtl = KAknCtLastControlId };
// CMyControl custom control id in resource file
enum {KMyCustomCtlId = 0x503 };
```

Launching the dialog.

```
CMyDialog::RunDlgLD();
```


(C) Copyright Nokia 2014. All rights reserved.

Postconditions

`CMyDialog` is created by the dialog resource and `CMyControl` is created into the dialog line by calling `CMyControl::ConstructFromResourceL()`. `CMyDialog` implements the `CEikDialog::CreateCustomControlL()` virtual method that creates the custom control into the dialog line.

TODO: How to enable skins behind the dialog

See also

Custom Control Series:

- How to create a custom control in Symbian C++ How to define custom control
- Constructing a Symbian custom control from a resource Creating a control from a resource
- Constructing a Symbian container control Creating a container control
- Changing the focus of a Symbian custom control Handling key events and changing active custom control focus
- Archived:Using scrollbars in Symbian container control Adding scroll bar to custom control
- File:CustomControl.zip Example code patch

(C) Copyright Nokia 2014. All rights reserved.