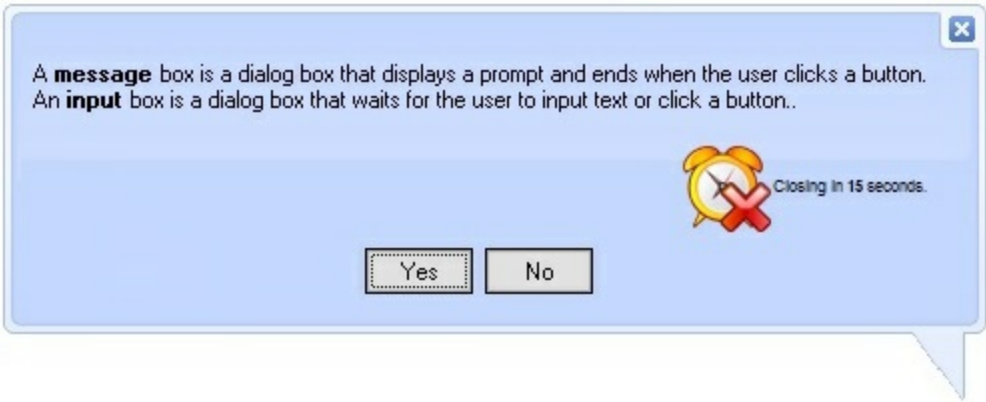




Exontrol's new ExSkinBox offers you skinnable message and input boxes. A message box is a dialog box that displays a prompt and ends when the user clicks a button. An input box is a dialog box that waits for the user to input text or click a button.

Features include:

- MsgBox and InputBox implementation.
- **AutoClose** support.
- Ability to center the message box relative to your window/dialog.
- Ability to apply custom **skins** to the message box.
- Ability to display custom **images** using the encoding **base64** format.
- Ability to load skins using the encoding base64 format.
- **HTML** format for message box's prompt, title or buttons as well.
- Ability to define the captions for standard buttons like Abort, Retry, Ignore and so on.
- Ability to define the font being used to display the message box.
- Ability to specify the position where the message box shows up.



Ž ExSkinBox is a trademark of Exontrol. All Rights Reserved.

## How to get support?

To keep your business applications running, you need support you can count on.

Here are few hints what to do when you're stuck on your programming:

- Check out the samples - they are here to provide some quick info on how things should be done
- Check out the how-to questions using the [eXHelper](#) tool
- Check out the help - includes documentation for each method, property or event
- Check out if you have the latest version, and if you don't have it send an update request [here](#).
- Submit your problem(question) [here](#).

Don't forget that you can contact our development team if you have ideas or requests for new components, by sending us an e-mail at [support@exontrol.com](mailto:support@exontrol.com) ( please include the name of the product in the subject, ex: exgrid ) . We're sure our team of developers will try to find a way to make you happy - and us too, since we helped.

Regards,  
Exontrol Development Team

<https://www.exontrol.com>





# constants OutResult

The OutResult type defines the buttons that a message box can handle. The [Out](#) method displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which button the user clicked.

Name	Value	Description
exCloseTimer	-1	The dialog is closed after elapsing time. The <a href="#">AutoClose</a> property specifies the number of seconds to let the message box being visible.
exClose	0	The dialog has been closed.
exOK	1	The OK button has been clicked.
exCancel	2	The Cancel button has been clicked.
exAbort	3	The Abort button has been clicked.
exRetry	4	The Retry button has been clicked.
exIgnore	5	The Ignore button has been clicked.
exYes	6	The Yes button has been clicked.
exNo	7	The No button has been clicked.

# constants OutStyle

The OutStyle enumeration values specifies the look and feel for your message box. Use the [Out](#) method to invoke a message box with a specified style. The first group of values (05) describes the number and type of buttons displayed in the message box; the second group (16, 32, 48, 64) describes the icon style; the third group (256, 512) determines which button is the default; the fourth group (4096) determines the modality of the message box, and the fifth group specifies whether or not the message box window is the foreground window, along with the alignment and direction of the text.

Name	Value	Description
exOKOnly	0	Displays OK button only.
exOKCancel	1	Displays OK and Cancel buttons.
exAbortRetryIgnore	2	Displays Abort, Retry, and Ignore buttons.
exYesNoCancel	3	Displays Yes, No, and Cancel buttons.
exYesNo	4	Displays Yes and No buttons.
exRetryCancel	5	Displays Retry and Cancel buttons.
exCritical	16	 Displays Critical Message icon.
exQuestion	32	 Displays Warning Query icon.
exExclamation	48	 Displays Warning Message icon.
exInformation	64	 Displays Information Message icon.
exDefaultButton2	256	Second button is default.
exDefaultButton3	512	Third button is default.
exSystemModal	4096	System is modal. All applications are suspended until the user responds to the message box.
exMsgBoxSetForeground	65536	Specifies the message box window as the foreground window.
exMsgBoxRight	524288	Text is right-aligned.
exMsgBoxRtlReading	1048576	Specifies text should appear as right-to-left reading on Hebrew and Arabic systems.

# MsgBox object

**Tip** The /COM object can be placed on a HTML page (with usage of the HTML object tag: <object classid="clsid:...">) using the class identifier: {6BBB0912-B723-4B47-A806-B529336D81B9}. The object's program identifier is: "Exontrol.MsgBox". The /COM object module is: "ExSkinBox.dll"

The MsgBox object provides message and input boxes. Use the [Load](#) method to assign a new skin for your message boxes. The [Out](#) method to displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which button the user clicked. The [Input](#) property displays a prompt in a dialog box, waits for the user to input text or click a button, and then returns a string containing the contents of the text box.

Name	Description
<a href="#">AutoClose</a>	Specifies the time in seconds to automatically close the message box.
<a href="#">Button</a>	Specifies the caption for the specified button.
<a href="#">EBNhWND</a>	Specifies the handle of the eXButton's control that controls the visual appearance for the message box's buttons.
<a href="#">FitTitle</a>	Specifies whether the MessageBox or InputBox ensures that its title/caption fits the dialog's title.
<a href="#">FitToScreen</a>	Specifies whether the MessageBox or InputBox ensures that it fits the screen ( current monitor ).
<a href="#">Font</a>	Retrieves or sets the control's font.
<a href="#">FormatAnchor</a>	Specifies the visual effect for anchor elements in HTML captions.
<a href="#">HTMLPicture</a>	Adds or replaces a picture in HTML captions.
<a href="#">Image</a>	Specifies the image being displayed for a specified style.
<a href="#">Images</a>	Sets at runtime the control's image list. The Handle should be a handle to an Images List Control.
<a href="#">ImageSize</a>	Retrieves or sets the size of icons the control displays..
<a href="#">Input</a>	Displays a prompt in a dialog box, waits for the user to input text or click a button, and then returns a string containing the contents of the text box.
<a href="#">Load</a>	Loads a skin file to be used.
<a href="#">MinHeight</a>	Specifies the minimum height of the message box.
<a href="#">MinWidth</a>	Specifies the minimum width of the message box.
<a href="#">Out</a>	Displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which

button the user clicked.

[Replacelcon](#)

Adds a new icon, replaces an icon or clears the control's image list.

[Version](#)

Retrieves the control's version.

# property MsgBox.AutoClose as Long

Specifies the time in seconds to automatically close the message box.

Type	Description
Long	A long expression that specifies the time in seconds to automatically close the message box.

By default, the AutoClose property is 0. If the AutoClose property is 0, the user must close the message box using the ui buttons, else the message box is automatically closed after specified time is elapsed. Use the AutoClose property to automatically let the message box being visible for a specified time. The [Out](#) method returns exCloseTimer value if the message box has been automatically closed after elapsing the specified time. The <%sec%> HTML tag indicates in the Prompt parameter of the Out method indicates the number of the seconds until the message box is closed. The <%sec%> is valid only if the AutoClose property is not 0. Use the <%sec%> HTML tag to indicate a prompt for number of seconds until the message box will be automatically closed.

The following VB sample displays a message box, and automatically closes it after 5 seconds:

```
With MsgBox1
    .AutoClose = 5
    .Out "This is a short message we want to display for few seconds<br> <r>Closing in
<b> <%sec%> </b> seconds.<br> <br>", vbInformation, "AutoClose"
End With
```

The message box displays the message "Closing in <%sec%> seconds" that indicates the number of seconds until the message box will be automatically closed.

# property MsgBox.Button(Button as OutResult) as String

Specifies the caption for the specified button.

Type	Description
Button as <a href="#">OutResult</a>	An OutResult expression that indicates the identifier of the button being changed.
String	A string expression that indicates the new button caption.

Use the Button property to assign a new caption for standard buttons. The Button has effect for the next call of the [Out](#) method. Use the [Image](#) property to assign new images for [exInformation](#), [exQuestion](#), [exCritical](#) or [exExclamation](#) predefined types.

The Button property accepts HTML format like follows:

- **<b> ... </b>** displays the text in **bold**
- **<i> ... </i>** displays the text in *italics*
- **<u> ... </u>** underlines the text
- **<s> ... </s>** Strike-through text
- **<a id;options> ... </a>** displays an [anchor](#) element that can be clicked. An anchor is a piece of text or some other object (for example an image) which marks the beginning and/or the end of a hypertext link. The <a> element is used to mark that piece of text (or inline image), and to give its hypertextual relationship to other documents. The control fires the *AnchorClick(AnchorID, Options)* event when the user clicks the anchor element. The *FormatAnchor* property customizes the visual effect for anchor elements.
- **<font face;size> ... </font>** displays portions of text with a different font and/or different size. For instance, the "**<font Tahoma;12>bit</font>**" draws the bit text using the Tahoma font, on size 12 pt. If the name of the font is missing, and instead size is present, the current font is used with a different size. For instance, "**<font ;12>bit</font>**" displays the bit text using the current font, but with a different size.
- **<fgcolor rrggbb> ... </fgcolor>** or **<fgcolor=rrggbb> ... </fgcolor>** displays text with a specified **foreground** color. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- **<bgcolor rrggbb> ... </bgcolor>** or **<bgcolor=rrggbb> ... </bgcolor>** displays text with a specified **background** color. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- **<solidline rrggbb> ... </solidline>** or **<solidline=rrggbb> ... </solidline>** draws a solid-line on the bottom side of the current text-line, of specified RGB color. The **<solidline> ... </solidline>** draws a black solid-line on the bottom side of the current text-line. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- **<dotline rrggbb> ... </dotline>** or **<dotline=rrggbb> ... </dotline>** draws a dot-line on the bottom side of the current text-line, of specified RGB color. The **<dotline> ...**



</dotline> draws a black dot-line on the bottom side of the current text-line. The rr/gg/bb represents the red/green/blue values of the color in hexa values.

