

Freelance Graphics: Activate method

```
{button ,AL(^H_ACTIVATE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;H_PLACEMENTBLOCK_CLASS;',0)}
```

[See list of classes](#)

Simulate a click action on a document or a "Click here..." block object. The object becomes the current object and is brought to the foreground.

Syntax

object.Activate

Parameters

None

Return values

None

Examples

```
CurrentApplication.Activate
```

Freelance Graphics: AddPoint method

{button ,AL('H_ADDPOINT_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS';,0)} [See list of classes](#)

Add a point to the object.

Syntax

drawobject.AddPoint(*segment*, *x*, *y*)

Parameters

segment as Integer

Identifies the point after which you want to add the new point. (The first drawn point is 1.) If the object in question is not a polygon, line, arrow, or curve, this value is ignored.

x as Integer

Horizontal coordinate of the point relative to the left edge of the window, in twips.

y as Integer

Vertical coordinate of the point relative to the bottom of the window, in twips.

Return values

None

Freelance Graphics: AddToPageSelection method

{button ,AL(^H_ADDTOPAGESELECTION_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Add a page to the set of selected pages in the document.

Syntax

documentobject.AddToPageSelection(pageobject)

Parameters

pageobject as Page

Any page object.

Return values

None

Examples

CurrentDocument.AddToPageSelection MyPage

Freelance Graphics: AddToSelection method

{button ,AL(^H_ADDTOSELECTION_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;H_PAGESELECTION_CLASSES;H_SELECTION_CLASS;','0)} [See list of classes](#)

Add the specified object to the set of selected documents, pages, or objects.

Syntax

object.AddToSelection(drawobject)

Parameters

drawobject as *DrawObject*

Any DrawObject object.

Return values

None

Examples

`CurrentPage.Selection.AddToSelection MyRect`

or

`Selection.AddToSelection MyRect`

Freelance Graphics: Align method

{button ,AL(^H_ALIGN_METHOD_MEMDEF_RT;H_SELECTION_CLASS;','0)} [See list of classes](#)

Align the currently selected objects.

Syntax

selectionobject.Align(aligntype, centeronpage)

Parameters

aligntype as Variant (Enumerated)

<u>Value</u>	<u>Description</u>
\$AlignLeft	Align left sides
\$AlignRight	Align right sides
\$AlignTop	Align tops
\$AlignBottom	Align bottoms
\$AlignRow	Center in a row
\$AlignColumn	Center in a column
\$AlignPoint	Center on a point

centeronpage as Integer (Boolean)

<u>Value</u>	<u>Description</u>
TRUE (-1)	Center on page
FALSE (0)	Do not center on page

Return values

None

Examples

`Selection.Align $AlignLeft, False`

Freelance Graphics: ApplyStyle method

{button ,AL(^H_APPLYSTYLE_METHOD_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;','0)} [See list of classes](#)

Apply the specified style name to the text block.

Syntax

textblockobject.ApplyStyle(*stylename*)

Parameters

stylename as *String*

Name to apply to the style.

Return values

None

Freelance Graphics: BrowseDiagrams method

{button ,AL(^H_BROWSEDIAGRAMS_METHOD_MEMDEF_RT;H_PLACEMENTBLOCK_CLASS;','0)} [See list of classes](#)

Launch the Clip Art browser using the specified diagram file. When the user clicks OK in the browser, insert the currently selected diagram in the "Click here..." block.

Syntax

placementblockobject.**BrowseDiagrams**(*filename*)

Parameters

filename as *String*

Optional name of the diagram file. If omitted, defaults to the first .DGM file in the \SMASTERS\FLG directory.

Return values

None

Freelance Graphics: BrowseSymbols method

{button ,AL(^H_BROWSESYMBOLS_METHOD_MEMDEF_RT;H_PLACEMENTBLOCK_CLASS;',0)} [See list of classes](#)

Launch the Clip Art browser using the specified symbol file. When the user clicks OK in the browser, insert the currently selected symbol in the "Click here..." block.

Syntax

placementblockobject.BrowseSymbols(*filename*)

Parameters

filename as String

Optional name of the symbols file. If omitted, defaults to the first .SYM file in the \SMASTERS\FLG directory.

Return values

None

Freelance Graphics: Cascade method

```
{button ,AL(^H_CASCADE_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;  
    ,0)} See list of classes
```

Cascade all MDI windows in the application or document window.

