jRLDialogsX v1.87

B4J Additional Library

Copyright © 2015 – 2019, Robert Linn, Pinneberg, Germany

07.04.2019

Table of Contents

Overview 3
Getting Started 4
Properties 7
Methods 8
SetParentWindow 8
InformationDialog 9
ExtendedDialog 10
Properties 10
WarningDialog 11
ErrorDialog 12
ConfirmationDialog 13
YesNoCancelDialog 14
TextInputDialog 15
TextInputDialog2 16
TextInputDialog3 17
TextInputDialog4 18
MultiInputFieldDialog 19
SimpleFormDialog 20
Properties 22
IntegerInputDialog 23
ChoiceDialog 24
LoginDialog 25
LoginDialog2 26
DidYouKnowDialog 27
DidYouKnowDialog2 28
ExceptionDialog 29
MessageDialog 30
Properties 30
MessageHTMLDialog 31
Properties 31
MessageHTMLDialog2 32
MessageHTMLDialog3 33
ListDialog 34
ListDialog2 35
ListFindDialog 36
Properties 36

SelectDialog 38
SpinnerIntegerDialog 39
SpinnerDoubleDialog 40
SpinnerListDialog 41
TextAreaDialog 42
ToastMessage 43
Properties 43
ToastMessageAlert 44
Properties 45
DatePickerDialog 46
TimePicker24Dialog 47
TimePicker12Dialog 48
ColorNameDialog 49
ColorPickerDialog 50
SliderDialog 51
Properties 52
DoNotAskAgainDialog 54
Properties 54
Localisation Properties 55
Introduction 55
Username Label 55
Username Prompt 55
Password Label 55
Password Prompt 55
OK Button Text 56
Cancel Button Text 56
YES Button Text 56
NO Button Text 56
Previous Button Text 56
Next Button Text 56
Login Button Text 56
Select Button Text 56

Overview

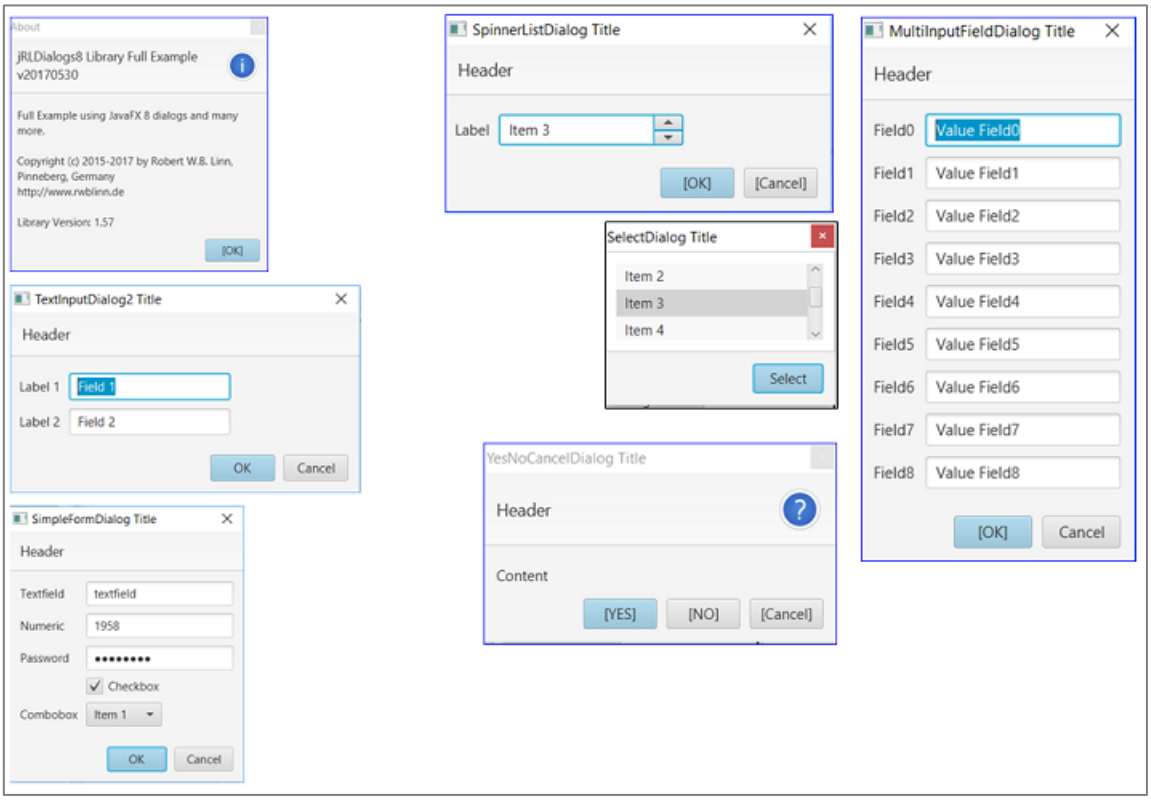
jRLDialogsX is an open source B4J Dialogs Library providing a comprehensive set of Dialogs.

B4J is a development tool for desktop, server and IoT solutions by Anywhere Software.

The library is written in **B4J** (requires v5.80 or higher) making (strong) use of Inline Java (requires Java 8 update 40 or higher) and published on the B4J Forum.

Note: Examples or screenshots use in cases the former library name jRLDialogs8.

Some Dialog Examples

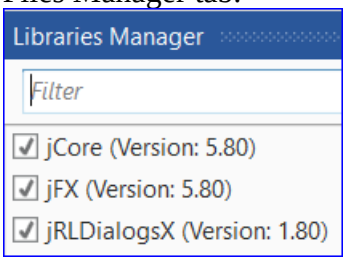


Getting Started

Install

- Download the B4J Library, Source Code and Examples.
- Unzip the jRLDialogsX.zip to a folder of choice.
- Copy from folder Library, the files jRLDialogsX.jar, jRLDialogsXFiles.jar and jRLDialogsX.xml to the B4J Additional Library Folder.

- The library jRLDialogsX, with version information, should be listed in the B4J IDE Files Manager tab.