- **<upline> ... </upline>** draws the line on the top side of the current text-line (requires <solidline> or <dotline>).
- **<r>** right aligns the text
- **<c>** centers the text
- **<br>** forces a line-break
- **<img>number[:width]</img>** inserts an icon inside the text. The number indicates the index of the icon being inserted. Use the Images method to assign a list of icons to your chart. The last 7 bits in the high significant byte of the number expression indicates the identifier of the skin being used to paint the object. Use the [Add](#) method to add new skins to the control. If you need to remove the skin appearance from a part of the control you need to reset the last 7 bits in the high significant byte of the color being applied to the part. The width is optional and indicates the width of the icon being inserted. Using the width option you can overwrite multiple icons getting a nice effect. By default, if the width field is missing, the width is 18 pixels.
- **<img>key[:width]</img>** inserts a custom size picture into the text being previously loaded using the HTMLPicture property. The Key parameter indicates the key of the picture being displayed. The Width parameter indicates a custom size, if you require to stretch the picture, else the original size of the picture is used.
- **&** glyph characters as **&amp;**; ( & ), **&lt;**; ( < ), **&gt;**; ( > ), **&qout;** ( " ) and **&#number;** ( the character with specified code ), For instance, the **&#8364;** displays the EUR character. The **&** ampersand is only recognized as markup when it is followed by a known letter or a #character and a digit. For instance if you want to display **<b>bold</b>** in HTML caption you can use **&lt;b&gt;bold&lt;/b&gt;**;
- **<off offset> ... </off>** defines the vertical offset to display the text/element. The offset parameter defines the offset to display the element. This tag is inheritable, so the offset is keep while the associated </off> tag is found. You can use the <off offset> HTML tag in combination with the <font face;size> to define a smaller or a larger font to be displayed. For instance: "Text with <font ;7><off 6>subscript" displays the text such as: Text with subscript The "Text with <font ;7><off -6>superscript" displays the text such as: Text with subscript
- **<gra rrggbb;mode;blend> ... </gra>** defines a gradient text. The text color or <fgcolor> defines the starting gradient color, while the rr/gg/bb represents the red/green/blue values of the ending color, 808080 if missing as gray. The mode is a value between 0 and 4, 1 if missing, and blend could be 0 or 1, 0 if missing. The <font> HTML tag can be used to define the height of the font. Any of the rrggbb, mode or blend field may not be specified. The <gra> with no fields, shows a vertical gradient color from the current text color to gray (808080). For instance the "<font ;18><gra FFFFFFFF;1;1>gradient-center</gra></font>" generates the following picture:

gradient-center

- **<out rrggbb;width> ... </out>** shows the text with outlined characters, where rr/gg/bb represents the red/green/blue values of the outline color, 808080 if missing as gray, width indicates the size of the outline, 1 if missing. The text color or <fgcolor> defines the color to show the inside text. The <font> HTML tag can be used to define the height of the font. For instance the "<font ;31><out 000000><fgcolor=FFFFFF>outlined</fgcolor></out></font>" generates the following picture:

outlined

- **<sha rrggbb;width;offset> ... </sha>** define a text with a shadow, where rr/gg/bb represents the red/green/blue values of the shadow color, 808080 if missing as gray, width indicates the size of shadow, 4 if missing, and offset indicates the offset from the origin to display the text's shadow, 2 if missing. The text color or <fgcolor> defines the color to show the inside text. The <font> HTML tag can be used to define the height of the font. For instance the "<font ;31><sha>shadow</sha></font>" generates the following picture:

shadow

or "<font ;31><sha 404040;5;0><fgcolor=FFFFFF>outline anti-aliasing</fgcolor></sha></font>" gets:

outline anti-aliasing

The following sample changes the color for Abort button:

With MsgBox1

```
.Button(exAbort) = "<fgcolor=FF0000>Abort</fgcolor>"
```

```
.Out "This is the message box's prompt", OutStyle.exAbortRetryIgnore
```

End With

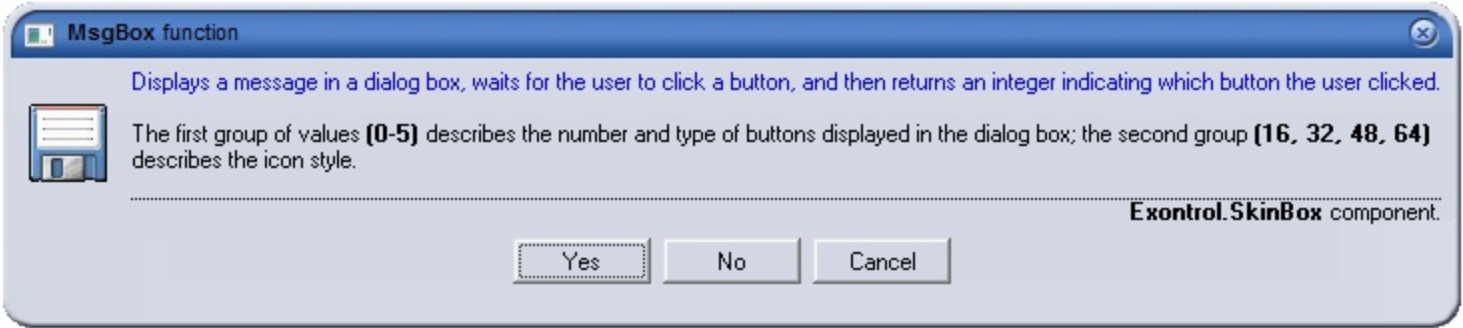
# property MsgBox.EBNhWND as Long

Specifies the handle of the eXButton's control that controls the visual appearance for the message box's buttons.

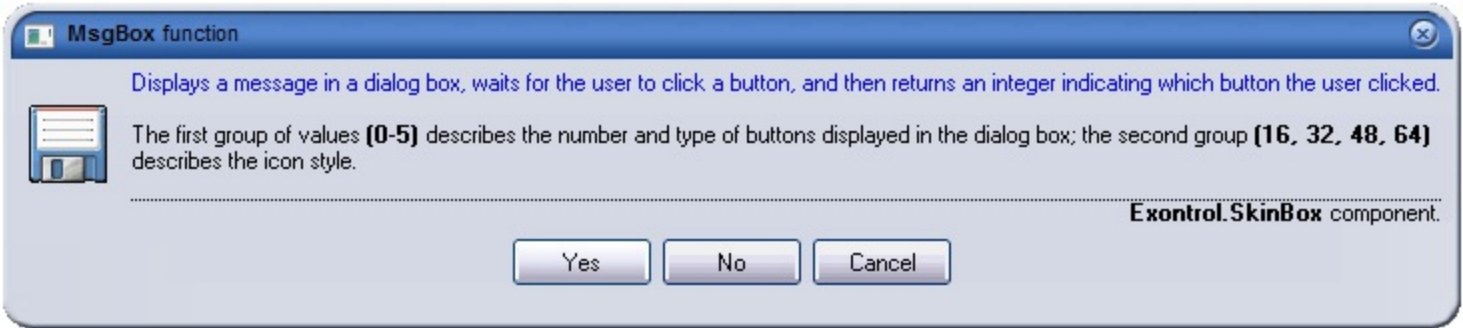
Type	Description
Long	A long expression that specifies the handle of the eXButton control that display the buttons of the message box in different states.

By default, The EBNhWND property is 0, which means that the buttons inside the message box are shown using the default background. The [hWnd](#) property of the eXButton specifies the handle of the eXButton needed to be passed to the EBNhWND property, so the look or the visual appearance of the buttons inside the eXSkinBox are handled by specified button. Changing the visual appearance for the eXButton makes the eXSkinBox's buttons visual appearance to be changed too.

The following screen shot shows the eXSkinBox control when the EBNhWND property is 0 ( by default ):



The following screen shot shows the eXSkinBox control when the EBNhWND property is not 0 ( *MsgBox1.EBNhWND = Button1.hWnd* ):



# property MsgBox.FitTitle as Boolean

Specifies whether the MessageBox or InputBox ensures that its title/caption fits the dialog's title.

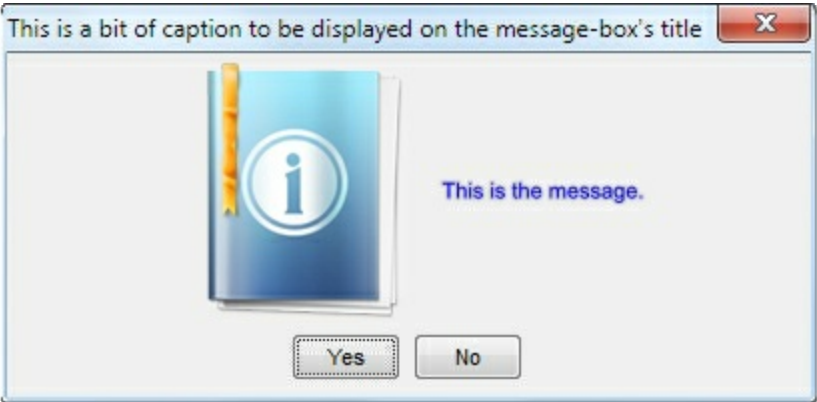
Type	Description
Boolean	A Boolean expression that specifies whether the MessageBox or InputBox ensures that its title/caption fits the dialog's title.