Syntax

applicationwindowobject.**Cascade**

Parameters

None

Return values

None

Examples

```
CurrentApplication.Cascade
```

Freelance Graphics: ClearSelection method

```
{button ,AL(^H_CLEARSELECTION_METHOD_MEMDEF_RT;H_PAGESELECTION_CLASS;H_SELECTION_CLAS  
S;';0)} See list of classes
```

Deselect all selected items.

Syntax

object.ClearSelection

Parameters

None

Return values

None

Examples

```
Selection.ClearSelection
```

Freelance Graphics: CloseWindow method

{button ,AL(^H_CLOSEWINDOW_METHOD_MEMDEF_RT;H_APPLICATION_CLASS;')} [See list of classes](#)

Close the current MDI window.

Syntax

applicationobject.CloseWindow(*closewindow*)

Parameters

closewindow as Integer

Optional and ignored.

Return values

None

Status codes

None

Examples

```
CurrentDocument.Save  
CurrentDocument.CloseWindow
```

Freelance Graphics: Close method

{button ,AL(^H_CLOSE_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCUMENT_CLASS;H_DOCWINDOW_CLASS;','0)} [See list of classes](#)

Close the Freelance Graphics application window (exit Freelance Graphics), or close a document or document window.

Syntax

object.Close(*exitapplication*)

Parameters

exitapplication as Integer

Optional and ignored.

Return values

None

Examples

CurrentApplication.Close

Freelance Graphics: ColorToRGB method

{button ,AL('H_COLORTORGB_METHOD_MEMDEF_RT;H_COLORS_CLASS;',0)} [See list of classes](#)

Return the RGB value of the color produced by the current values of the Red, Green, and Blue properties of a Color object.

Syntax

colorobject.ColorToRGB(*colorobject*)

Parameters

colorobject as Color

The Color object whose RGB value you want returned.

Return values

Long, RGB value.

Examples

```
Dim MyColorRGB as Long
```

```
MyColorRGB = CurrentApplication.Colors.ColorToRGB(MyColor)
```

Freelance Graphics: Connect method

{button ,AL(^H_CONNECT_METHOD_MEMDEF_RT;H_SELECTION_CLASS;',0)} [See list of classes](#)

Connect the currently selected line objects.

Syntax

selection.Connect

Parameters

None

Return values

None

Example

`Selection.Connect`

Freelance Graphics: ConvertTo method

{button ,AL(^H_CONVERTTO_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;!,0)} [See list of classes](#)

Convert the object to lines or polygons.

Syntax

drawobject.ConvertTo(conversiontype)

Parameters

conversiontype as Integer (Enumerated)

<u>Value</u>	<u>Description</u>
\$ConvertToPolygons	Create one or more polygons from the object
\$ConvertToLines	Create one or more lines from the object

Return values

None

Examples

```
Dim MyRect as DrawObject
Set MyRect = CurrentPage.CreateRect
MyRect.ConvertTo ($ConvertToLines)
```

Freelance Graphics: CopyPage method

{button ,AL(^H_COPYPAGE_METHOD_MEMDEF_RT;H_PAGE_CLASS;')0)} [See list of classes](#)

Copy a page to the Clipboard.

Syntax

pageobject.CopyPage

Parameters

None

Return values

None

Examples

CurrentPage.CopyPage

Freelance Graphics: CopySelection method

{button ,AL(^H_COPYSELECTION_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Copy the selection to the Clipboard.

Syntax

documentobject.CopySelection

Parameters

None

Return values

None

Examples

CurrentPage.CopySelection

Freelance Graphics: Copy method

{button ,AL(^H_COPY_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;0)} [See list of classes](#)

Copy an object to the Clipboard.

Syntax

drawobject.Copy

Parameters

None

Return values

None

Examples

```
CurrentSelection.Copy
```

or

```
Dim MyRect as DrawObject  
Set MyRect = CurrentPage.CreateRect  
MyRect.Copy
```

Freelance Graphics: CreateArrow method

{button ,AL('H_CREATEARROW_METHOD_MEMDEF_RT;H_PAGE_CLASS';,0)} [See list of classes](#)

Draw an arrow on the page.

Syntax

pageobject.CreateArrow(*xstart*, *ystart*, *xfinish*, *yfinish*)

Parameters

xstart as Integer

(Optional, see note) Starting horizontal coordinate of the arrow, in twips.

ystart as Integer

(Optional, see note) Starting vertical coordinate of the arrow, in twips.

xfinish as Integer

(Optional, see note) Ending horizontal coordinate of the arrow, in twips.

yfinish as Integer

(Optional, see note) Ending vertical coordinate of the arrow, in twips.