- **jRLDialogsXFiles.jar**
  Since jRLDialogsX v1.80, the jRLDialogsXFiles.jar archive contains additional files (i.e. bitmaps, layouts) required by the library and must be placed in the B4J Additional Library Folder.
  Bitmaps used are: LoginDialog2 login.png, DidYouKnowDialog dyk.png.
  If other bitmaps (icons) are required, then update the file jRLDialogXFiles.jar in the additional library folder:
  Rename jRLDialogXFiles.jar to jRLDialogXFiles.zip, update the icons, save the zip file and rename back to jRLDialogXFiles.jar.

- IMPORTANT: jRLDialogs8 is the former name of this library.
  To update change the object declaration from Dialogs8 to DialogsX, use like:

```
'Define the dialog using jRLDialogsX
Private Dlg As DialogsX
```

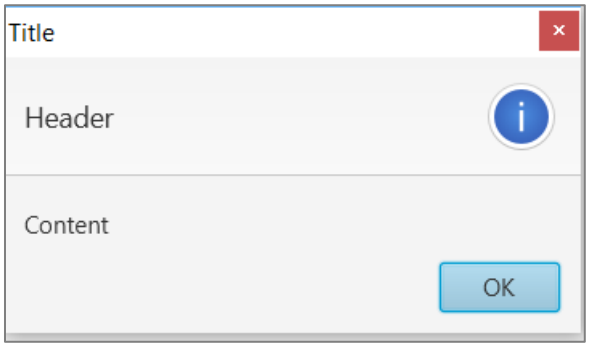
Examples

The folder Examples contains several samples.
For all dialogs in action, look at the project in the fullexample folder.

Example Simple Information Dialog

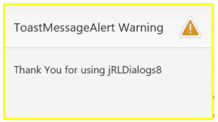
To create a dialog, define a DialogsX object, initialize the object, set parent window, set dialog properties (optional) and call the dialog method.

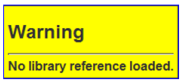
```
' Define the DialogsX object using jrlDialogsX
Private Dlg As DialogsX

' Initialize the object - DO NOT FORGET
Dlg.Initialize

' Set the parent window - the mainform
Dlg.SetParentWindow(MainForm)

' Call the object method with parameter
Dlg.InformationDialog("Title", "Header", "Content")
```

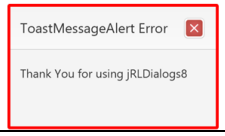


Screenshots of other Dialog Examples

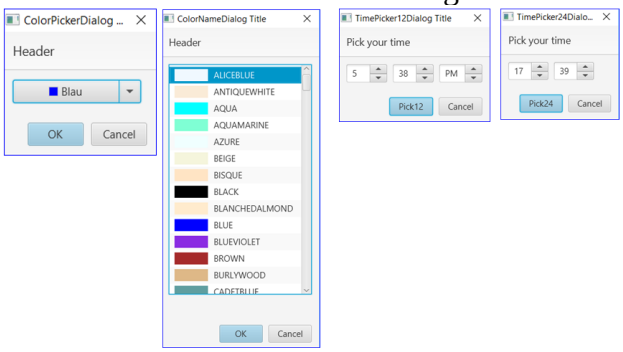
ToastMessage and ToastMessageAlert



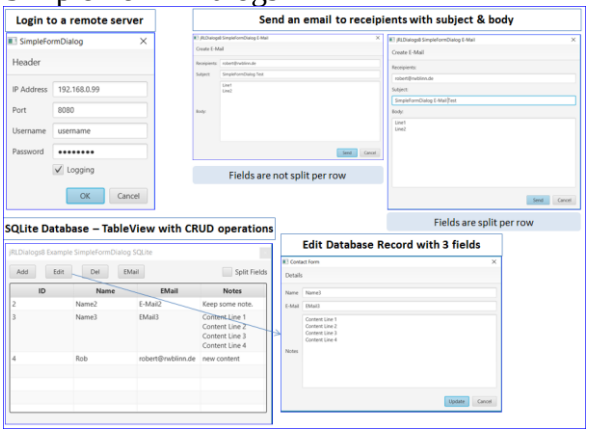




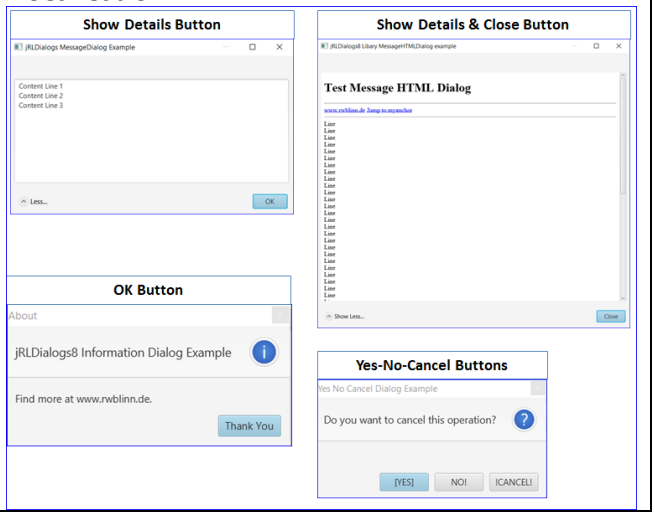
Color and Date/Time Picker Dialogs



Simple Form Dialogs



Localisation



Properties

Property	Brief	Example

Version as String	Get the library version	`Log($"${Dlg.version}"$)`
		`Output:`
		`1.76`

Methods

The list of library methods with parameter, their possible return values, additional notes and selective examples.

SetParentWindow

SetParentWindow(ParentForm As Form)

Set the parent window for the dialogs.

ParentForm - Form to be used as parent window.

Returns

None

Example

```
SetParentWindow(MainForm)
```

Notes

- To reset the parent window, call SetParentWindow(Null).
- The parent window cannot be set for the ToastMessage dialog.

InformationDialog

InformationDialog(Title As String , Header As String, Content As String)

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).

Returns

None

ExtendedDialog

ExtendedDialog(Title As String , Header As String, Content As String, ContentExtended As String)

Show an Extended Dialog which has an expandable text area.
The text area is by default not expanded.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).
ContentExtended - string to show in a text area if the clicked on the show details options.

