

roMessageDialog

This component is deprecated.

Beginning July 1st, 2017, any new channels using this component will be rejected during certification.

Beginning January 1st, 2018, any updates to existing channels using this component will be rejected during certification.

Interfaces Supported

- [ifMessageDialog](#)
- [ifSetMessagePort](#)
- [ifGetMessagePort](#)

Events

- [roMessageDialogEvent](#)

Description

The Message Dialog displays a formatted, multi-line text message to the user. The dialog may optionally be displayed with a busy animation to indicate progress on a long running operation. The dialog will automatically handle formatting of text and resize to fit. It may also display buttons to get user acknowledgment or a selection choice.

The following example shows an roMessageDialog with a single done button. When the title, text and button are added, the dialog automatically formats and resizes the dialog as needed for display when Show() is called.

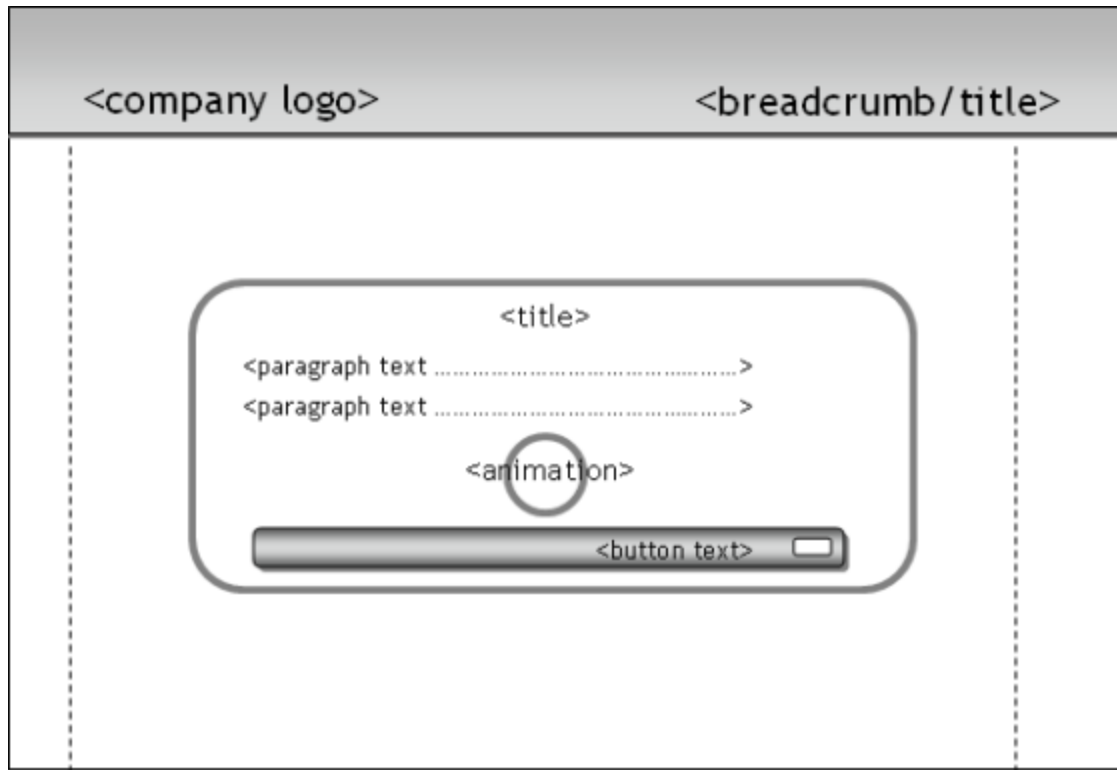


Diagram: `roMessageDialog`

The following code example creates a message dialog and displays it to the user. Note that dialogs are not full screen and that the previous screen is dimmed and displays in the background. When the user presses the message dialog button, the dialog is dismissed and the previous screen comes to the foreground. Since `dialog.EnableBackButton(true)` is also called, the message dialog is dismissed when the remote control's back button is pressed as well. You can of course add additional buttons to your message dialogs that do things other than dismiss the dialog. You would simply need to implement button specific event handling code for these cases in the `dlgMsg.isButtonPressed()` code block.

Example

```
Function ShowMessageDialog() As Void
    port = CreateObject("roMessagePort")
    dialog = CreateObject("roMessageDialog")
    dialog.SetMessagePort(port)
    dialog.SetTitle("[Message dialog title]")
    dialog.SetText("[Message dialog text.....]")

    dialog.AddButton(1, "[button text]")
    dialog.EnableBackButton(true)
    dialog.Show()
    While True
        dlgMsg = wait(0, dialog.GetMessagePort())
        If type(dlgMsg) = "roMessageDialogEvent"
            if dlgMsg.isButtonPressed()
                if dlgMsg.GetIndex() = 1
                    exit while
                end if
            else if dlgMsg.isScreenClosed()
                exit while
            end if
        end if
    end while
End Function
```

Image: roMessageDialog example results