Note If you omit the parameters, the arrow will be centered on the page.

Return values

An instance of the DrawObject class (the drawn arrow).

Examples

```
Dim MyArrow as DrawObject
Set MyArrow = CurrentPage.CreateArrow(1000, 1000, 3000, 3000)
```

Freelance Graphics: CreateChart method

{button ,AL(^H_CREATECHART_METHOD_MEMDEF_RT;H_PAGE_CLASS;')} [See list of classes](#)

Create a chart on the page.

Note For more information on working with charts in LotusScript, search for ChartBase in the Help Index.

Syntax

pageobject.CreateChart(*x*, *y*, *width*, *height*)

Parameters

x as Integer

(Optional, see note) Right edge of the chart object relative to the left edge of the window, in [twips](#).

y as Integer

(Optional, see note) Bottom edge of the chart object relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

Note If you omit the parameters, the chart will be centered on the page.

Return values

An instance of the DrawObject class (the drawn chart).

Examples

```
Dim MyChart as Chart
Set MyChart = CurrentPage.CreateChart (4000,10000,6000,6000)
```

Freelance Graphics: CreateComment method

{button ,AL(^H_CREATECOMMENT_METHOD_MEMDEF_RT;H_PAGE_CLASS;!,0)} [See list of classes](#)

Create TeamReview comments on a page.

Syntax

pageobject.CreateComment(*x*, *y*, *width*, *height*)

Parameters

x as Integer

(Optional, see note) Right edge of the comment object relative to the left edge of the window, in [twips](#).

y as Integer

(Optional, see note) Bottom edge of the comment object relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

Note If you omit the parameters, the comment will be centered on the page.

Return values

An instance of the DrawObject class (the TeamReview comments).

Examples

```
CurrentPage.CreateComment "Four score and seven years ago"
```

Freelance Graphics: CreateLine method

{button ,AL('H_CREATELINE_METHOD_MEMDEF_RT;H_PAGE_CLASS;')} [See list of classes](#)

Draw a line on the page.

Syntax

pageobject.CreateLine(*xstart*, *ystart*, *xfinish*, *yfinish*)

Parameters

xstart as Integer

(Optional, see note) Starting horizontal coordinate of the line, in twips.

ystart as Integer

(Optional, see note) Starting vertical coordinate of the line, in twips.

xfinish as Integer

(Optional, see note) Ending horizontal coordinate of the line, in twips.

yfinish as Integer

(Optional, see note) Ending vertical coordinate of the line, in twips.

Note If you omit the parameters, the line will be centered on the page.

Return values

An instance of the DrawObject class (the drawn line).

Example

```
Dim MyLine as DrawObject
Set MyLine = CurrentPage.CreateLine(1000, 1000, 3000, 3000)
```

Freelance Graphics: CreateMovie method

{button ,AL(^H_CREATEMOVIE_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Add a movie to the page.

Syntax

pageobject.CreateMovie(x, y, width, height, filename)

Parameters

x as Integer

(Optional, see note) Left edge of the movie window relative to the left edge of the window, in [twips](#).

y as Integer

(Optional, see note) Bottom edge of the movie window relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

filename as String

Name of the file containing the movie to be added to the page.

Note If you omit the optional parameters, the movie will be centered on the page.

Return values

An instance of the DrawObject class (the movie object).

Freelance Graphics: CreateOval method

{button ,AL('H_CREATEOVAL_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Draw an oval on the page.

Syntax

pageobject.CreateOval(*xcenter*, *ycenter*, *width*, *height*)

Parameters

xcenter as Integer

(Optional, see note) Horizontal coordinate of the center of the oval, in twips.

ycenter as Integer

(Optional, see note) Vertical coordinate of the center of the oval, in twips.

width as Integer

(Optional, see note) Width of the oval, in twips.

height as Integer

(Optional, see note) Height of the oval, in twips.

Note If you omit the parameters, the oval will be centered on the page.

Return values

An instance of the DrawObject class (the drawn oval).

Examples

```
Dim MyOval as DrawObject
Set MyOval = CurrentPage.CreateOval(1000, 1000, 3000, 3000)
```


Freelance Graphics: CreatePageFromTemplate method

{button ,AL(^H_CREATEPAGEFROMTEMPLATE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

Create a new page using a content topic.

Syntax

document.CreatePageFromTemplate pagetitle, templateindex

Parameters

PageTitle as String

Name for the new page.

TemplateIndex as Integer

Identifies the content topic page type (the first type is 1).

Return values

An instance of the Page class.