Returns
None

Properties

Show content expanded

ExtendedDialogExpanded(Expanded As Boolean)
Expanded - False does not expand the text area with extended information.
ExtendedDialogExpanded As Boolean

Text for the Show more or less details hyperlink button

ExtendedDialogShowMoreDetailsText(Txt As String)
Txt - Text to display.
ExtendedDialogShowMoreDetailsText As String
ExtendedDialogShowLessDetailsText(Txt As String)
Txt - Text to display.
ExtendedDialogShowLessDetailsText As String

Show or hide the less details hyperlink button

ExtendedDialogHideDetails(Hide As Boolean)
Hide - Hide (False) or Show (True) the details hyperlink button.
ExtendedDialogHideDetails As Boolean

WarningDialog

WarningDialog(Title As String , Header As String, Content As String)

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).

Returns

None

ErrorDialog

ErrorDialog(Title As String , Header As String, Content As String)

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).

Returns

None

ConfirmationDialog

ConfirmationDialog(Title As String , Header As String, Content As String) As Boolean

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).

Returns

True for OK
False for Cancel

YesNoCancelDialog

YesNoCancelDialog(Title As String , Header As String, Content As String) As Int

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).

Returns

Integer
Yes = 1
No = 0
Cancel = -1.

TextInputDialog

TextInputDialog(Title As String , Header As String, Label As String, Text As String) As String

Text Input Dialog with one field.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - string to show as label left from the input field.
Text - string as default for the input field.

Returns

Text entered as string or an empty string

TextInputDialog2

TextInputDialog2(Title As String , Header As String, Label1 As String, Label2 As String, Field1 As String, Field2 As String) As Map

Text Input Dialog with two fields.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label1 - Label for field 1
Label2 - Label for field 2
Field1 - Default values for field1.
Field2 - Default values for field2.

Returns

Map with 2 entries holding field1:textinputfield1, field2:textinputfield2.
If cancelled a non initialized Map.

Example

```
Dim m As Map = Dlg.TextInputDialog2("Text Input Dialog 2", "Name & EMail",
"Name", "EMail", "name", "email")

Log($"${m.Get("field1")} ${m.Get("field2")}"$)
```

TextInputDialog3

TextInputDialog3(Title As String , Header As String, Label As String, Text As String) As String

Text Input Dialog with one field with user cancel handling.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - string to show as label left from the input field.
Text - string as default for the input field.

Returns

Text entered as string or an empty string.

Notes

To test the result of the dialog, use i.e.

```
If result.EqualsIgnoreCase(Null) Then ...
```

TextInputDialog4

TextInputDialog4(Title As String , Header As String, Text As String, Width As Int) As String

Text Input Dialog with one field with user cancel handling.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Text - string as default for the input field.
Width - Width of the input field. Set to 0 to use the default width.

Returns

Text entered as string or an empty string.

Notes

To test the result of the dialog, use i.e.

```
If result.EqualsIgnoreCase(Null) Then ...
```

Example

```
Dim url As String = Dlg.TextInputDialog4("Text Input Dialog 4", "Enter the
URL", "http://www.rwblinn.de", 600)

lblTextInputDialog.Text = url
```

MultiInputFieldDialog

MultiInputFieldDialog(Title As String , Header As String, Fields As Map, FieldCount As Int) As Map

Multi Input Field build from a field map.

Title - string to show in the title bar.
Fields - map which holds for each field the pair label:text.
FieldCount - size of the field map.

Returns

Map holding for each field the pair fieldN:fieldtext.
If cancelled, a non initialized map.
N is the field number, starting with 0.
The max number of fields is determined by the field map size.

Example

```
SetParentWindow(MainForm)

Dlg.OKButtonText = tfOKButtonText.Text

Dim fieldmap As Map = CreateMap("FieldA": "Value FieldA", "FieldB": "Value
FieldB", "FieldC": "Value FieldC", "FieldD": "Value FieldD")

Dim resultmap As Map = Dlg.MultiInputFieldDialog("MultiInputField Title",
"Header", fieldmap, fieldmap.size)
```

The result map holds for each field the pair fieldN:text

```
For i = 0 To resultmap.Size - 1
 Log($"${resultmap.GetKeyAt(i)} = ${resultmap.GetValueAt(i)}"$)
Next
```

SimpleFormDialog

SimpleFormDialog(Title As String , Header As String, FieldList As List) As Map

Simple Form Dialog enabling to use the field types Text, Password, Numeric, CheckBox, ComboBox.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
FieldList - list of maps. Each map represents a field with id, label, value, fieldtype.
The fieldtypes (must be uppercase) TextField "T", Password "P" , Numeric "N", TextArea "A", CheckBox "C", ComboBox "B", Datepicker “D”

Field Notes

- CheckBox: field value “1” or ”0”, means checked or unchecked
- ComboBox: field value comma separated string, i.e. "Item 1,Item 2,Item 3"

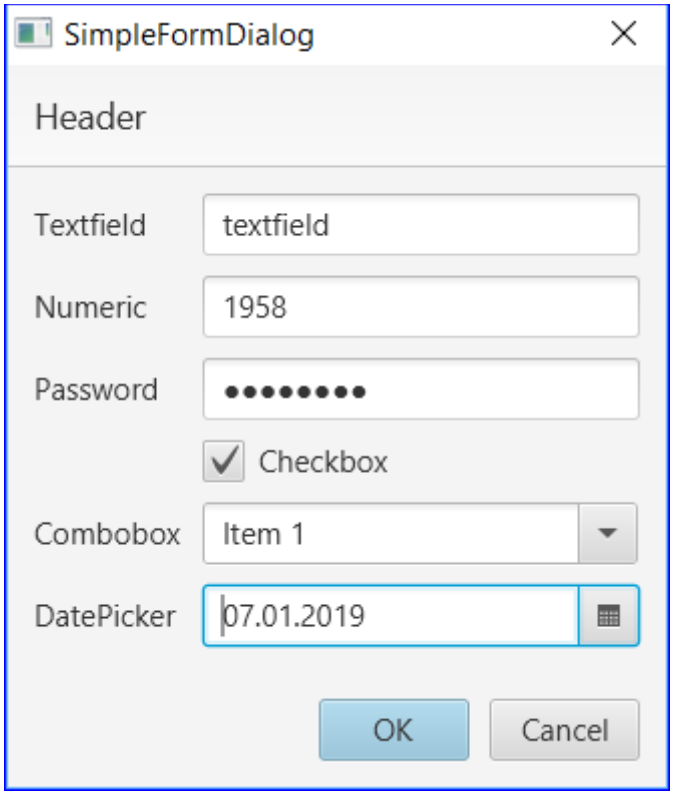
Returns

Map holding for each field the key:value pair id:value.
The returned field values are strings, means for numeric fields conversion (string to number) has to be done manually.

