

# ifMessageDialog

**This interface 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.

## Implemented By

- `roMessageDialog`

## Supported Methods

- `SetTitle(title as String) as Void`
- `SetText(text as String) as Void`
- `UpdateText(text as String) as Void`
- `AddStaticText(text as String) as Void`
- `AddButton(id as Integer, title as String) as Boolean`
- `AddLeftButton(id as Integer, title as String) as Boolean`
- `UpdateButton(id as Integer, title as String) as Boolean`
- `AddButtonSeparator() as Boolean`
- `AddRatingButton(id as Integer, userRating as Integer, aggregateRating as Integer, tip as String) as Boolean`
- `AddLeftRatingButton(id as Integer, userRating as Integer, aggregateRating as Integer, tip as String) as Boolean`
- `ShowBusyAnimation() as Void`
- `Show() as Boolean`
- `Close() as Void`
- `EnableOverlay(enable as Boolean) as Void`
- `SetMenuTopLeft(topLeft as Boolean) as Void`
- `SetFocusedMenuItem(item as Integer) as Boolean`
- `EnableBackButton(enableBackButton as Boolean) as Void`

## Description of Methods

### **SetTitle(title as String) as Void**

Set the title for the dialog to the specified string. The title appears in the top center of the dialog in bold text.

### **SetText(text as String) as Void**

Appends a paragraph with the specified text.

The dialog will automatically resize to accommodate the text, up to the limit of available display space.

### **UpdateText(text as String) as Void**

This function replaces the last paragraph with the specified text, or appends a paragraph if there is not a prior paragraph.

Note: if the last paragraph added was a static paragraph, UpdateText appends a new paragraph.

### **AddStaticText(text as String) as Void**

Appends a static paragraph with the specified text.

Note: this static text does not get changed by subsequent calls to SetText or UpdateText().

### **AddButton(id as Integer, title as String) as Boolean**

Adds a button to the screen identified by the title and ID provided.

The buttons are at the bottom of the dialog and appear in the order added.

When the button is pressed, the script will receive an event from the application indicating the ID of the button pressed.

### **AddLeftButton(id as Integer, title as String) as Boolean**

This function is similar to AddButton(), except that the button that gets added is left instead of right justified in the dialog box.

### **UpdateButton(id as Integer, title as String) as Boolean**

If a button with the specified id exists, sets the button title to the new title.

### **AddButtonSeparator() as Boolean**

Adds horizontal line separating buttons into sections on the dialog.

### **AddRatingButton(id as Integer, userRating as Integer, aggregateRating as Integer, tip as String) as Boolean**

Adds a star rating button to the dialog. The rating is specified as an integer 1-100 which indicates the number of stars (1 to 5) to be displayed, or 0 if unrated. Think of this as a percentage value <20% = 1 star. This button is displayed in the menu area of the screen and appears in the order added. The userRating specifies the value the user rated the title, while the aggregate Rating represents the total for all users. The userRating takes precedence and determines the color of the buttons if set. The aggregateRating may display half-stars. The button returns the userRating in the event data field.

The rating widget is only allowed on message dialogs that call SetMenuTopLeft(true).

### **AddLeftRatingButton(id as Integer, userRating as Integer, aggregateRating as Integer, tip as String) as Boolean**

This function adds a rating button similar to the rating buttons that can be inserted in roSpringboard screens. The button is left justified in the dialog.

### **ShowBusyAnimation() as Void**

Display a spinning busy animation to indicate work in progress. The animation will continue until the screen is closed.

**Show() as Boolean**

Display or refresh the screen after creation or state changes.

**Close() as Void**

Close the screen and delete the associated object. Useful for avoiding screen flicker when the display order of your screens does not resemble a stack.

**EnableOverlay(enable as Boolean) as Void**

With overlay enabled, the background screen is no longer dimmed.

**SetMenuTopLeft(topLeft as Boolean) as Void**

If true, set the format of the buttons to left and top justified. Otherwise default to bottom right justified. Note that if you call `AddRatingButton()`, you must make the buttons top left justified.

**SetFocusedMenuItem(item as Integer) as Boolean**

Set a button id to highlight. Default is the first button from the top. Return true if successful.

**EnableBackButton(enableBackButton as Boolean) as Void**

Must set `enableBackButton` to true in order to send the `isScreenClosed()` event when the remote control's back button is pressed.

By default the `MessageDialog` will not send an `isScreenClosed()` event so that scripts that did not expect this event will not break.