Freelance Graphics: CreatePage method

{button ,AL(^H_CREATEPAGE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;!,0)} [See list of classes](#)

Create a new page.

Syntax

documentobject.CreatePage pagetitle, masterindex

Parameters

pagetitle as String

Name for the new page

masterindex as Integer

Identifies the type of content topic page (the first type is 1).

Return values

An instance of the Page class.

Examples

```
Dim Page1 as Page
```

```
Set Page1 = CurrentDocument.CreatePage("My Title Page",1)
```

Freelance Graphics: CreatePlacementBlock method

{button ,AL(^H_CREATEPLACEMENTBLOCK_METHOD_MEMDEF_RT;H_PAGE_CLASS;';0)} [See list of classes](#)

Create a "Click here..." block.

Syntax

pageobject.CreatePlacementBlock(x, y, width, height, blocktype, text)

Parameters

x as Integer

(Optional, see note) Left edge of the object relative to the left edge of the window, in twips.

y as Integer

(Optional, see note) Bottom edge of the object relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

blocktype as Variant

<u>Value</u>	<u>Description</u>
pbTypeText	Click here to enter text
pbTypeSymbol	Click here to add a symbol
pbTypeChart	Click here to create a chart
pbTypeOrgChart	Click here to create an organization chart
pbTypeTable	Click here to create a table
pbTypeButton	Click here to add a button

text as String

Prompt text to include in the "Click here..." box.

Note If you omit the optional parameters, the block will be centered on the page.

Return values

An instance of the DrawObject class.

Freelance Graphics: CreateRect method

{button ,AL('H_CREATERECT_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Draw a rectangle on the page.

Syntax

pageobject.CreateRect(*xcenter*, *ycenter*, *width*, *height*)

Parameters

xcenter as Integer

(Optional, see note) Horizontal coordinate of the center of the rectangle, in twips.

ycenter as Integer

(Optional, see note) Vertical coordinate of the center of the rectangle, in twips.

width as Integer

(Optional, see note) Width of the rectangle, in twips.

height as Integer

(Optional, see note) Height of the rectangle, in twips.

Note If you omit the parameters, the rectangle will be centered on the page.

Return values

An instance of the DrawObject class (the drawn rectangle).

Examples

```
Dim MyRect as DrawObject
Set MyRect = CurrentPage.CreateRect(1000, 1000, 2000, 2000)
```

Freelance Graphics: CreateStyle method

{button ,AL(^H_CREATESTYLE_METHOD_MEMDEF_RT;H_TEXTBLOCK_CLASS;','0)} [See list of classes](#)

Create a new named style based on the attributes of the text block.

Syntax

textblockobject.CreateStyle(*stylename*)

Parameters

stylename as *String*

Name of the new style.

Return values

None

Freelance Graphics: CreateSymbol method

{button ,AL('H_CREATESYMBOL_METHOD_MEMDEF_RT;H_PAGE_CLASS';,0)} [See list of classes](#)

Add a symbol to the page.

Syntax

pageobject.CreateSymbol(*x*, *y*, *width*, *height*, *filename*, *index*)

Parameters

x as Integer

(Optional, see note) Left edge of the symbol relative to the left edge of the window, in [twips](#).

y as Integer

(Optional, see note) Bottom edge of the symbol relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

filename as String

Name of the file containing the symbol.

index as Integer

Sequence number of the symbol in the file.

Note If you omit the optional parameters, the symbol will be centered on the page.

Return values

An instance of the DrawObject class (the symbol object).

Freelance Graphics: CreateTable method

{button ,AL('H_CREATETABLE_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Create a table on the page.

Syntax

pageobject.CreateTable(*type*, *rows*, *columns*, *x*, *y*, *width*, *height*)

Parameters

type as Integer

See the Create Table dialog's Table Gallery for examples of the following grid types.

<u>Value</u>	<u>Description</u>
1	Full grid
2	Grid except for first row and column
3	Outline around outside of table
4	No grid lines

rows as Integer

Number of rows.

columns as Integer

Number of columns.

x as Integer

(Optional, see note) Left edge of the table relative to the left edge of the window, in twips.

y as Integer

(Optional, see note) Bottom edge of the table relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

Note If you omit the optional parameters, the table will be centered on the page.

Return values

An instance of the Table class.

Freelance Graphics: CreateText method

{button ,AL(^H_CREATETEXT_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Create a text block.