Example

```
'Create the list of fieldmaps
Dim fieldlist As List
fieldlist.Initialize

fieldlist.Add(CreateMap("id":"textfield", "label":"Textfield",
"value":"textfield", "type":"T"))

fieldlist.Add(CreateMap("id":"numeric", "label":"Numeric", "value":"1958",
"type":"N"))

fieldlist.Add(CreateMap("id":"password", "label":"Password",
"value":"password", "type":"P"))

fieldlist.Add(CreateMap("id":"checkbox", "label":"Checkbox", "value":"1",
"type":"C"))

fieldlist.Add(CreateMap("id":"combobox", "label":"Combobox", "value":"Item
1,Item 2,Item 3", "type":"B"))

'IMPORTANT TO SET THIS FORMAT prior adding datepicker
DateTime.DateFormat = "yyyy-MM-dd"
fieldlist.Add(CreateMap("id":"datepicker", "label":"DatePicker",
"value":"2019-01-01", "type":"D"))

'Open the dialog
Dlg.SimpleFormSplitFields = True
Dlg.SimpleFormComboBoxEditable = True
Dim resultmap As Map = Dlg.SimpleFormDialog("SimpleFormDialog", "Header",
fieldlist)
```

```
'The resultmap holds for each field the pair id:value
If resultmap.IsInitialized Then
  Dim sb As StringBuilder
  sb.Initialize
  For i = 0 To resultmap.Size - 1
    sb.Append($"${resultmap.GetKeyAt(i)} =
${resultmap.GetValueAt(i)}"$).Append(CRLF)
  Next
  Log(sb.tostring)
Else
  Log("User Abort")
End If
```



Output after selecting OK

```
textfield = textfield
numeric = 1958
password = password
checkbox = 1
combobox = Item 1
datepicker = 2019-01-07
```

Properties

Content per Row

SimpleFormSplitFields(Value As Boolean)
If value false, then row contains label : field else row contains label or field.
SimpleFormSplitFields As Boolean

Input Field Width

SimpleFormWidth(Width As Int)
Width - Input field width.
SimpleFormWidth As Int

ComboBox editable

SimpleFormComboBoxEditable(editable As Boolean)
Editable - enables editing of the combobox input field.
SimpleFormComboBoxEditable As Boolean

IntegerInputDialog

IntegerInputDialog(Title As String , Header As String, Label As String, Default As Int) As Int

Show a Integer Input Dialog

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - string to show as label left from the input field.
Default - default value as integer in range -32768 to +32767.

Returns

Value or the default if cancelled

Example

```
Dim default As Int = 9

Dim result As Int = IntegerInputDialog("jRLDialogsX Integer Input Dialog",
"Header", "Label", default)

If result <> default Then
 Log(result)
End If
```

ChoiceDialog

ChoiceDialog(Title As String , Header As String, Label As String, Choices As List, Defaultitem As Int) As String

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - string to show as label left from the input field.
Choices - List
Defaultitem - Selected item between 0 and choices.size - 1.

Returns

The selection as string or an empty string if cancelled

LoginDialog

LoginDialog(Title As String , Header As String) As Map

Title - string to show in the title bar.
Header - string to show in the dialog header area.

Returns

Map
username:username, password:password

If cancel selected, the returned map will not be initialized.

LoginDialog2

LoginDialog2(Title As String , Header As String, UserName As String) As Map

Login Dialog with Icon and default UserName.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Username - string as the default username.

Returns

Map
username:username, password:password.

If cancel selected, the returned map will not be initialized.

Notes

The **icon login.png** is required, which is included in the additional jar jRLDialogsXFiles.jar.
To change the icon, update the file jRLDialogsXFiles.jar, located in the B4J additional libraries folder:
Rename jRLDialogsXFiles.jar to jRLDialogsXFiles.zip, extract the content, update login.png, zip the content back to jRLDialogsXFiles.zip and finally rename jRLDialogsXFiles.zip to jRLDialogsXFiles.jar.

DidYouKnowDialog

DidYouKnowDialog(Title As String , Header As String, Content As String) As Map

Did You Know Dialog with icon.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).

Returns

None

Notes

The **icon dyk.png** is required, which is included in the additional jar jRLDialogsXFiles.jar.
To change the icon, update the file jRLDialogsXFiles.jar, located in the B4J additional libraries folder:
Rename jRLDialogsXFiles.jar to jRLDialogsXFiles.zip, extract the content, update dyk.png, zip the content back to jRLDialogsXFiles.zip and finally rename jRLDialogsXFiles.zip to jRLDialogsXFiles.jar.

DidYouKnowDialog2

DidYouKnowDialog2(Title As String , Header As String, Items As List, DefaultItem As Int, Width As Double, Height As Double)) As Int

Did You Know Dialog with items list to move to previous or next item and an icon.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Items - the items as list
DefaultItem - set the default item between 0 and items.size - 1 or -1 to set the first item
Width - the dialog width. -1 sets default 500.
Height - the dialog height. -1 sets default 400

Returns

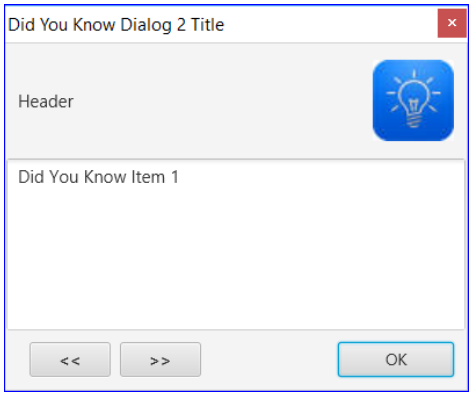
Integer - Index of the last select Did You Know item

Notes

The **icon dyk.png** is required, which is included in the additional jar jRLDialogsXFiles.jar.
To change the icon, update the file jRLDialogsXFiles.jar, located in the B4J additional libraries folder:
Rename jRLDialogsXFiles.jar to jRLDialogsXFiles.zip, extract the content, update dyk.png, zip the content back to jRLDialogsXFiles.zip and finally rename jRLDialogsXFiles.zip to jRLDialogsXFiles.jar.