By default, the FitTitle property is False. Change the FitTitle property on True, to ensure that the dialog's title fit entirely the window. The Title parameter of the [Out/Input](#) method indicates the caption to be displayed on the dialog's title.

The following screen shot shows the message-box with the FitTitle property on False:



The following screen shot shows the message-box with the FitTitle property on True:



# property MsgBox.FitToScreen as Boolean

Specifies whether the MessageBox or InputBox ensures that it fits the screen ( current monitor ).

Type	Description
Boolean	A Boolean expression that specifies whether the MessageBox or InputBox ensures that it fits the screen ( current monitor ).

By default, the FitToScreen property is True, which means that the MessageBox or InputBox always ensures that it fits the screen or the current monitor. The X and Y parameters of the [Out](#) or [Input](#) method specifies the position of the MessageBox / InputBox dialogs. For instance, if the FitToScreen property is False, the X or Y parameters could point anywhere on the screen, else the MessageBox/InputBox will always fit the screen or the current monitor.

# property MsgBox.Font as IFontDisp

Retrieves or sets the control's font.

Type	Description
IFontDisp	A Font object being used to display the message box as well the input box.

By default, the Font property is Arial, 8. The [Out](#) method to displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which button the user clicked. The [Input](#) property displays a prompt in a dialog box, waits for the user to input text or click a button, and then returns a string containing the contents of the text box.

# property MsgBox.FormatAnchor(New as Boolean) as String

Specifies the visual effect for anchor elements in HTML captions.

Type	Description
New as Boolean	A Boolean expression that indicates whether to specify the anchors never clicked or anchors being clicked.
String	A String expression that indicates the HTML format to apply to anchor elements.

By default, the FormatAnchor(**True**) property is "<u><fgcolor=0000FF>#" that indicates that the anchor elements ( that were never clicked ) are underlined and shown in light blue. Also, the FormatAnchor(**False**) property is "<u><fgcolor=000080>#" that indicates that the anchor elements are underlined and shown in dark blue. The visual effect is applied to the anchor elements, if the FormatAnchor property is not empty. For instance, if you want to do not show with a new effect the clicked anchor elements, you can use the FormatAnchor(**False**) = "", that means that the clicked or not-clicked anchors are shown with the same effect that's specified by FormatAnchor(**True**). An anchor is a piece of text or some other object (for example an image) which marks the beginning and/or the end of a hypertext link. The <a> element is used to mark that piece of text (or inline image), and to give its hypertextual relationship to other documents. The control fires the [AnchorClick](#) event to notify that the user clicks an anchor element. This event is fired only if prior clicking the control it shows the hand cursor. The AnchorClick event carries the identifier of the anchor, as well as application options that you can specify in the anchor element. The hand cursor is shown when the user hovers the mouse on the anchor elements.

# property MsgBox.HTMLPicture(Key as String) as Variant

Adds or replaces a picture in HTML captions.

Type	Description
Key as String	A String expression that indicates the key of the picture being added or replaced. If the Key property is Empty string, the entire collection of pictures is cleared.
Variant	<p>The HTMLPicture specifies the picture being associated to a key. It can be one of the followings:</p> <ul style="list-style-type: none"><li>• a string expression that indicates the path to the picture file, being loaded.</li><li>• a string expression that indicates the base64 encoded string that holds a picture object, Use the <a href="#">eximages</a> tool to save your picture as base64 encoded format.</li><li>• A Picture object that indicates the picture being added or replaced. ( A Picture object implements IPicture interface ),</li></ul> <p>If empty, the picture being associated to a key is removed. If the key already exists the new picture is replaced. If the key is not empty, and it doesn't not exist a new picture is added.</p>

The HTMLPicture property handles a collection of custom size picture being displayed in the HTML captions, using the <img> tags. By default, the HTMLPicture collection is empty. Use the HTMLPicture property to add new pictures to be used in HTML captions. For instance, the HTMLPicture("pic1") = "c:\winnt\zapotec.bmp", loads the zapotec picture and associates the pic1 key to it. Any "<img>pic1</img>" sequence in HTML captions, displays the pic1 picture. On return, the HTMLPicture property retrieves a Picture object ( this implements the IPictureDisp interface ).

The following sample shows how to put a custom size picture in the column's header:

```
<CONTROL>.HTMLPicture("pic1") = "c:/temp/editors.gif"  
<CONTROL>.HTMLPicture("pic2") = "c:/temp/editpaste.gif"
```







# property MsgBox.Image(Style as OutStyle) as Variant

Specifies the image being displayed for a specified style.

Type	Description
Style as <a href="#">OutStyle</a>	A OutStyle expression that indicates the predefined image being changed. Only exCritical, exQuestion, exExclamation or exInformation are accepted.
Variant	A Picture object that indicates the image's picture. ( A Picture object implements IPicture interface ), a string expression that indicates the path to the image file being loaded, a string expression that indicates the base64 encoded string that holds a picture object. Use the <a href="#">eximages</a> tool to save your picture as base64 encoded format.

Use the Image property to assign a new image to exInformation, exQuestion, exCritical or exExclamation predefined styles. Use the [Button](#) property to assign new captions for predefined buttons.

By default, the following images are loaded:

-  exCritical
-  exQuestion
-  exExclamation
-  exInformation

The following sample changes the default image for exInformation style:

```
With MsgBox1

    Dim s As String
    s =
"gbHJJGHA5MKAAEie4AAAFh0ODYFCA7AoILYFGBIjBwAo7QoFIqbApQUoFLa7kzFApIIrWlrC

    s = s +
"oIGGghm0YgcimNJPIOPpACUlocEodBoIWGBJkEFh6GoCJrCYEIWgSZQhisZwqV8TAGHSaBC

    s = s +
```

"AaRUjIEEAQCYeAgBDA2G0EAoh2CyDuLQGIBQGgTCINsAICwsAMCkGsLwSQchSHSLYbYvh1

s = s +

"oBY8BLAQFYAoBwYQEDBAUEMDQBhwibH0DkeAWwQDsEqAgI4KAhA3AYCoLALBIDAAoBA

s = s +

"qG8PY8x0jnEkBlagfQNBpDgFMFI6goinHUKIUg9gUBLBNguA1AkG0BcghAMggBVJJAIsRml

s = s +

"FARh3AABWgEoJB/gEBrB5BOArAEBFgTBjgahbgjgeheBRAqBpBhBBglhaBggWhFhigiAQOA

s = s +

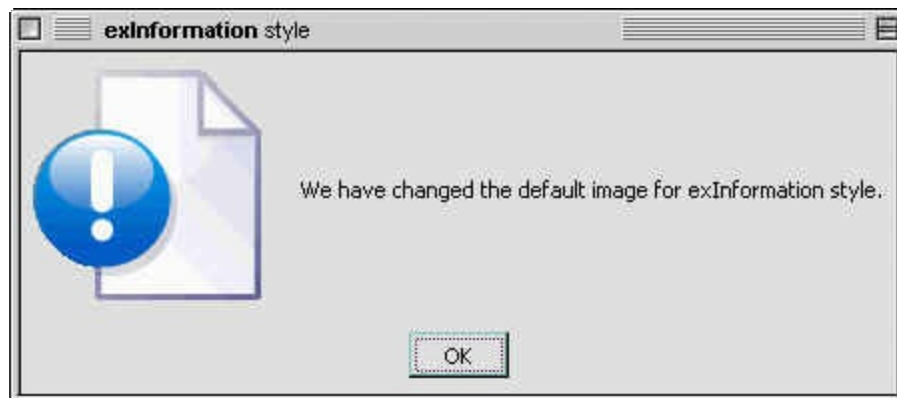
"6B+j/h/gTBEBiA0h7g5BxS6gAB1BAgJgTAnh8hlArgfATBLBnh+ByA9hgAmg6AhAAhOAPhm

s = s +

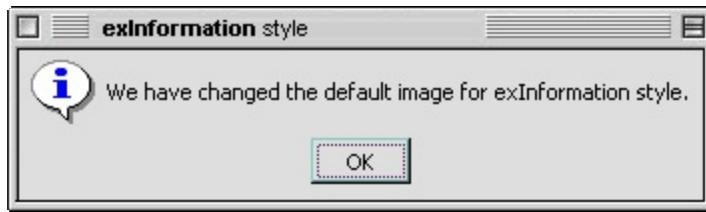
"sAzgYAsA6ATAEgdhKAshLBcArg+hKhEhggBgdhPB8BEg1gZAwhrgxBDBOh6gsBDgQhFBwB

.Image(exInformation) = s

.Out "We have changed the default image for exInformation style.",  
OutStyle.exInformation, "**exInformation** style"  
End With



The following screen shot shows the same message without changing the default image of the exInformation style:



# method MsgBox.Images (Handle as Variant)

Sets the control's images list at runtime. The Handle should be a handle to an Image List control.