Syntax

pageobject.CreateText(*x*, *y*, *width*, *height*)

Parameters

x as Integer

(Optional, see note) Left edge of the text block relative to the left edge of the window, in [twips](#).

y as Integer

(Optional, see note) Bottom edge of the text block relative to the bottom of the window, in twips.

width as Integer

(Optional, see note) Width in twips.

height as Integer

(Optional, see note) Height in twips.

Note If you omit the parameters, the text will be centered on the page.

Return values

An instance of the DrawObject class.

Freelance Graphics: CutPage method

{button ,AL(^H_CUTPAGE_METHOD_MEMDEF_RT;H_PAGE_CLASS;')} [See list of classes](#)

Cut a Page object and place it on the Clipboard.

Syntax

pageobject.CutPage

Parameters

None

Return values

None

Examples

CurrentPage.CutPage

Freelance Graphics: CutSelection method

{button ,AL('H_CUTSELECTION_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;');0)} [See list of classes](#)

Cut the current selection and place it on the Clipboard.

Syntax

documentobject.CutSelection

Parameters

None

Return values

None

Examples

CurrentSelection.CutSelection

Freelance Graphics: Cut method

{button ,AL(^H_CUT_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;')0)} [See list of classes](#)

Cut a DrawObject and place it on the Clipboard.

Syntax

drawobject.Cut

Parameters

None

Return values

None

Examples

```
Dim MyRect as DrawObject
Set MyRect = CurrentPage.CreateRect()
MyRect.Cut
```

Freelance Graphics: DeleteCol method

{button ,AL(^H_DELETECOL_METHOD_MEMDEF_RT;H_TABLE_CLASS;')} [See list of classes](#)

Delete a specified column from the table.

Syntax

tableobject.DeleteCol(column)

Parameters

column as Integer

Number of the column to delete.

Return values

None

Freelance Graphics: DeletePage method

{button ,AL(^H_DELETEPAGE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Delete the specified page.

Syntax

documentobject.DeletePage(pageobject)

Parameters

pageobject as *Page*

The page to delete.

Return values

None

Freelance Graphics: DeleteReviewer method

{button ,AL(^H_DELETEREVIEWER_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Open the dialog box used to delete a TeamReview reviewer.

Syntax

documentobject.DeleteReviewer

Parameters

None

Return values

None

Freelance Graphics: DeleteRow method

{button ,AL(^H_DELETE_ROW_METHOD_MEMDEF_RT;H_TABLE_CLASS;',0)} [See list of classes](#)

Delete a specified row from the table.

Syntax

tableobject.DeleteRow(row)

Parameters

row as Integer

The number of the row to delete.

Return values

None

Freelance Graphics: DeleteSpeakerNote method

{button ,AL(^H_DELETESPEAKERNOTE_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Delete speaker notes from the page.

Syntax

pageobject.DeleteSpeakerNote

Parameters

None

Return values

None

Examples

CurrentPage.DeleteSpeakerNote

Freelance Graphics: DistributeForComment method

{button ,AL(^H_DISTRIBUTEFORCOMMENT_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')0)} [See list of classes](#)

Open the Distribute for Comment dialog box.

Syntax

documentobject.**DistributeForComment**

Parameters

None

Return values

None

Freelance Graphics: EnterEditMode

{button ,AL(^H_ENTEREDITMODE_METHOD_MEMDEF_RT;H_TEXTBLOCK_CLASS;',0)} [See list of classes](#)

Enter edit mode on a text block.

Syntax

textblockobject.EnterEditMode

Parameters

None

Return values

None

Freelance Graphics: FindNextObject method

{button ,AL('H_FINDNEXTOBJECT_METHOD_MEMDEF_RT;H_PAGE_CLASS;');0)} [See list of classes](#)

Find the next occurrence of the specified named object on the Page.

Note Use FindObject to find the first occurrence of an object on a Page.

Syntax

pageobject.FindNextObject(*objectname*, *afterme*)

Parameters

objectname as String

Name of the object to find.

afterme as DrawObject

(Optional) The object after which to begin the search.

Return values

An instance of the DrawObject class (the found object).

Examples

```
Dim Rect2 as DrawObject
Set Rect2 = CurrentPage.FindNextObject ("MyRect")
```

Freelance Graphics: FindObject method

{button ,AL('H_FINDOBJECT_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;H_PAGE_CLASS;',0)} [See list of classes](#)

Find the first occurrence of the specified named object in the Document or on the Page.

Note Use FindNextObject to find subsequent occurrences of an object on a Page.

Syntax

object.FindObject(*objectname*)

Parameters

objectname as String

Name of the object to find.

Return values

An instance of the DrawObject class.