Sample Code

```
Private Dlg As DialogsX
Private mItem As Int = 0
Dlg.Initialize
Dlg.SetParentWindow(MainForm)
Dlg.OKButtonText = "OK"
Dlg.PreviousButtonText = "<<"
Dlg.NextButtonText = ">>"
Dim l As List : l.Initialize : l.AddAll(Array As String ("Did You Know Item
1", "Did You Know Item 2", "Did You Know Item 3"))
Dim result As Int = Dlg.DidYouKnowDialog2("Did You Know Dialog 2 Title",
"Header", l, mItem, 400, 300)
mItem = result
Log($"Item Index selected: ${mItem} which is ${l.Get(mItem)}"$)
```



ExceptionDialog

ExceptionDialog(Title As String , Header As String, Content As String, ExceptionContent As Exception)

Exception Dialog in an expanded text area.
Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).
ExceptionContent - string containing the exception shown in the text area.

Returns

None

Notes

For the Content, an option is to use LastException.Message
For the ExceptionContent, the LastException can be used.

MessageDialog

MessageDialog(Title As String, Content As String)

Title - string to show in the title bar.
Content - string with plain text in an expanded text area.

Returns

None

Properties

Show content expanded

MessageDialogExpanded(Expanded As Boolean)
Expanded - False does not expand the textarea with extended information.
MessageDialogExpanded As Boolean

Set the text for the show more or less details hyperlink button

MessageDialogShowMoreDetailsText(Txt As String)
Txt - Text to display.
MessageDialogShowMoreDetailsText As String
MessageDialogShowLessDetailsText(Txt As String)
Txt - Text to display.
MessageDialogShowLessDetailsText As String

Wrap the content

MessageDialogWrapText(Wrap as Boolean)
Wrap - False does not wrap the content textarea.
MessageDialogWrapText As Boolean

MessageHTMLDialog

MessageHTMLDialog(Title As String, Content As String)

Title - string to show in the title bar.
Content - string with HTML formatted content in an expanded web view.

Returns

None

Notes

Anchors are handled if defined like

```
<a href="#myanchor">Jump To myanchor</a> ... <a name="myanchor"></a>
```

Properties

Show content expanded

MessageHTMLDialogExpanded(Expanded As Boolean)
Expanded - False does not expand the textarea with extended information.
MessageHTMLDialogExpanded As Boolean

Set the text of the show more or less details hyperlink button

MessageHTMLDialogShowMoreDetailsText(Txt As String)
Txt - Text to display.
MessageHTMLDialogShowMoreDetailsText As String
MessageHTMLDialogShowLessDetailsText(Txt As String)
Txt - Text to display.
MessageHTMLDialogShowLessDetailsText As String

Show or hide the less details hyperlink button

MessageHTMLDialogHideDetails(Hide As Boolean)
Hide - Hide (False) or Show (True) the hyperlink button.
MessageHTMLDialogHideDetails As Boolean

MessageHTMLDialog2

MessageHTMLDialog2(Title As String, Header As String, Content As String)

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string with HTML formatted content in an expanded web view.

Returns

None

Notes

The content is always expanded.
The anchors are handled if defined like

```
<a href="#myanchor">Jump To myanchor</a> ... <a name="myanchor"></a>
```

MessageHTMLDialog3

MessageHTMLDialog3(Title As String, Content As String)

Same as the MessageHTMLDialog, but with OK and Cancel buttons.

Title - string to show in the title bar.
Content - string with HTML formatted content in an expanded webview.

Returns
True for OK button, False for Cancel button.

Notes

Anchors are handled if defined like

```
<a href="#myanchor">Jump To myanchor</a> ... <a name="myanchor"></a>
```

The same properties can be set as the MessageHTMLDialog

ListDialog

ListDialog(Title As String, Header As String, Items As List, Defaultitem As Int) As String

List Dialog with single selection.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Items - List of items to select
Defaultitem - set between 0 - items.size - 1. If -1, then no item selected.

Returns

The selected item as String or an empty String

ListDialog2

ListDialog2(Title As String, Header As String, Items As List, Defaultitem As Int) As Map

List Dialog with multiple selection.
Title - string to show in the title bar.
Header - string to show in the dialog header area.
Items - List of items to select
DefaultItem - set between 0 - items.size - 1. If -1, then no item selected.

Returns

One or more selected items as a Map (nr:item) or a non-initialized Map.

ListFindDialog

ListFindDialog(Title As String, Items As List, DefaultItem As Int, Width As Double, Height As Double) As String

A List Find Dialog with option to find an item in the given list.
Title - set the title text as String
Items - the items as List
DefaultItem - set the default item between 0 and items.size - 1 or -1 to set the first item
Width - the dialog width. -1 sets as default 300
Height - the dialog height. -1 sets as default 400

Properties

The properties are accessible using an ListFindDialog object (see example below).

ItemSelected(Item As String)
Set or get the item selected in the listview.
Item - text of the item to be selected

Title(Title As String)
Set or get the text of the title.

FindItem(Value As String)
Set or get the content of the textfield FindItem

FindItemPromptText(PromptText As String)
Set or get the text of the textfield find item.

Button_CancelText(Text As String)
Set or get the text of the cancel button.

Button_SelectText(Text As String)
Set or get the text of the select button.

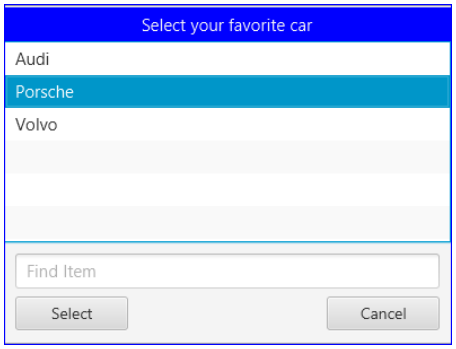
TitleStyle(Style As String)
Set or get the style property of the title label.
Style Examples

```
Dim lfd As ListFindDialog
lfd.TitleStyle = "-fx-text-fill:blue;-fx-background-color:white;"
or
lfd.TitleStyle = "-fx-text-fill:black;-fx-background-color:white;-fx-font-weight: bold;"
```

Returns

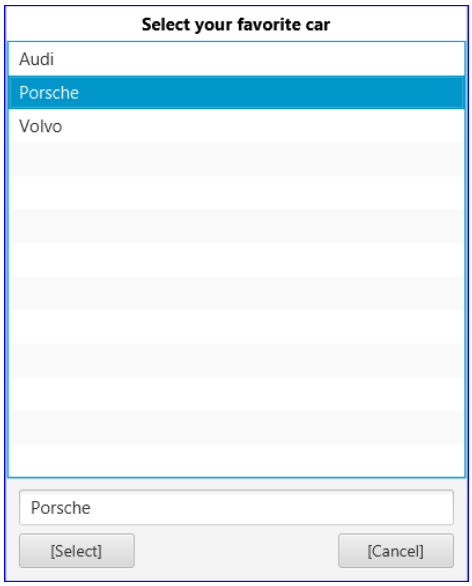
The selected item as string or an empty string (if the dialog was cancelled).