Type	Description
------	-------------

The Handle parameter can be:

- A string expression that specifies the ICO file to add. The ICO file format is an image file format for computer icons in Microsoft Windows. ICO files contain one or more small images at multiple sizes and color depths, such that they may be scaled appropriately. For instance, Images("c:\temp\copy.ico") method adds the sync.ico file to the control's Images collection (*string, loads the icon using its path*)
- A string expression that indicates the BASE64 encoded string that holds the icons list. Use the Exontrol's [ExImages](#) tool to save/load your icons as BASE64 encoded format. In this case the string may begin with "gBJJ..." (*string, loads icons using base64 encoded string*)
- A reference to a Microsoft ImageList control (mscomctl.ocx, MSComctlLib.ImageList type) that holds the icons to add (*object, loads icons from a Microsoft ImageList control*)
- A reference to a Picture (IPictureDisp implementation) that holds the icon to add. For instance, the VB's LoadPicture (Function LoadPicture([FileName], [Size], [ColorDepth], [X], [Y]) As IPictureDisp) or LoadResPicture (Function LoadResPicture(id, restype As Integer) As IPictureDisp) returns a picture object (*object, loads icon from a Picture object*)
- A long expression that identifies a handle to an Image List Control ( the Handle should be of HIMAGELIST type ). On 64-bit platforms, the Handle parameter must be a Variant of LongLong / LONG\_PTR data type ( signed 64-bit (8-byte) integers ), saved under lVal field, as VT\_I8 type. The LONGLONG / LONG\_PTR is \_\_int64, a 64-bit integer. For instance, in C++ you can use as Images( COleVariant( (LONG\_PTR)hImageList) ) or Images( COleVariant(

Handle as Variant

(LONGLONG)hImageList) ), where hImageList is of HIMAGELIST type. The GetSafeHandle() method of the CImageList gets the HIMAGELIST handle (long, loads icon from HIMAGELIST type)

---

The control provides an image list window, that's displayed at design time. The [ImageSize](#) property defines the size (width/height) of the icons within the control's Images collection. At runtime, the user can use the Images and [Replacelcon](#) method to change the Images collection. The Images collection is 1 based. The control's context list may contain icons.

# property MsgBox.ImageSize as Long

Specifies the size of the control' icons.

Type	Description
Long	A long expression that defines the size of icons the control displays.

By default, the ImageSize property is 16 (pixels). The ImageSize property specifies the size of icons being loaded using the [Images](#) method. The control's Images collection is cleared if the ImageSize property is changed, so it is recommended to set the ImageSize property before calling the Images method. The ImageSize property defines the size (width/height) of the icons within the control's Images collection. For instance, if the ICO file to load includes different types the one closest with the size specified by ImageSize property is loaded by Images method. The ImageSize property does NOT change the height for the control's font.

## method MsgBox.Input (Prompt as String, [Title as Variant], [DefaultResponse as Variant], [Image as Variant], [X as Variant], [Y as Variant])

Displays a prompt in a dialog box, waits for the user to input text or click a button, and then returns a string containing the contents of the text box.

Type	Description
Prompt as String	A String expression displayed as the message in the dialog box. The Prompt parameter may includes HTML format like explained bellow.
Title as Variant	A String expression displayed in the title bar of the dialog box. The Title parameter may includes HTML format like explained bellow. The <a href="#">FitTitle</a> property specifies whether the MessageBox or InputBox ensures that its title/caption fits the dialog's title
DefaultResponse as Variant	A string expression displayed in the text box as the default response if no other input is provided. If you omit DefaultResponse, the displayed text box is empty.
Image as Variant	A string expression that indicates the path to a picture file being displayed, or a string expression that stores a picture in encoded base64 format. Use the <a href="#">eximages</a> tool to save your picture as base64 encoded format.
X as Variant	<p>The X parameter could be a numeric or a string expression like follows:</p> <ul style="list-style-type: none"><li>• If missing/omit, the dialog box is horizontally centered relative to active screen/monitor.</li><li>• If -1, the dialog is shown at the current cursor position ( x-coordinate ).</li><li>• A Numeric expression that specifies, in pixels, the distance of the left edge of the dialog box from the left edge of the screen.</li><li>• A String expression that indicates the handle of the window to center relative to. You need to pass the handle of the window ( hWnd for /COM or Handle or hWnd for /NET ) as a string.</li></ul> <p>The <a href="#">FitToScreen</a> property specifies whether the InputBox ensures that it fits the screen ( current monitor ).</p>

The Y parameter could be a numeric or a string expression like follows:

- If missing/omit, the dialog box is vertically centered relative to active screen/monitor.
- If -1, the dialog is shown at the current cursor position ( y-coordinate ).
- A Numeric expression that specifies, in pixels, the distance of the upper edge of the dialog box from the top of the screen.
- A String expression that indicates the handle of the window to center relative to. You need to pass the handle of the window ( hWnd for /COM or Handle or hWnd for /NET ) as a string.

Y as Variant

The [FitToScreen](#) property specifies whether the InputBox ensures that it fits the screen ( current monitor ).

## Return

## Description

String

A string expresion that indicates the string entered by the user. If the user select cancel the Input method retrieves the DefaultResponse value.

Use the Input method to let user enters text using a dialog box and two buttons OK and Cancel. Use the [Button](#) property to assign a new caption for exOK or exCancel buttons.

The following samples aligns the input box as follow ( using /COM version ):

- MsgBox1.Input "Your message", displays the input box centered in the active screen.
- MsgBox1.Input "Your message", , , , 0, displays the input box on top, horizontally centered in the active screen.
- MsgBox1.Input "Your message", , , , **Str(Me.hWnd), Str(Me.hWnd)**, displays the input box centered relative to the parent form.
- MsgBox1.Input "Your message", , , , 0, **Str(Me.hWnd)**, displays the input box on the left of the active screen, vertically centered relative to the form.
- MsgBox1.Input "Your message", , , , 100, 100, displays the input box at specified position
- MsgBox1.Input "Your message", , , , -1, -1, displays the input box at cursor position

The following samples aligns the input box as follow ( using /NET version ):

- MsgBox1.Input("Your message"), displays the input box centered in the active screen.
- MsgBox1.Input("Your message", Nothing, Nothing, Nothing, **Nothing, 0**), displays the



input box on top, horizontally centered in the active screen.

- `MsgBox1.Input("Your message", Nothing, Nothing, Nothing, Me.Handle.ToString(), Me.Handle.ToString())`, displays the input box centered relative to the parent form.
- `MsgBox1.Input("Your message", Nothing, Nothing, Nothing, 0, Me.Handle.ToString())`, displays the input box on the left of the active screen, vertically centered relative to the form.
- `MsgBox1.Input("Your message", Nothing, Nothing, Nothing, 100, 100)`, displays the input box at specified position
- `MsgBox1.Input("Your message", Nothing, Nothing, Nothing, -1, -1)`, displays the input box at cursor position

The Prompt and Title arguments support the following HTML elements:

- `<b> ... </b>` displays the text in **bold**
- `<i> ... </i>` displays the text in *italics*
- `<u> ... </u>` underlines the text
- `<s> ... </s>` ~~Strike-through~~ text
- `<a id;options> ... </a>` displays an [anchor](#) element that can be clicked. An anchor is a piece of text or some other object (for example an image) which marks the beginning and/or the end of a hypertext link. The `<a>` element is used to mark that piece of text (or inline image), and to give its hypertextual relationship to other documents. The control fires the *AnchorClick(AnchorID, Options)* event when the user clicks the anchor element. The *FormatAnchor* property customizes the visual effect for anchor elements.
- `<font face;size> ... </font>` displays portions of text with a different font and/or different size. For instance, the "`<font Tahoma;12>bit</font>`" draws the bit text using the Tahoma font, on size 12 pt. If the name of the font is missing, and instead size is present, the current font is used with a different size. For instance, "`<font ;12>bit</font>`" displays the bit text using the current font, but with a different size.
- `<fgcolor rrggbb> ... </fgcolor>` or `<fgcolor=rrggb> ... </fgcolor>` displays text with a specified **foreground** color. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- `<bgcolor rrggbb> ... </bgcolor>` or `<bgcolor=rrggb> ... </bgcolor>` displays text with a specified **background** color. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- `<solidline rrggbb> ... </solidline>` or `<solidline=rrggb> ... </solidline>` draws a solid-line on the bottom side of the current text-line, of specified RGB color. The `<solidline> ... </solidline>` draws a black solid-line on the bottom side of the current text-line. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- `<dotline rrggbb> ... </dotline>` or `<dotline=rrggb> ... </dotline>` draws a dot-line on the bottom side of the current text-line, of specified RGB color. The `<dotline> ... </dotline>` draws a black dot-line on the bottom side of the current text-line. The rr/gg/bb represents the red/green/blue values of the color in hexa values.

- **<upline> ... </upline>** draws the line on the top side of the current text-line (requires **<solidline>** or **<dotline>**).
- **<r>** right aligns the text
- **<c>** centers the text
- **<br>** forces a line-break
- **<img>number[:width]</img>** inserts an icon inside the text. The number indicates the index of the icon being inserted. Use the Images method to assign a list of icons to your chart. The last 7 bits in the high significant byte of the number expression indicates the identifier of the skin being used to paint the object. Use the [Add](#) method to add new skins to the control. If you need to remove the skin appearance from a part of the control you need to reset the last 7 bits in the high significant byte of the color being applied to the part. The width is optional and indicates the width of the icon being inserted. Using the width option you can overwrite multiple icons getting a nice effect. By default, if the width field is missing, the width is 18 pixels.
- **<img>key[:width]</img>** inserts a custom size picture into the text being previously loaded using the HTMLPicture property. The Key parameter indicates the key of the picture being displayed. The Width parameter indicates a custom size, if you require to stretch the picture, else the original size of the picture is used.
- **&** glyph characters as **&amp;**; ( & ), **&lt;**; ( < ), **&gt;**; ( > ), **&qout;** ( " ) and **&#number;** ( the character with specified code ), For instance, the **&#8364;** displays the EUR character. The **&** ampersand is only recognized as markup when it is followed by a known letter or a #character and a digit. For instance if you want to display **<b>bold</b>** in HTML caption you can use **&lt;b&gt;bold&lt;/b&gt;**;
- **<off offset> ... </off>** defines the vertical offset to display the text/element. The offset parameter defines the offset to display the element. This tag is inheritable, so the offset is keep while the associated **</off>** tag is found. You can use the **<off offset>** HTML tag in combination with the **<font face;size>** to define a smaller or a larger font to be displayed. For instance: "Text with **<font ;7><off 6>subscript**" displays the text such as: Text with subscript The "Text with **<font ;7><off -6>superscript**" displays the text such as: Text with superscript
- **<gra rrggbb;mode;blend> ... </gra>** defines a gradient text. The text color or **<fgcolor>** defines the starting gradient color, while the rr/gg/bb represents the red/green/blue values of the ending color, 808080 if missing as gray. The mode is a value between 0 and 4, 1 if missing, and blend could be 0 or 1, 0 if missing. The **<font>** HTML tag can be used to define the height of the font. Any of the rrggbb, mode or blend field may not be specified. The **<gra>** with no fields, shows a vertical gradient color from the current text color to gray (808080). For instance the "**<font ;18><gra FFFFFFFF;1;1>gradient-center</gra></font>**" generates the following picture:

gradient-center

- **<out rrggbb;width> ... </out>** shows the text with outlined characters, where rr/gg/bb represents the red/green/blue values of the outline color, 808080 if missing as gray,

width indicates the size of the outline, 1 if missing. The text color or <fgcolor> defines the color to show the inside text. The <font> HTML tag can be used to define the height of the font. For instance the "<font ;31><out 000000>

