

Contents

- [1 Annotation](#)
- [2 Introduction](#)
- [3 RBufConsole Constructor](#)
- [4 RBufConsole Append\(\) method](#)
- [5 RBufConsole Show\(\) method](#)
- [6 Resource file for Message Query Dialog](#)
- [7 Usage examples](#)
- [8 Download this example](#)
- [9 Internal Links](#)
- [10 External Links](#)

Annotation

This example introduces to RBuf descriptors, observers paradigm. Has all-sufficient example file.

Introduction

Whilst your application does some actions it needs to generate messages for user or for debug purposes.

If you have GUI application you will need some text storage which can be filled up and be shown as GUI message. I've developed such a class called **RBufConsole** which is comfortable for the aims above and is a good example of **RBuf** Symbian descriptor.

RBuf is a Symbian descriptor designed to manipulate strings and binary data.

In this example RBuf is used for appending strings, reallocation memory buffer, passing function parameters.

In addition, some Active Object can save messages to the same storage while performing it's tasks. This is possible with **RBufConsoleObserver** interface class which relates to Observer paradigm.

So when Active object produces some action it calls Observer's callback function to add message to the *RBufConsole* storage.

After Active Object completes it can call the show message event or main program can do it.

RBufConsole Constructor

Assigns the resource (from rss file) for GUI Message (will be shown below) Initializes MessageStorage variable ? RBuf class instance.

```
CRBufConsole::CRBufConsole(TInt aResourceID)
{
    //Set resource ID from rss file
    MessageDialogResourceID = aResourceID;

    //initializaing storage
    TInt maxSizeOfMsgData = 150;
    _LIT(KGenesis, "App starts\n");
    MessageStorage.Create(KGenesis(), maxSizeOfMsgData);
}
```

RBufConsole Append() method

The Append() method will check if the new message can exceed the descriptor's size and will reallocate memory in this case

```
void CRBufConsole::Append(TPtrC16 appendString)
{
    const TInt newLength = MessageStorage.Length() + appendString.Length();

    if (MessageStorage.MaxLength() < newLength )
    {
        //Reallocate storage's memory if it is not enough
        MessageStorage.ReSize(newLength);
    }

    MessageStorage.Append(appendString);
}
```

RBufConsole Show() method

When programmer decides, he can show all messages stored in RBufConsole storage, in GUI Popup.

```
void CRBufConsole::Show()
{
    (KDialogHeader, "Output");

    RBuf title
    CAknMessageQueryDialog* new (ELeave)CAknMessageQueryDialog();
    ->PrepareLC(MessageDialogResourceID);

    Create(KDialogHeader());
    ->QueryHeading()->SetTextL(title);

    //Assign storage to be outputed
    ->SetMessageTextL(MessageStorage);
}
```

RBufConsole_class_example

```
        //show scrollable message
        ->RunDd();

        Close();

return;
}
```

Resource file for Message Query Dialog

In your .rss file you should define resource for CAknMessageQueryDialog. It would be passed as a parameter to constructor: CRBufConsole (R_RBUFCONSOLE_QUERY_DIALOG);

```
RESOURCE_DIALOG r_rbufconsole_query_dialog
{
    = EGeneralQueryFlags | EEikDialogFlagNoBorder | EEikDialogFlagNoShadow;
    = AVKON_SOFTKEYS_OK_EMPTY;
    = items
    {
        DLG_LINE
    {
        = EAknCtPopUpHeadingPane;
        = EAknMessageQueryHeaderId;
        = EEikDialogFlagNonFocusing;
        = AVKON_HEADING
    {
    };
    },
        DLG_LINE
    {
        = EAknCtMessageQuery;
        = EAknMessageQueryContentId;
        = AVKON_MESSAGE_QUERY
    {
    };
    }
};
```

Usage examples

Append new message:

```
iConsole.Append(_L("step 1\n"));
```

Show storage contents:

```
iConsole.Show(); //Shows RBufConsole contents as a GUI scrollable message
```

Execute Append() callback function inside Active Object:

```
iRBufConsoleObserver->ObserverAppend(_L("Active object completes\n"));
```

Download this example

Download Zip archive with full example: [File:WikiConsoleExample.zip](#) . It is based on Carbide C++ basic GUI application. I will grow functionality in the course of time.

To know how to import project from Zip [look here](#).

Internal Links

- [Observer](#)
- [How to get project from CVS to Carbide C++](#)

External Links

- [Introduction to RBuf](#)