Example using the global object Dlg as DialogX




```
Dlg.SetParentWindow(MainForm)
Dlg.OKButtonText = "Select"
Dlg.CancelButtonText = "Cancel"
Dim items As List : items.Initialize : items.AddAll(Array As String
("Audi", "Porsche", "Volvo"))
Dim result As String = Dlg.ListFindDialog("Select your favorite car",
items, 1, 400, 300)
lblListFindDialog.Text = $"Car selected: ${result}"$
```

Example using a local object lfd as ListFindDialog




```
Dim lfd As ListFindDialog
Dim items As List : items.Initialize : items.AddAll(Array As String
("Audi", "Porsche", "Volvo"))
Dim FindCar As String = "Porsche"
lfd.Initialize(MainForm, "Select your favorite car", items, 0, 400, 500)
lfd.Button_SelectText = "[Select]"
lfd.Button_CancelText = "[Cancel]"
lfd.TitleStyle = "-fx-text-fill:black;-fx-background-color:white;-fx-font-
weight: bold;"
lfd.FindItem = FindCar
Dim result As String = lfd.ShowAndWait
If result.Length = 0 Then Return
FindCar = lfd.FindItem
lblListFindDialog.Text = $"Car selected: ${result}"$
```

SelectDialog

SelectDialog(Title As String, Items As List, Defaultitem As Int, Height As Double, Width As Double) As Int

Title - string to show in the title bar.
Items - List of items to select
Height - Dialog height
Width - Dialog width.

Returns

Selected item as Int or -1 if nothing selected

SpinnerIntegerDialog

SpinnerIntegerDialog(Title As String, Header As String, Label As String, MinValue As Int, MaxValue As Int, InitialValue As Int, AmountToStepBy As Int) As Int

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - Label shown left from the spinner.
MinValue - Spinner minimum value.
MaxValue - Spinner maximum value (must be greater minimum value).
InitialValue - Integer as the defalt value.
AmountToStepBy - Value to increase or decrease when stepping up or down.

Returns

Integer with selected value or -1 if cancelled

SpinnerDoubleDialog

SpinnerDoubleDialog(Title As String, Header As String, Label As String, MinValue As Double, MaxValue As Double, InitialValue As Double, AmountToStepBy As Double) As Double

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - Label shown left from the spinner.
MinValue - Spinner minimum value.
MaxValue - Spinner maximum value (must be greater minimum value).
InitialValue - Integer as the defalt value.
AmountToStepBy - Value to increase or decrease when stepping up or down.

Returns

Double with selected value or -1 if cancelled

SpinnerListDialog

SpinnerListDialog(Title As String, Header As String, Label As String, Items As List) As String

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - Label shown left from the spinner.
Items - List of items to select from.

Returns

String with selected value or empty string if cancelled

TextAreaDialog

TextAreaDialog(Title As String , Header As String, Label As String, Text As String) As String

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - Label shown left from the text area.
Text - Content of the text area.

Returns

Text entered as string or a null string if dialog cancelled

Notes

To test the result of the dialog, use i.e.

```
If result.EqualsIgnoreCase(Null) Then ...
```

ToastMessage

ToastMessage(Message As String , Duration As Int) As String

Message - String which can include html tags (ensure to include the start <html> and end </html> tags.
Duration - Miliseconds for keeping the message visible. Must be greater 0.

Returns

None

Example

```
Dlg.ToastMessageBackgroundColor = FX.Colors.Yellow
Dlg.ToastMessageFontSize = 16

Dlg.ToastMessage("<html><h1>Warning</h1><hr><p>This is a Toastmessage with
a duration of 5 seconds.</p></html>", 5000)
```

Properties

Message font size

ToastMessageFontSize(size As Int)
Fontsize - set as px
ToastMessageFontSize As Int

Message border width

ToastMessageBorderWidth(width As Int)
Width - set to 0 for no border
ToastMessageBorderWidth As Int

Background color

ToastMessageBackgroundColor(color As Paint)
Color - Value as paint, i.e. fx.colors.blue. ToastMessageBackgroundColor() As Paint

Border line color

ToastMessageBorderColor(color As Paint)
Color - Value as paint, i.e. fx.colors.blue. ToastMessageBorderColor() As Paint

ToastMessageAlert

ToastMessageAlert(AlertType As String, Header As String, Content As String, Duration As Int)

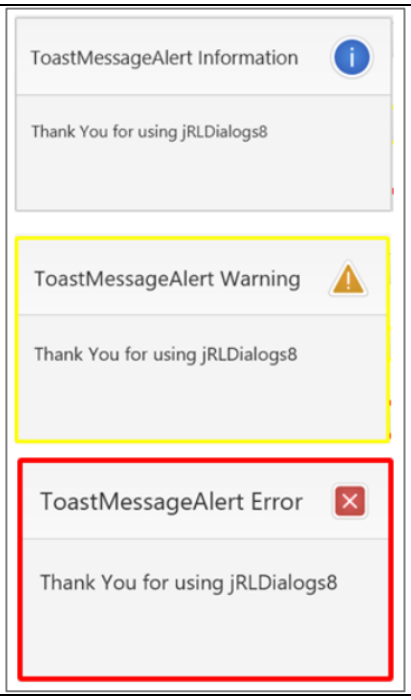
Display a ToastMessage using the standard JavaFX Alert Dialog method.