<fgcolor=FFFFFF>outlined</fgcolor></out></font>" generates the following picture:

outlined

- <sha rrggbb;width;offset> ... </sha> define a text with a shadow, where rr/gg/bb represents the red/green/blue values of the shadow color, 808080 if missing as gray, width indicates the size of shadow, 4 if missing, and offset indicates the offset from the origin to display the text's shadow, 2 if missing. The text color or <fgcolor> defines the color to show the inside text. The <font> HTML tag can be used to define the height of the font. For instance the "<font ;31><sha>shadow</sha></font>" generates the following picture:

shadow

or "<font ;31><sha 404040;5;0><fgcolor=FFFFFF>outline anti-aliasing</fgcolor></sha></font>" gets:

outline anti-aliasing

The following sample asks user to enter his name:

With MsgBox1

Dim s As String

s =

"gBHJJGHA5MloAEle4AAAFaoDCAPBQIEgYGAaBUXCIwiowF8bF4fHcSHYaCI7HlmlAsHZEfh

s = s +

"DgmYAYg4OBOgqYABA0JxtF0LpBGIBBIhEI5NCQEA3DmNQ5GWPQUnEBAIXcA5DIWWYdj8

Debug.Print .Input(" Please enter your **name** here: ", "Input **Name**", , s)

End With



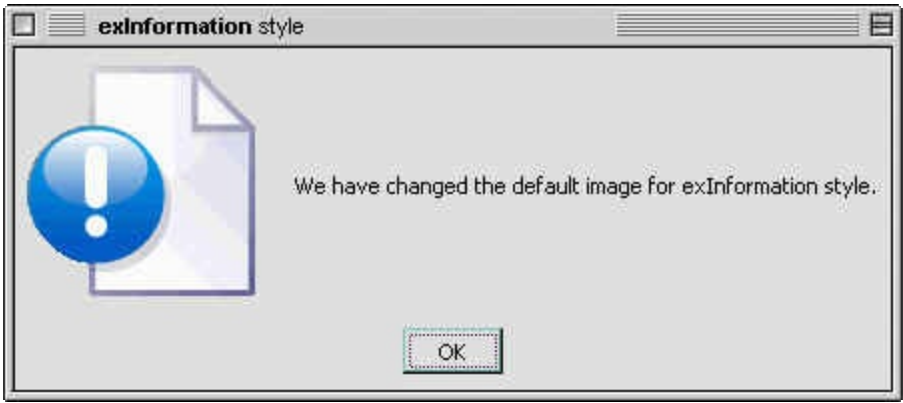
# method MsgBox.Load (SkinFile as String)

Loads a skin file to be used.

Type	Description
SkinFile as String	A string expression that indicates the path to the skin file ( *.esk files ), or a string expression that holds a skin file in encoded base64 format. Use the <a href="#">eximages</a> tool to save your picture as base64 encoded format.

The skin file must be created using the skin builder that is provided by the Exontrol's [exSkin](#) component. Once that a skin is loaded, all the message boxes or input boxes will use the same skin, so you don't have to call the Load method each time when [Out](#) or [Input](#) method is called. Use the [Image](#), [Button](#) property to assign new images/caption for predefined styles/buttons.

By default, the message box skin looks like follows:



The following sample loads a new skin for the message box:

```
With MsgBox1

  Dim s As String
  s =
"gbHJJGHA5MKAAEie4AAAFh0ODYFCA7AoILYFGBIjBwAo7QoFIqbaPQUoFLa7kzFApIIrWlrC

  s = s +
"oIGGghm0YgcimNJPIOPpACUlocEodBoIWGBJkEFh6GoCJrCYEIWgSZQhisZwqV8TAGHSaBC

  s = s +
"AaRUjIEEAQCYeAgBDA2G0EAoh2CyDuLQGIBQGgTCINsAICwsAMCkGsLwSQchSHSLYbYvh1
```

s = s +

"oBY8BLAQFYAoBwYQEDBAUEMDQBhwibH0DkeAWwQDsEqAgI4KAhA3AYCoLALBIDAAoB/

s = s +

"qG8PY8x0jnEkBlagfQNBpDgFMFI6goinHUKIUg9gUBLBNguA1AkG0BcghAMggBVJJAlsRml

s = s +

"FARh3AABWgEoJB/gEBrB5BOArAEBFgTBjgahbgjgeheBRAqBpBhBBglhaBggWhFhigiAQOA

s = s +

"6B+j/h/gTBEBiA0h7g5BxS6gAB1BAgJgTAnh8hlArgfATBLBnh+ByA9hgAmg6AhAAhOAPhm

s = s +

"sAzgYAsA6ATAEgdhKAshLBcArg+hKhEhggBgdhPB8BEg1gZAwhrgxBDBoh6gsBDgQhFBwB

.Image(exInformation) = s

Dim sSkin As String

sSkin =

"gBFLBTJYCAEHhEJAABhABU4Yg6AABACAxWgKBADQKAAyDQKkEQGGQbRjhEAlGglg4JYjjl

sSkin = sSkin +

"WCqBIhiiCYOgqHxAakGQVHAEhECQbRCEGWWhVgCGZZkCYZaGUVg4hISwHCoShWBwa4lG

sSkin = sSkin +

"AeA+BiBqPAXooBzAgCGIANYDgwiIBcD0EAlAKkGBwAgQAzAKiGBgK4CYgxMBtAaG8BQDc

sSkin = sSkin +

"lVgUwkDxD1hYQAYxPDTHEE4J4uqNhnHYP8SY1AHCBCiAoDYUxUB9CeKoKoRsyAKE6EQQ

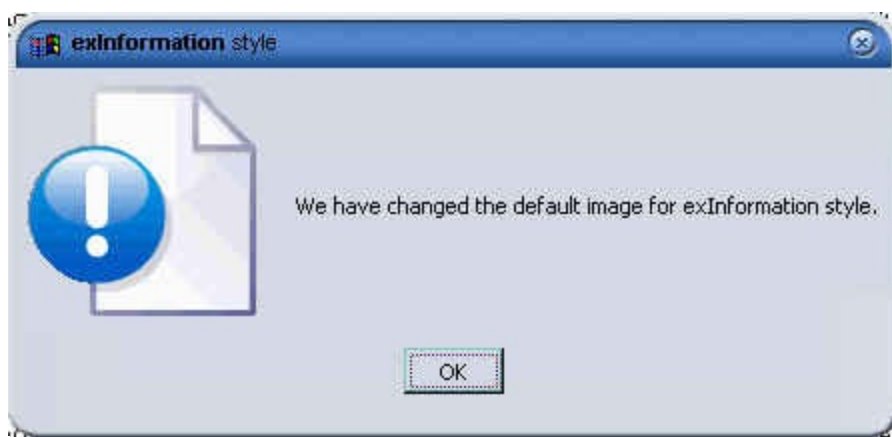
sSkin = sSkin +

"KKAVAcBDU4AwFgVYFgmgMF+lWgGRRRei/HylsGYZhOgOuLhDA1BpArgPgPhwgGBugrg2B

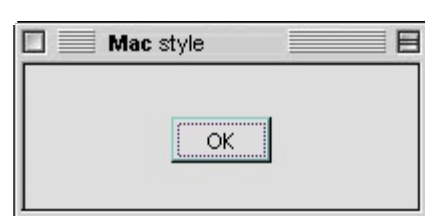
sSkin = sSkin +

"8gHgDhyheBThLyGAnhmB7gXgchwBYg/hQB6gvBHhqh4BPglhXg8hDjaB/BbhtA9Bng2hvg9

```
sSkin = sSkin +  
"2gLBYg4k3A6gLB4g7gLgYg8gLh4g+khg9gMA4hAgMBlhCgKB4gvgKhIhJgMglgqgMhInB1  
  
sSkin = sSkin +  
"eAMBgg3ANhghFALBQhXALgggmYrB3AFAJ3jhZgJ4Zg4gYgZg6gbhJg+gcAJhCgcBJhGgcg.  
  
sSkin = sSkin +  
"th6higuBahkguB6hjgugahmgugqhogug6huguhahwguAAgw1zrFgQgvViBKAKQBwAEhgg  
  
sSkin = sSkin +  
"ATxgghCAKBAGIAFgQA7hAhwA7APBQhEAKBsAUe9gShBAghnALgQgeg/hLh+ADg8AAhA  
  
sSkin = sSkin +  
"Y4RuYMoSZLoaV9JcqShl4FQaHUpyaH0qAalUqCaEoGxhGwEzqMAvBoFoEjgPAcCAAkzhk  
  
sSkin = sSkin +  
"CQAh/CkNwXlzxACZGeN0Al0AugJGgN0Cl0QuG YA2D0Do0gNg1GkHuDY3QYjTC6DkaY3Q  
  
sSkin = sSkin +  
"KTFhUFuTUEALgExvgqAuAAdgCQgiEAel4Z47BAiLHaB8Xo+w/PEB+L8fgXxgj8D+8QX40Q  
  
sSkin = sSkin +  
"AwZA9QwiZAcMkSgQgeAPDGOkAQPRIH0AcEESIJA8gaCSJEMguQRBMEkKQbIQgniUFIOk  
  
sSkin = sSkin +  
"cDkixDDCUd9A4AuAHCDDhC4BeAjCEDiC4B+AvCJhgD+A7CRDjB5A+BDCUDkC5BeBjCXD  
  
.Load sSkin  
  
.Out "We have changed the default image for exInformation style.",  
OutStyle.exInformation, "exInformation style"  
End With
```



The following list contains available skins on exSkin component ( **You are free to use them, and to create new skins using the exSkin's WYSWYG builder** ). In order to use a skin, you have two options. Passing a path to the file name that holds the skin, or using the s string defined on the right side of your selected skin.