Examples

```
Dim Rect1 as DrawObject  
Set Rect1 = CurrentPage.FindObject("My Rect")
```

Freelance Graphics: Flip method

{button ,AL(`H_FLIP_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

Flip the object top-to-bottom or left-to-right.

Syntax

drawobject.Flip(flipdirection)

Parameters

flipdirection as Variant (Enumerated)

<u>Value</u>	<u>Description</u>
\$FlipLeftToRight	Mirror the image on the vertical plane
\$FlipTopToBottom	Mirror the image on the horizontal plane

Return values

None

Examples

```
CurrentSelection.Flip($FlipLeftToRight)
```

Freelance Graphics: GetBulletCount method

{button ,AL(^H_GETBULLETCOUNT_METHOD_MEMDEF_RT;H_TEXTBLOCK_CLASS;','0)} [See list of classes](#)

Return the number of bulleted items in the text block.

Syntax

textblock.GetBulletCount

Parameters

None

Return values

Number of bullets (Integer)

Freelance Graphics: GetCell method

{button ,AL(^H_GETCELL_METHOD_MEMDEF_RT;H_TABLE_CLASS;')} [See list of classes](#)

Return the specified cell as a drawn object.

Syntax

tableobject.GetCell(*row*, *column*)

Parameters

row as *Integer*

Row number of the row containing the text.

column as *Integer*

Column number of the column containing the text.

Return values

An instance of the TextBlock class.

Freelance Graphics: GetEnumerator method

{button ,AL(^H_GETENUM_METHOD_MEMDEF_RT;H_APPLICATION_CLASS;')0)} [See list of classes](#)

Resolve the specified enumerated type to its value. This is useful for cross-application scripts.

Syntax

applicationobject.GetEnumerator(*enumerationname*)

Parameters

enumerationname as Variant

An enumeration name as a string, with or without the \$.

Return values

Value of the specified enumeration name.

Example

```
CurrentApplication.GetEnumerator("ViewDraw")
```


Freelance Graphics: GetIndex method

{button ,AL(^H_GETINDEX_METHOD_MEMDEF_RT;H_COLORS_CLASS;H_DOCUMENTS_CLASS;H_OBJECTS_CLASS;H_PAGES_CLASS;';0)} [See list of classes](#)

Returns the index (e.g., the fifth object in the set).

Syntax

objectset.GetIndex(*object*, *startingindex*)

Parameters

object as *object*

The instance of the object whose index you want.

startingindex as *Integer*

Optional starting position in the set for the search.

Return values

Integer indicating the object occurrence within the set of objects (for example, the page number of a specified page within the set of pages).

Example

```
Dim ThisIndex as Integer  
ThisIndex = CurrentPage.Objects.GetIndex(MyRect)
```

or

```
Dim ThisIndex as Integer  
ThisIndex = CurrentApplication.Colors.GetIndex(MyColor)
```

Freelance Graphics: GetMarkup method

{button ,AL(^H_GETMARKUP_METHOD_MEMDEF_RT;H_TEXTBLOCK_CLASS;'0)} [See list of classes](#)

Return the text and all markup attributes from a TextBlock, as a string.

Syntax

textblockobject.GetMarkup

Parameters

None

Return values

A string containing the text from a TextBlock object, and all markup attributes for that text.

Freelance Graphics: GetNearestColor method

{button ,AL(^H_GETNEARESTCOLOR_METHOD_MEMDEF_RT;H_COLORS_CLASS;',0)} [See list of classes](#)

Return the closest available color (in the Freelance Color Library) to the color specified.

Syntax

colorobject.GetNearestColor(*colorobject*)

Parameters

colorobject as Color

The Color object whose color you want to match as closely as possible.

Return values

An object of the Color class that is the closest match to the passed color.

Example

```
Dim MyLibraryColor as Color
Dim MyColor as Color
Set MyColor = CurrentApplication.Colors.RGBToColor(17395023)
Set MyLibraryColor = CurrentApplication.Colors.GetNearestColor(MyColor)
```

Freelance Graphics: GetNearestIndex method

{button ,AL(^H_GETNEARESTINDEX_METHOD_MEMDEF_RT;H_COLOR_CLASS;',0)} [See list of classes](#)

Return the index of the closest available color for the color specified.

Syntax

colorobject.GetNearestIndex(*colorobject*)

Parameters

colorobject as Color

The Color object whose color you want to match as closely as possible.

Return values

An integer containing the index of the Color class that is the closest match to the passed color.