AlertType - String, types are Information, Warning, Error
Header - String, Header displayed above the content
Content - String, Content
Miliseconds - int, duration of the dialog being displayed

Returns

None

Example

This example shows the Alert for 5 seconds after pressing a buttons. The library CSSUtils is used to set the border style.	

```
Private Dlg As DialogsX
Private btnToastMessageAlertError As Button
Private btnToastMessageAlertInformation As Button
Private btnToastMessageAlertWarning As Button
Private Const Duration As Int = 5000

Dlg.Initialize

CSSUtils.SetBorder(btnToastMessageAlertInformation,
2,fx.Colors.LightGray,2)
CSSUtils.SetBorder(btnToastMessageAlertWarning, 4,fx.Colors.Yellow,2)
CSSUtils.SetBorder(btnToastMessageAlertError, 6,fx.Colors.Red,2)
```

```
Sub btnToastMessageAlertInformation_Action
 Dlg.ToastMessageAlertStyle(CSSUtils.ColorToHex(fx.Colors.LightGray),
2,2,16)
 Dlg.ToastMessageAlert("information", "Information", "Content", Duration)
End Sub

Sub btnToastMessageAlertWarning_Action
 Dlg.ToastMessageAlertStyle(CSSUtils.ColorToHex(fx.Colors.Yellow), 4,2,20)
 Dlg.ToastMessageAlert("warning", "Warning", "Content", Duration)
End Sub

Sub btnToastMessageAlertError_Action
 Dlg.ToastMessageAlertStyle(CSSUtils.ColorToHex(fx.Colors.Red), 6,2,24)
 Dlg.ToastMessageAlert("error", "Error", "Content", Duration)
End Sub
```

Properties

ToastMessageAlertStyle

ToastMessageAlertStyle(BorderColor As String, BorderWidth As Int, BorderRadius As Int, Fontsize As Int)

Property to get / set the style properties of an Alert Dialog.

BorderColor - String, i.e. BLUE
BorderWidth - int, default 1
BorderRadius - int, default 0
Fontsize - int, default 16

DatePickerDialog

DatePickerDialog(Title As String, Header As String, Label As String, DefaultDate As String) As String

DatePicker Dialog with default date, weeknumbers.

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - Label shown left from the date picker.

Returns

String with selected date or an empty string if cancelled.

Notes

It is important to set the date pattern prior calling this method.
If the default date (DefaultDate) left empty (""), the actual date is set.

TimePicker24Dialog

TimePicker24Dialog(Title As String, Header As String, Hours As Int, Minutes As Int, SetNow As Boolean) As String

24hr Time Picker Dialog with default values hours, minutes

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Hours - 0-23
Minutes - 0-59
SetNow - True = Set the actual time (parameter Hours, Minutes are omitted).

Returns

Time String HH:mm or an empty string if cancelled.

Example

```
Dlg.OKButtonText = "Pick"

'Pick a time with current time as default. The given hours and minutes are
not used.
Dim timepicked As String = Dlg.TimePicker24Dialog("TimePicker", "Pick your
time", 0,0, True)

'Pick a time with a default time set 17:08.
Dim timepicked As String = Dlg.TimePicker24Dialog("TimePicker", "Pick your
time", 17,8, False)
```

TimePicker12Dialog

TimePicker12Dialog(Title As String, Header As String, Hours As Int, Minutes As Int, AMPM As String, SetNow As Boolean) As String

12hr Time Picker Dialog with default values hours, minutes, AMPM

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Hours - 0-11
Minutes - 0-59
AMPM - "AM" or "PM"
SetNow - True = Set the actual time (parameter Hours, Minutes are omitted).

Returns

Time String HH:mm:ss or empty string if cancelled

Example

```
Dlg.OKButtonText = "Pick"

'Pick a time with current time as default. The given hours and minutes are
not used.
Dim timepicked As String = Dlg.TimePicker12Dialog("TimePicker", "Pick your
time", 0,0,"", True)

'Pick a time with a default time set 0:08 AM
Dim timepicked As String = Dlg.TimePicker12Dialog("TimePicker", "Pick your
time", 10,8,"AM", False)
```

ColorNameDialog

ColorNameDialog(Title As String , Header As String, Default As Int) As Paint

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Default - Color index to show as default selected.

Returns

Selected color as paint or null if cancelled

Example

```
Dim default As Int = 9

Dim color As Paint = ColorNameDialog("jRLDialogsX Color Dialog", "Header",
default)

If color.IsInitialized Then
 Dim ColorInt As Int = fx.Colors.To32Bit(color)
 Dim HexARGB As String = Bit.ToHexString(ColorInt)
 Dim HexRGB As String = HexARGB.SubString2(2,8)
 Log($"Color Int:${ColorInt} ${CRLF} HEXARGB:${HexARGB}
${CRLF}HexRGB:${HexRGB}"$)
End If
```

ColorPickerDialog

ColorPickerDialog(Message As String , Header As String, Default As Paint) As Paint

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Default - Color as Paint to show as default selected.

Returns

Selected color as paint or null if cancelled

Example

```
Dim default As Int = 9

Dim color As Paint = ColorPickerDialog("jRLDialogsX Color Picker Dialog",
"Header", fx.Colors.blue)

If color.IsInitialized Then
 Dim ColorInt As Int = fx.Colors.To32Bit(color)
 Dim HexARGB As String = Bit.ToHexString(ColorInt)
 Dim HexRGB As String = HexARGB.SubString2(2,8)
 Log($"Color Int:${ColorInt} ${CRLF}HEXARGB:${HexARGB}
${CRLF}HexRGB:${HexRGB}"$)
End If
```

SliderDialog

SliderDialog(Title As String, Header As String, MinValue As Double, MaxValue As Double, DefaultValue As Double) As Double

Title - string to show in the title bar.
Header - string to show in the dialog header area.
Label - Label shown left from the spinner.
MinValue - Minimum value of the slider (start value).
MaxValue - Maximum value of the slider (end value).
DefaultValue - Default value set in range Min / Max value.

Returns

Slider value As Double or if cancelled the default value given

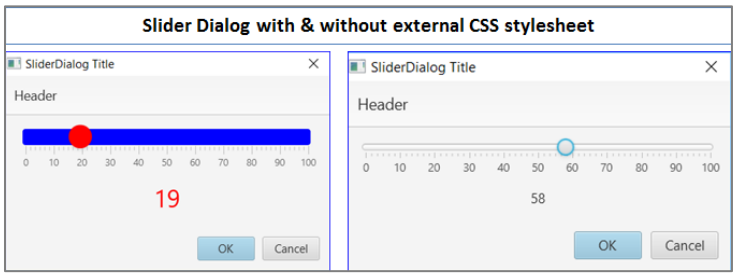
Notes

The slider style can be customized by using an external CSS file called Slider.css (case sensitive) which must be placed in the B4J Project Files (DirAssets) folder.