Dim s As String

s =

"gBFLBTJYCAEHhEJAABhABBYpg6AABACAxWgKBADQKAAXjILEEQGGQbRjhGIwABOCYGRRE

s = s +

"BAMXZAAGZQJhmJhIFKGoKiaWQxn2FxxhikfQUiCY4aEgAAIFCF5RAMFJeCoah8ECX5KHYcLI

s = s +

"oXx6AAD+PAAwAggA4AaAAUARACiBBgB4AoQBsAZAGOALgERAjQCUAgIIOWgFDAMQ

s = s +

"gUw1BzCeCmLoPoThVAFB6KgGg/sFBaoSOkOoIBPjDAwFcBYWsfqDaE8VQVQjZIAUK0Kg6l

s = s +

"YGcLgoxNByAMCQZg1w5gyDOlwbHzRTBRGSDYOI4BHA4hHAZhaAbBEBoB9ATgjA0BegOh

S = S +  
"ih4B3gShygRAzhPh5B/BHhmh4g8DegegjhFh6gvALhnglAfgWhqAeBnhMB7AgBXhpB4gHq

S = S +  
"wBdAGA4gwV2gogKBo1FB4gvgLAYgzgLg4g617gRnGhAgLBYg5gLAog9gMBlhGgLhIhHgM

S = S +  
"8gbh5g7gcAZg+gcAphAgcA5hGgcBZhlgcBJhKgcB5hKAPgQhNAMhphOgdAJhPg dAZhUg

S = S +  
"BgAh3ANBwhSAPa17bh+gobMbzBQAPgAhIbXGlgwhQA OgqgCgsgg74Aawrhw5Bha7LBA

S = S +  
"hBfb4hE+xAUgthw4tB5g1cnBlbzB/AMhcAic1hBhCbz7Yg0ALBwhoAJAggYAOHigJA8AeAM

S = S +  
"IBgUGwgAfPkikOHYFw6LkqjWLRDh4BcYT+BYmigBgQSCB4WiKBwAQIIZTSdFcBxbLUIAVAE

S = S +  
"ISniAA2OAewEAADceqPsVwtQOhkBSEcQYwRJAdEAN4B4AQJPVEsHMaw8IFAgGQB0Sz0hP

S = S +  
"cKLQtCulGOqRMpzjAOHCMYCY4VHDBC7aMfwEACByCKP4ZgYxfjZCYEc fQlQoB9AUIULoKf

S = S +  
"DEBgJ5bg5AMJEWg/waDiEGCAYXPQpgUDoEcBgoRbBIbmEwS47QKAwAWPgZgAL1CUEIl4

S = S +  
"QABDpDHAIBIA4CWjKDBB9BRAAAPDRC4BpA8AFDRBmCjAAArCDD8C2AqexDECIC7BPDx

S = S +  
"OAZAhaBAKBYB0AOBuDBDYAPBwDOAYCHA5AqmEDRDEAKCxBEDpCKBYBrANDyCgDWCl

S = S +  
"zCiAmBHB YCLAQCSCcBHAIjqBDkBW BaApASADBYAyCGA1BMBNCEDYCcsJBmBcuMAiAlBVl



S = S +  
"0DkDze2AoAlk/KJBDtwAPD6B8BuApAiDvBVDqB3ACCMRZA8AgDJB5AmDCAMBEVMC0A\

S = S +  
"UAHADA8AcAPDMAyAJB0AzCDA4Ca5xAyDBC0Q1CoB6BO22CCCMBBi4R5CkAZBBDJC+D

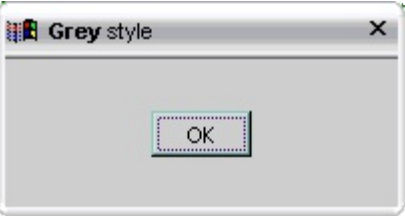
S = S +  
"iU/BcLlgZEZIC5HFgWOVuilrMYTMMeDxhMCst4pOVNF9SD0EIN6C8qL4moJot1BLAyJ98ilh

S = S +  
"p/iUMwbmSLivnUMhhHkL4DCSNBTF8Vgvn8RgJBANIBhQRQGGkEz+CCU47CYIoCGaIUoCM

S = S +  
"ldAdgbG8EIFw6kGD2F5EsfoaxPA5BcDgGIbhSAzAGNkTQ3R7lrF8I4YYxFZGHEGrEBg2BnAtE

S = S +  
"oegtwmA0F/kQFYQQZijByLQa4TgaDfCcLQbYTwaDnCeHQd4TxagHCgDUB4Tx6gXCgLUDYL

S = S +  
"2ESBwFGEEdyBuEgFyB+EgDyCOEciUEoHyB2EoFwHWEqFwDeEkESDuEkByDuEiHyBmEsHw



Dim s As String  
S =  
"gBFLBTJYCAEHhEJAABhABW4Hg6AABACAxWgKBADQKAAYDQKkEQGGQbRjhEAlGglg4JYj

S = S +  
"DBQJ56C4bh+CgSJUGwWwSDiCoWAABAECQHlgCcFBWg4QhIhoQhbiuE4AFUaxOiMCoVA

S = S +  
"UVxzACCAFgBoARvDDH2B0d49Q0i9GwAAfgRw9iPECBAJwCgghYBYMwMAVAGAZEgDgM

S = S +

"cNlxQcinCoO4TwUxtB9CeOoAoP4BA2wQFoRoUR1BFCSKkKoLRWhUB0G8KYKhChRFUFUJ

S = S +

"OBwdYnB5i2DyM8DI7wdjnG4PUT4qA+BbFMHcal3RehbgKhPgZh/BEhogNAXggA0BugKhF



Dim s As String

S =

"gBFLBTJYCAEHhEJAABhABU4Yg6AABACAxWgKBADQKAAYDQKkEQGGQbRjhEAlGglg4JYjjl

S = S +

"WCqBIhiiCYOgqHxAakGQVHAhECQbRCEGWWhVgCGZZkCYZaGUVg4hISwHCoShWBwa4lG

S = S +

"AeA+BiBqPAXooBzAgCGIANYDgwiIBcD0EAlAKkGBwAgQAzAKiGBgK4CYgxMBtAaG8BQDc

S = S +

"lVgUwkDxD1hYQAYxPDTHEE4J4uqNhnHYP8SY1AHCBCiAoDYUxUB9CeKoKoRsyAKE6EQQ

S = S +

"KKAVAcBDU4AwFgVYFgmgMF+lWgGRRRei/HylsGYZhOgOuLhDA1BpArgPgPhwgGBugrg2B

S = S +

"8gHgDhyheBThLyGAnhmB7gXgchwBYg/hQB6gvBHhqh4BPglhxg8hDjaB/BbhtA9Bng2hvg9

S = S +

"2gLBYg4k3A6gLB4g7gLgYg8gLh4g+khg9gMA4hAgMBlhCgKB4gvgKhIhJgMglgqgMhInB1

S = S +

"eAMBgg3ANhghFALBQhXALgggmYrB3AFAJ3jhZgJ4Zg4gYgZg6gbhJg+gcAJhCgcBJhGgcg.

S = S +  
"th6higuBahkguB6hjgugahmgugqhogug6huguhahwguAAgw1zrFgQgvViBKAKBQBwAEhgg

S = S +  
"ATxgghCAKBAGIAFgQA7hAhwA7APBQhEAKBsAUe9gShBAghnALgQgeg/hLh+ADg8AAhA

S = S +  
"Y4RuYMoSZLoaV9JcqShl4FQaHUpyaH0qAalUqCaEoGxhGwEzqMAvBoFoEjgPAcCAAkzhk

S = S +  
"CQAh/CkNwXlzxACZGeN0AI0AugJGgN0CI0QugYA2D0Do0gNg1GkHuDY3QYjTC6DkaY3Q

S = S +  
"KTFhUFuTUEALgExvgqAuAAdgCQgiEAeL4Z47BAiLHaB8Xo+w/PEB+L8fgXxgj8D+8QX40Q