Freelance Graphics: GetNthBullet method

{button ,AL(^H_GETNTHBULLET_METHOD_MEMDEF_RT;H_TEXTBLOCK_CLASS;!,0)} [See list of classes](#)

Return the text from the nth bulleted item within the text block.

Syntax

textblockobject.GetNthBullet(*n*)

Parameters

n as *Integer*

Sequence number of the bullet.

Return values

A string containing the text from the nth bulleted item.

Freelance Graphics: GetObjectData method

{button ,AL(^H_GETOBJECTDATA_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

Return user-defined object data (persistent, string-valued name and value pair).

Syntax

drawobject.GetObjectData(*variablename*)

Parameters

variablename as String

Name of a user-defined variable.

Return values

String containing the value of the named variable.

Freelance Graphics: GetRGB method

{button ,AL(^H_GETRGB_METHOD_MEMDEF_RT;H_COLOR_CLASS;')0)} [See list of classes](#)

Return the RGB value of the color produced by the current values of the Red, Green, and Blue properties of a Color object.

Syntax

colorobject.GetRGB

Parameters

None

Return values

Long, RGB value.

Examples

```
Dim RGBValue as Long
RGBValue = MyColor.GetRGB()
```

Freelance Graphics: GetSelection method

```
{button ,AL(^H_GETSELECTION_METHOD_MEMDEF_RT;H_PAGESELECTION_CLASS;H_SELECTION_CLASS;',  
  0)} See list of classes
```

Returns the specified selected object from the set of currently selected objects.

Syntax

objectselection.GetSelection(*objectnumber*)

Parameters

objectnumber as Integer

The number of the object or page to be returned.

Return values

The DrawObject at the specified index in the selection.

Examples

```
'with at least two draw objects selected:  
Dim Rect2 as DrawObject  
Set Rect2 = Selection.GetSelection(2)
```


Freelance Graphics: GetSpeakerNoteMarkup method

{button ,AL(^H_GETSPEAKERNOTEMARKUP_METHOD_MEMDEF_RT;H_PAGE_CLASS;’,0)} [See list of classes](#)

Return the text from the speaker notes from a page, including all markups.

Syntax

pageobject.GetSpeakerNoteMarkup

Parameters

None

Return values

A string containing the text from the speaker notes for a page, and all markup attributes for that text.

Freelance Graphics: GotoNotes method

{button ,AL(^H_GOTONOTES_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;',0)} [See list of classes](#)

Switch window focus to Lotus Notes. (If Notes is not running, it will be launched.)

Syntax

applicationwindow.**GotoNotes**

Parameters

None

Return values

None

Freelance Graphics: GotoPage method

{button ,AL(^H_GOTOPAGE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Go to the page specified by a page number, page name, or object name.

Syntax

documentobject.GotoPage pageindicator

Parameters

pageindicator as Variant

Page number (integer), page name (string), or Page object.

Return values

None

Examples

`CurrentDocument.GotoPage (5)`

Freelance Graphics: Group method

{button ,AL(^H_GROUP_METHOD_MEMDEF_RT;H_SELECTION_CLASS;',0)} [See list of classes](#)

Group currently selected objects.

Syntax

selectionobject.Group

Parameters

None

Return values

A new instance of a grouped DrawObject.

Examples

```
Dim MyGroup as DrawObject  
Set MyGroup = Selection.Group()
```

Freelance Graphics: Import method

{button ,AL('H_IMPORT_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Import a file.

Syntax

`documentobject.Import(filename, includewithfile, converttopostscript, includetemplate, codepage)`

Parameters

filename as String

Name of the file to import.

includewithfile as Integer (Enumerated)

<u>Value</u>	<u>Description</u>
TRUE (-1)	Include the imported file with the presentation
FALSE (0)	Do not include the imported file (default)

converttopostscript as Integer (Enumerated)

<u>Value</u>	<u>Description</u>
TRUE (-1)	Convert the imported file to PostScript
FALSE (0)	Do not convert the imported file (default)

includetemplate as Integer (Enumerated)

<u>Value</u>	<u>Description</u>
TRUE (-1)	Include the imported template with the presentation
FALSE (0)	Do not include the imported template (default)

codepage as integer

Identifies a special code page for the import operation (defaults to 0).

Return values

The imported object.

Freelance Graphics: InsertCol method

{button ,AL(^H_INSERTCOL_METHOD_MEMDEF_RT;H_TABLE_CLASS;')} [See list of classes](#)

Insert a new column as the specified column in a table.

Syntax

tableobject.InsertCol(newcolumnnumber)

Parameters

newcolumnnumber as Integer

The column number of the new column in the table.