Example

```
Dim defaultValue As Double = 58
Dlg.SliderShowTickLabels = True
Dlg.SliderShowTickMarks = True
Dlg.SliderShowValue = True
Dlg.SliderBlockIncrement = 10
Dlg.SliderMajorTickUnit = 10
Dlg.SliderValueStyle = "-fx-text-fill: red; -fx-font-size: 24pt;"

Dim value As Double = Dlg.SliderDialog("SliderDialog Title", "Header", 0,
100, defaultValue)

If value <> defaultValue Then
 Log($"Slider Value:${CRLF}${value}"$)
Else
 Log($"User Abort.${CRLF}Selected Value ${value} = Default Value
${defaultValue}"$)
End If
```

Other Examples



Properties

Block increment step

SliderBlockIncrement(increment As Double)
Increment - amount by which to adjust the slider if the track of the slider is clicked.
SliderBlockIncrement As Double

Major tick unit

SliderMajorTickUnit(unit As Double)
Unit - unit distance between major tick marks.
SliderMajorTickUnit As Double

Minor tick count

SliderMinorTickCount(count As Int)
Count - number of minor ticks to place between any two major ticks.
SliderMinorTickCount As Int

Show tick labels

SliderShowTickLabels(show As Boolean)
Show - Indicates that the labels for tick marks should be shown.
SliderShowTickLabels As Boolean

Show tick marks

SliderShowTickMarks(show As Boolean)
Show - Specifies whether the Skin implementation should show tick marks.
SliderShowTickMarks As Boolean

Snap to ticks

SliderSnapToTicks(snap As Boolean)
Snap - Indicates whether the value of the Slider should always be aligned with the tick marks.
SliderSnapToTicks As Boolean

Show value in a separate label

SliderShowValue(show As Boolean)
Show - Value in a label.
SliderShowValue As Boolean

Slider Width

SliderPrefWidth(width As Double)

Width - Slider width. If 0, then the default setting is used.
SliderPrefWidth As Double

CSS style value label

SliderValueStyle(style As String) {
Style - CSS style, i.e. "-fx-text-fill: red; -fx-font-size: 24pt;".
SliderValueStyle As String

DoNotAskAgainDialog

DoNotAskAgainDialog(Title As String, Header As String, Content As String, DoNotAskAgainMessage As String, DefaultOption As Boolean) As Boolean

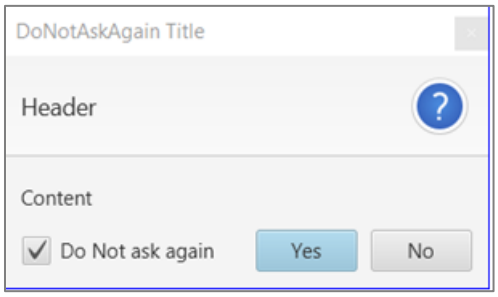
Title - string to show in the title bar.
Header - string to show in the dialog header area.
Content - string to show in the content area (below the header).
DoNotAskAgainMessage - string to show the option of not asking this dialog again.
DefaultOption - boolean to indicate the choice of not asking this dialog again.

Returns

1. From the Buttons: True = Yes, False = No
2. From the DoNotAskAgain Checkbox: True if checked which means, do not ask again to show the dialog.

Example

```
Dim result As Boolean = Dlg.DoNotAskAgainDialog("DoNotAskMeAgain Title",
"Header", "Content", "Do Not ask Me again", True)

Dim resultOption As Boolean = Dlg.DoNotAskAgainOption

Log($"Do Not Ask Me
Again${CRLF}Result:${result}${CRLF}Option:${resultOption}"$)
```



Properties

Option for not asking again.

DoNotAskAgainOption(askagain As Boolean)
DoNotAskAgainOption As Boolean

Localisation Properties

Introduction

The localisation properties are used to set the text of the generic buttons and labels.
These are i.e. OKButtonText, CancelButtonText.

Dialog specific localization are described under the relevant dialog, i.e. ExtendedDialog, MessageDialog, MessageHTMLDialog.

Text Property Re-use

If a text property is set for a dialog, the same text is re-used if another dialog is called.
Ensure to (re)set the text property prior calling the dialog. Here an example for an OK Button Text:

```
Dlg.OKButtonText = "YES"
Dlg.CancelButtonText = "NO"

Dim result As Boolean = Dlg.ConfirmationDialog("Confirmation", "Please
Confirm", "Content Confirmation Dialog example.")

'Set the OK button text for the next dialog
'If not the case, then YES is used as previous defined.

Dlg.OKButtonText = "Thank You"
Dlg.InformationDialog("Confirmation Result", "", $"Result Confirmation
Dialog is ${result}."$)
```

Username Label

Dlg.UsernameLabel = "Username:"
Dim text as String = Dlg.UsernameLabel

Username Prompt

Dlg.UsernamePrompt = "Enter Username"
Dim text as String = Dlg.UsernamePrompt

Password Label

Dlg.PasswordLabel = "Password:"
Dim text as String = Dlg.PasswordLabel

Password Prompt

```
Dlg.PasswordPrompt = "Enter Password"
Dim texts as String = Dlg.PasswordPrompt
```

OK Button Text

```
Dlg.OKButtonText = "OK"
Dim text as String = Dlg.OKButtonText
```

Cancel Button Text

```
Dlg.CancelButtonText = "Cancel"
Dim text as String = Dlg.CancelButtonText
```

YES Button Text

```
Dlg.YesButtonText = "YES"
Dim text as String = Dlg.YesButtonText
```

NO Button Text

```
Dlg.NoButtonText = "NO"
Dim text as String = Dlg.NoButtonText
```

Previous Button Text

```
Dlg.PreviousButtonText = "Prev"
Dim text as String = Dlg.PreviousButtonText
```

Next Button Text

```
Dlg.NextButtonText = "Next"
Dim text as String = Dlg.NextButtonText
```

Login Button Text

```
Dlg.LoginButtonText = "Login"
Dim text as String = Dlg.LoginButtonText
```

Select Button Text

```
Dlg.SelectButtonText = "Select"
```

Dim text as String = Dlg.SelectButtonText