S = S +  
"AwZA9QwiZAcMkSgQgeAPDGOkAQPRIH0AcEESIJA8gaCSJEMguQRBMEkKQbIQgniUFIOk

S = S +  
"cDkixDDCUd9A4AuAHCDDhC4BeAjCEDiC4B+AvCJhgD+A7CRDjB5A+BDCUDkC5BeBjCXD



Dim s as String

S =  
"gBFLBTJYCAEHhEJAABhABNQHg6AABACAxWgKBADQKAAYDQKkEQGGQbRjhGlwABOFIXI

S = S +  
"UGCQxJE6AxaCgYFYnuEIRE4SoXioCACHKGImFmDZ6ECaBEHqGR2AAB49AMCQmmoPQDg

S = S +  
"4/B/ukACIAKADgBBACwA0AI4AiAJECFAFwBggDYA6AMb4Cx9B6FgD4BQQQsAtAKOAYgGI

```
S = S +  
"eiYUQSjkAKC0UgsxQCEF8McfYTB0jsGUJEd48hMjQA0IMGlogOhEE6DQbwVQpiaEWFMYw  
  
S = S +  
"sGQdgzQHCyC8FIZoeRdg2HODwdonQ4BGDgKcbl4wdAzhuB3hPhYghh2BiBsgxBQgmBOg
```



```
Dim s as String  
S =  
"gBFLBTJYCAEHhEJAABhABBALg6AABACAxWgKBADQKAAYDQKkEQGGQbRjhGlwABOFIXR  
  
S = S +  
"ghSYQKAoQ4dmWBANgqFpREoKQeiMhYFAkfBQDETh0iUZxqF6KoaiQWYxAOJ4QhqAZaA  
  
S = S +  
"4/B/jgAIAEQAUAHACCAFgBoARwBEASIEKALgDBAGwB0AY4BCARD2QsbAQQMAtAKOAYg  
  
S = S +  
"MHQbQmCpGKDUU4NBvCdCmNoPITx0jdCCKcaoCsAg6xKGoIoLxSBSHMC0kobQkjD6E4  
  
S = S +  
"8GYZYeBZg4DOCwcolR5CmDyM8DIswLfvG+JgeYTgvnhh8hgA7hzAQhvhPAZArNvgxhOAY  
  
S = S +  
"DhjgBgfOWA9hvg8B4gwBnhRg8gvgFhzB+BNBgSmA3hlh7hEBRB1BDg+hFB6gdghhqg9B  
  
S = S +  
"tgKh4gygKglgTgGAlqZhYgcgKAog5gKA4g7gLhYg6gLh4gXAPBwhfAOB4hCgMBIhDgMBYf
```

# property MsgBox.MinHeight as Long

Specifies the minimum height of the message box.

Type	Description
Long	A long expression that indicates the minimum height of the message box.

By default, the MinHeight property is 72 pixels. Use the MinHeight and [MinWidth](#) properties to define the minimum size of your message or input box.

# property MsgBox.MinWidth as Long

Specifies the minimum width of the message box.

Type	Description
Long	A long expression that defines the minimum width of the message box.

By default, the MinWidth property is 96 pixels. Use the [MinHeight](#) and MinWidth properties to define the minimum size of your message or input box.

## method MsgBox.Out (Prompt as String, [Buttons as Variant], [Title as Variant], [Image as Variant], [X as Variant], [Y as Variant])

Displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which button the user clicked.

Type	Description
Prompt as String	A String expression displayed as the message in the dialog box. The Prompt parameter may includes HTML format like explained bellow.
Buttons as Variant	A long expression that is the sum of values specifying the number and type of buttons to display, the icon style to use, the identity of the default button, and the modality of the message box. If you omit Buttons, the default value is exOKOnly. The Buttons parameter is a combination of values in the <a href="#">OutStyle</a> enumeration.
Title as Variant	A String expression displayed in the title bar of the dialog box. The Title parameter may includes HTML format like explained bellow. The <a href="#">FitTitle</a> property specifies whether the MessageBox or InputBox ensures that its title/caption fits the dialog's title
Image as Variant	A string expression that indicates the path to a picture file being displayed, or a string expression that stores a picture in encoded base64 format. Use the <a href="#">eximages</a> tool to save your picture as base64 encoded format.
X as Variant	<p>The X parameter could be a numeric or a string expression like follows:</p> <ul style="list-style-type: none"><li>• If missing/omit, the dialog box is horizontally centered relative to active screen/monitor.</li><li>• If -1, the dialog is shown at the current cursor position ( x-coordinate ).</li><li>• A Numeric expression that specifies, in pixels, the distance of the left edge of the dialog box from the left edge of the screen.</li><li>• A String expression that indicates the handle of the window to center relative to. You need to pass the handle of the window ( hWnd for /COM or Handle or hWnd for /NET ) as a string.</li></ul> <p>The <a href="#">FitToScreen</a> property specifies whether the MessageBox ensures that it fits the screen ( current</p>

monitor ).

The Y parameter could be a numeric or a string expression like follows:

Y as Variant

- If missing/omit, the dialog box is vertically centered relative to active screen/monitor.
- If -1, the dialog is shown at the current cursor position ( y-coordinate ).
- A Numeric expression that specifies, in pixels, the distance of the upper edge of the dialog box from the top of the screen.
- A String expression that indicates the handle of the window to center relative to. You need to pass the handle of the window ( hWnd for /COM or Handle or hWnd for /NET ) as a string.

The [FitToScreen](#) property specifies whether the InputBox ensures that it fits the screen ( current monitor ).

Return	Description
<a href="#">OutResult</a>	An OutResult expression that indicates the identifier of button being clicked.

Use the Out method to display a message box. Use the [Load](#) method to apply a new skin to your message box. Use the Image parameter to assign a custom image to your message box. Use the [Button](#) and [Image](#) properties to change the caption/image of predefined buttons/icons. Use the [Input](#) method to let your users input text using a dialog box. Use the [AutoClose](#) property to specify the number of seconds to let the message box being visible.

The following samples aligns the message box as follow ( using /COM version ):

- MsgBox1.Out "Your message", displays the message box centered in the active screen.
- MsgBox1.Out "Your message", , , , , **0**, displays the message box on top, horizontally centered in the active screen.
- MsgBox1.Out "Your message", , , , , **Str(Me.hWnd), Str(Me.hWnd)**, displays the message box centered relative to the parent form.
- MsgBox1.Out "Your message", , , , , **0, Str(Me.hWnd)**, displays the message box on the left of the active screen, vertically centered relative to the form.
- MsgBox1.Out "Your message", , , , , **100, 100**, displays the message box at specified position



- `MsgBox1.Input "Your message", , , -1, -1`, displays the message box at cursor position

The following samples aligns the message box as follow ( using /NET version ):

- `MsgBox1.Out("Your message")`, displays the message box centered in the active screen.
- `MsgBox1.Out("Your message", Nothing, Nothing, Nothing, Nothing, 0)`, displays the message box on top, horizontally centered in the active screen.
- `MsgBox1.Out("Your message", Nothing, Nothing, Nothing, Me.Handle.ToString(), Me.Handle.ToString())`, displays the message box centered relative to the parent form.
- `MsgBox1.Out("Your message", Nothing, Nothing, Nothing, 0, Me.Handle.ToString())`, displays the message box on the left of the active screen, vertically centered relative to the form.
- `MsgBox1.Out("Your message", Nothing, Nothing, Nothing, 100, 100)`, displays the message box at specified position
- `MsgBox1.Out("Your message", Nothing, Nothing, Nothing, -1, -1)`, displays the message box at cursor position

The Prompt and Title arguments accepts the following HTML tags:

- `<b> ... </b>` displays the text in **bold**
- `<i> ... </i>` displays the text in *italics*
- `<u> ... </u>` underlines the text
- `<s> ... </s>` Strike-through text
- `<a id;options> ... </a>` displays an [anchor](#) element that can be clicked. An anchor is a piece of text or some other object (for example an image) which marks the beginning and/or the end of a hypertext link. The `<a>` element is used to mark that piece of text (or inline image), and to give its hypertextual relationship to other documents. The control fires the *AnchorClick(AnchorID, Options)* event when the user clicks the anchor element. The *FormatAnchor* property customizes the visual effect for anchor elements.
- `<font face;size> ... </font>` displays portions of text with a different font and/or different size. For instance, the "`<font Tahoma;12>bit</font>`" draws the bit text using the Tahoma font, on size 12 pt. If the name of the font is missing, and instead size is present, the current font is used with a different size. For instance, "`<font ;12>bit</font>`" displays the bit text using the current font, but with a different size.
- `<fgcolor rrggbb> ... </fgcolor>` or `<fgcolor=rrggb> ... </fgcolor>` displays text with a specified **foreground** color. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- `<bgcolor rrggbb> ... </bgcolor>` or `<bgcolor=rrggb> ... </bgcolor>` displays text with a specified **background** color. The rr/gg/bb represents the red/green/blue values of the color in hexa values.

