



| | | | |
|-----------------|----------------------|--------------------------|----------------|
| ID | CS000860 | Creation date | March 25, 2008 |
| Platform | S60 3rd Edition, FP1 | Tested on devices | Nokia N95 |
| Category | Symbian C++ | Subcategory | UI |

Keywords (APIs, classes, methods, functions): CCoeControl, TResourceReader

Overview

This code snippet shows how to create `CCoeControl` from a resource. This example extends the code snippet [CS000859 - Custom control](#).

`CMyControl::ConstructFromResourceL()` initiates the control in the same way as the `CMyControl::ConstructL()` method, but the parameters are obtained from `TResourceReader`. The resource reader is a utility class that reads bytes from a resource file stream; therefore, reading must be in the same order and the data sizes must match those defined in the resource structure declaration.

MMP file

The following capabilities and libraries are required:

```
LIBRARY bafl.lib //TResourceReader
```

Header

Add a new function that is called when creating the class from the resource and add the `BARSREAD.H` include. The default constructor `CMyControl()` should be moved to public.

```
#include <BARSREAD.H>

public:
    CMyControl();
    void ConstructFromResourceL(TResourceReader& aReader);
```

Source

This code reads text for CEikLabel from the resource.

```
void CMyControl::ConstructFromResourceL(TResourceReader& aReader)
{
    // No parent owner, so create an own window
    CreateWindowL();

    // Initialize component array
    InitComponentArrayL();

    // Create contained controls
    iStatusText = new (ELeave) CEikLabel;
    iStatusText->SetContainerWindowL(*this);

    // Read label from resource
    TPtrC label = aReader.ReadTPtrC16();
    iStatusText->SetTextL(label);

    // Store component to component array
    Components().AppendLC(iStatusText);
    CleanupStack::Pop(iStatusText);

    // Set component rect to CMultiViewsAppUi::ClientRect()
    CMultiViewsAppUi* appui = (static_cast<CMultiViewsAppUi*>(iEikonEnv->AppUi()));
    appui->ClientRect();
    if (appui)
    {
        SetRect(appui->ClientRect());
    }

    ActivateL();
}
```

CMyControl component resource customcontrol.rh file

Resource config parameters of the CMyControl component

```
STRUCT CUSTOMCONTROL
{
    LTEXT txt;
}
```

Resource .rss file

Make the following changes to the Multiviews example application resource file multiviews.rss.

```
// New include that is our own component resource config
#include "customcontrol.rh"

// Defining resource for our component.
RESOURCE CUSTOMCONTROL r_custom_control
{
    txt = STRING_r_custom_control;
```

```
}
```

STRING_r_custom_control is defined in multiviews.rls.

Creating component from resource

Create the resource reader, create the component via the default constructor, and call `ConstructFromResourceL()`

```
// Creating control from resource
TResourceReader reader;
iEikonEnv->CreateResourceReaderLC(reader, R_CUSTOM_CONTROL);
iContainer2 = new (ELeave) CMyControl;
iContainer2->ConstructFromResourceL(reader);
iContainer2->SetRect(ClientRect());
CleanupStack::PopAndDestroy(); // reader
```

Postconditions

`CMyControl` is created by defined resource parameters.

See also

Custom Control Series:

- [CS000859 - Custom control](#) Defining a custom control
- [CS000861 - Custom control: container control](#) Creating a container control
- [CS000868 - Custom control: Focusing](#) Handling key events and changing active custom control focus
- [CS000869 - Custom control: Scrollbars](#) Adding scroll bar to custom control
- [CS000870 - Custom control: In dialog](#) Adding a custom control into `CAknDialog`
- [File:CustomControl.zip](#) Example code patch