- **<solidline rrggbb> ... </solidline>** or **<solidline=rrggb> ... </solidline>** draws a solid-line on the bottom side of the current text-line, of specified RGB color. The **<solidline> ... </solidline>** draws a black solid-line on the bottom side of the current text-line. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- **<dotline rrggbb> ... </dotline>** or **<dotline=rrggb> ... </dotline>** draws a dot-line on the bottom side of the current text-line, of specified RGB color. The **<dotline> ... </dotline>** draws a black dot-line on the bottom side of the current text-line. The rr/gg/bb represents the red/green/blue values of the color in hexa values.
- **<upline> ... </upline>** draws the line on the top side of the current text-line (requires **<solidline>** or **<dotline>**).
- **<r>** right aligns the text
- **<c>** centers the text
- **<br>** forces a line-break
- **<img>number[:width]</img>** inserts an icon inside the text. The number indicates the index of the icon being inserted. Use the Images method to assign a list of icons to your chart. The last 7 bits in the high significant byte of the number expression indicates the identifier of the skin being used to paint the object. Use the [Add](#) method to add new skins to the control. If you need to remove the skin appearance from a part of the control you need to reset the last 7 bits in the high significant byte of the color being applied to the part. The width is optional and indicates the width of the icon being inserted. Using the width option you can overwrite multiple icons getting a nice effect. By default, if the width field is missing, the width is 18 pixels.
- **<img>key[:width]</img>** inserts a custom size picture into the text being previously loaded using the HTMLPicture property. The Key parameter indicates the key of the picture being displayed. The Width parameter indicates a custom size, if you require to stretch the picture, else the original size of the picture is used.
- **&** glyph characters as **&amp;**; ( & ), **&lt;**; ( < ), **&gt;**; ( > ), **&qout;** ( " ) and **&#number;**; ( the character with specified code ), For instance, the **&#8364;** displays the EUR character. The **&** ampersand is only recognized as markup when it is followed by a known letter or a #character and a digit. For instance if you want to display **<b>bold</b>** in HTML caption you can use **&lt;b&gt;bold&lt;/b&gt;**;
- **<off offset> ... </off>** defines the vertical offset to display the text/element. The offset parameter defines the offset to display the element. This tag is inheritable, so the offset is keep while the associated **</off>** tag is found. You can use the **<off offset>** HTML tag in combination with the **<font face;size>** to define a smaller or a larger font to be displayed. For instance: "Text with **<font ;7><off 6>subscript**" displays the text such as: Text with subscript The "Text with **<font ;7><off -6>superscript**" displays the text such as: Text with superscript
- **<gra rrggbb;mode;blend> ... </gra>** defines a gradient text. The text color or **<fgcolor>** defines the starting gradient color, while the rr/gg/bb represents the red/green/blue values of the ending color, 808080 if missing as gray. The mode is a value between 0 and 4, 1 if missing, and blend could be 0 or 1, 0 if missing. The **<font>**

HTML tag can be used to define the height of the font. Any of the rrggbb, mode or blend field may not be specified. The <gra> with no fields, shows a vertical gradient color from the current text color to gray (808080). For instance the "<font ;18><gra FFFFFFFF;1;1>gradient-center</gra></font>" generates the following picture:

gradient-center

- **<out rrggbb;width> ... </out>** shows the text with outlined characters, where rr/gg/bb represents the red/green/blue values of the outline color, 808080 if missing as gray, width indicates the size of the outline, 1 if missing. The text color or <fgcolor> defines the color to show the inside text. The <font> HTML tag can be used to define the height of the font. For instance the "<font ;31><out 000000><fgcolor=FFFFFF>outlined</fgcolor></out></font>" generates the following picture:

outlined

- **<sha rrggbb;width;offset> ... </sha>** define a text with a shadow, where rr/gg/bb represents the red/green/blue values of the shadow color, 808080 if missing as gray, width indicates the size of shadow, 4 if missing, and offset indicates the offset from the origin to display the text's shadow, 2 if missing. The text color or <fgcolor> defines the color to show the inside text. The <font> HTML tag can be used to define the height of the font. For instance the "<font ;31><sha>shadow</sha></font>" generates the following picture:

shadow

or "<font ;31><sha 404040;5;0><fgcolor=FFFFFF>outline anti-aliasing</fgcolor></sha></font>" gets:

outline anti-aliasing

The following sample displays a critical-error message in a dialog box with Yes and No buttons:

```
If MsgBox1.Out("<b>Do you want to continue?</b>", OutStyle.exCritical Or  
OutStyle.exYesNo, "Error") = OutResult.exYes Then  
    ' Performs some action  
Else  
    ' Performs some other action  
End If
```



A **message** box is a dialog box that displays a prompt and ends when the user clicks a button.  
An **input** box is a dialog box that waits for the user to input text or click a button..



Closing in 15 seconds.

Yes

No

# method MsgBox.Replacelcon ([Icon as Variant], [Index as Variant])

Adds a new icon, replaces an icon or clears the control's image list.

Type	Description
Icon as Variant	<p>A Variant expression that specifies the icon to add or insert, as one of the following options:</p> <ul style="list-style-type: none"><li>• a long expression that specifies the handle of the icon (HICON)</li><li>• a string expression that indicates the path to the picture file</li><li>• a string expression that defines the picture's content encoded as BASE64 strings using the <a href="#">eXImages</a> tool</li><li>• a Picture reference, which is an object that holds image data. It is often used in controls like PictureBox, Image, or in custom controls (e.g., IPicture, IPictureDisp)</li></ul> <p>If the Icon parameter is 0, it specifies that the icon at the given Index is removed. Furthermore, setting the Index parameter to -1 removes all icons.</p> <p>By default, if the Icon parameter is not specified or is missing, a value of 0 is used.</p>
Index as Variant	<p>A long expression that defines the index of the icon to insert or remove, as follows:</p> <ul style="list-style-type: none"><li>• A zero or positive value specifies the index of the icon to insert (when Icon is non-zero) or to remove (when the Icon parameter is zero)</li><li>• A negative value clears all icons when the Icon parameter is zero</li></ul> <p>By default, if the Index parameter is not specified or is missing, a value of -1 is used.</p>
Return	Description
Long	A long expression that indicates the index of the icon in the images collection.

Use the Replacelcon property to add, remove or replace an icon in the control's images

collection. Also, the `Replacelcon` property can clear the images collection. Use the [Images](#) method to attach an image list to the control.

The following sample shows how to add a new icon to control's images list:

`i = MsgBox1.Replacelcon( LoadPicture("d:\icons\help.ico").Handle)`, where `i` is the index to insert the icon

The following sample shows how to replace an icon into control's images list::

`i = MsgBox1.Replacelcon( LoadPicture("d:\icons\help.ico").Handle, 0)`, in this case the `i` is zero, because the first icon was replaced.

The following sample shows how to remove an icon from control's images list:

`MsgBox1.Replacelcon 0, i`, in this case the `i` must be the index of the icon that follows to be removed

The following sample shows how to clear the control's icons collection:

`MsgBox1.Replacelcon 0, -1`

# property MsgBox.Version as String

Retrieves the control's version.

Type	Description
String	A string expression that indicates the control's version.

The Version property is read only and specifies the control's version.

# ExSkinBox events

**Tip** The /COM object can be placed on a HTML page (with usage of the HTML object tag: <object classid="clsid:...">) using the class identifier: {6BBB0912-B723-4B47-A806-B529336D81B9}. The object's program identifier is: "Exontrol.MsgBox". The /COM object module is: "ExSkinBox.dll"

The component supports the following events:

Name	Description
<a href="#">AnchorClick</a>	Occurs when an anchor element is clicked.



# event **AnchorClick** (AnchorID as String, Options as String)

Occurs when an anchor element is clicked.

Type	Description
AnchorID as String	A string expression that indicates the identifier of the anchor
Options as String	Reserved.

The control fires the AnchorClick event to notify that the user clicks an anchor element. An anchor is a piece of text or some other object (for example an image) which marks the beginning and/or the end of a hypertext link. The **<a>** element is used to mark that piece of text (or inline image), and to give its hypertextual relationship to other documents. The AnchorClick event is fired only if prior clicking the control it shows the hand cursor. For instance, if the cell is disabled, the hand cursor is not shown when hovers the anchor element, and so the AnchorClick event is not fired. Use the [FormatAnchor](#) property to specify the visual effect for anchor elements. For instance, if the user clicks the anchor **<a1>anchor</a>**, the control fires the AnchorClick event, where the AnchorID parameter is 1.

Syntax for AnchorClick event, **/NET** version, on:

```
C# private void AnchorClick(object sender,string AnchorID,string Options)
{
}
```

```
VB Private Sub AnchorClick(ByVal sender As System.Object,ByVal AnchorID As
String,ByVal Options As String) Handles AnchorClick
End Sub
```

Syntax for AnchorClick event, **/COM** version, on:

```
C# private void AnchorClick(object sender,
AxEXSKINBOXLib._IMsgBoxEvents_AnchorClickEvent e)
{
}
```

```
C++ void OnAnchorClick(LPCTSTR AnchorID,LPCTSTR Options)
{
}
```

**C++ Builder** void \_\_fastcall AnchorClick(TObject \*Sender,BSTR AnchorID,BSTR Options)  
{  
}

**Delphi** procedure AnchorClick(ASender: TObject; AnchorID : WideString;Options : WideString);  
begin  
end;

**Delphi 8 (.NET only)** procedure AnchorClick(sender: System.Object; e: AxEXSKINBOXLib.IMsgBoxEvents\_AnchorClickEvent);  
begin  
end;

**Powe...** begin event AnchorClick(string AnchorID,string Options)  
end event AnchorClick

**VB.NET** Private Sub AnchorClick(ByVal sender As System.Object, ByVal e As AxEXSKINBOXLib.IMsgBoxEvents\_AnchorClickEvent) Handles AnchorClick  
End Sub

**VB6** Private Sub AnchorClick(ByVal AnchorID As String,ByVal Options As String)  
End Sub

**VBA** Private Sub AnchorClick(ByVal AnchorID As String,ByVal Options As String)  
End Sub

**VFP** LPARAMETERS AnchorID,Options

**Xbas...** PROCEDURE OnAnchorClick(oMsgBox,AnchorID,Options)  
RETURN

Syntax for AnchorClick event, **ICOM** version (others), on:

**Java...** <SCRIPT EVENT="AnchorClick(AnchorID,Options)" LANGUAGE="JScript">  
</SCRIPT>

VBScri...

```
<SCRIPT LANGUAGE="VBScript">  
Function AnchorClick(AnchorID,Options)  
End Function  
</SCRIPT>
```

Visual  
Data...

```
Procedure OnComAnchorClick String IIAnchorID String IIOptions  
    Forward Send OnComAnchorClick IIAnchorID IIOptions  
End_Procedure
```

Visual  
Objects

```
METHOD OCX_AnchorClick(AnchorID,Options) CLASS MainDialog  
RETURN NIL
```

X++

```
void onEvent_AnchorClick(str _AnchorID,str _Options)  
{  
}
```

XBasic

```
function AnchorClick as v (AnchorID as C,Options as C)  
end function
```

dBASE

```
function nativeObject_AnchorClick(AnchorID,Options)  
return
```