Return values

None

Freelance Graphics: InsertRow method

{button ,AL(^H_INSERTROW_METHOD_MEMDEF_RT;H_TABLE_CLASS;0)} [See list of classes](#)

Insert a new row as the specified row in a table.

Syntax

tableobject.InsertRow(newrownumber)

Parameters

newrownumber as Integer

The row number of the new row in the table.

Return values

None

Freelance Graphics: Insert method

{button ,AL(^H_INSERT_METHOD_MEMDEF_RT;H_PLACEMENTBLOCK_CLASS;!,0)} [See list of classes](#)

Insert the specified object in a placement ("Click here...") block.

Syntax

placementblockobject.Insert object

Parameters

object as DrawObject

The object to be inserted in the "Click here..." block.

Return values

None

Freelance Graphics: IsEmpty method

{button ,AL('H_ISEMPY_METHOD_MEMDEF_RT;H_COLORS_CLASS;H_DOCUMENTS_CLASS;H_OBJECTS_C
LASS;H_PAGES_CLASS;',0)} [See list of classes](#)

Test if a collection type object (Colors, Documents, Objects, or Pages) is empty (contains no items).

Syntax

objectcollection.IsEmpty

Parameters

None

Return values

Integer (Boolean)

<u>Value</u>	<u>Description</u>
TRUE	There are no items in the object
FALSE	There are one or more items in the object

Examples

```
If CurrentPage.Objects.IsEmpty() Then  
Print "There are no objects on this page"  
End If
```

Freelance Graphics: Item method

{button ,AL(^H_ITEM_METHOD_MEMDEF_RT;H_COLORS_CLASS;H_DOCUMENTS_CLASS;H_OBJECTS_CLASS;H_PAGES_CLASS;'0)} [See list of classes](#)

Return an item from a collection type object (Colors, Documents, Objects, or Pages).

Syntax

objectcollection.Item(*index*)

Parameters

index as Variant

The index of the item to return, or the name of the color.

Return values

An instance of the Color, Document, DrawObject, or Page class.

Examples

```
Dim MyPage2 as Page
```

```
Set MyPage2 = CurrentDocument.Pages.Item(2)
```

or

```
Dim MyBlueColor as Color
```

```
Dim MyGreenColor as Color
```

```
Set MyBlueColor = CurrentApplication.Colors.Item("Blue")
```

```
Set MyGreenColor = CurrentApplication.Colors.Item(35)
```

Freelance Graphics: LeaveEditMode

{button ,AL(^H_LEAVEEDITMODE_METHOD_MEMDEF_RT;H_TEXTBLOCK_CLASS;')} [See list of classes](#)

Leave edit mode for the text block (simulate clicking outside that block).

Syntax

textblockobject.LeaveEditMode

Parameters

None

Return values

None

Freelance Graphics: Maximize method

```
{button ,AL('H_MAXIMIZE_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;' ,0)} See list of classes
```

Maximize the Freelance application or document window.

Syntax

windowobject.Maximize

Parameters

None

Return values

None

Examples

```
CurrentApplicationWindow.Maximize
```

or

```
CurrentDocWindow.Maximize
```

Freelance Graphics: Minimize method

```
{button ,AL(^H_MINIMIZE_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;',  
  0)} See list of classes
```

Minimize the Freelance application or document window.

Syntax

windowobject.Minimize

Parameters

None

Return values

None

Examples

```
CurrentApplicationWindow.Minimize
```

or

```
CurrentDocWindow.Minimize
```

Freelance Graphics: Move method

{button ,AL('H_MOVE_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;H_PAGE_CLASS;','0)} [See list of classes](#)

Move an object to a new position on the page, or move a page to or before the specified page number.

Syntax

drawobject.**Move** *xtwips*, *ytwips*

or

pageobject.**Move** *pagenumber*, *insertbefore*

Parameters

xtwips as *Long*

Horizontal twips to move (positive moves to right, negative to left).

ytwips as *Long*

Vertical twips to move (positive moves up, negative down).

pagenumber as *Integer*

Page number of the new page, or page before which to insert the new page (see note).

insertbefore as *Integer (Enumerated)*

(Optional)

<u>Value</u>	<u>Description</u>
TRUE (-1)	Insert the page before the specified page number
FALSE (0)	(Default) Insert the page at the specified page number

Return values

None

Freelance Graphics: NearestColorFromRGB method

{button ,AL('H_NEARESTCOLORFROMRGB_METHOD_MEMDEF_RT;H_APPLICATION_CLASS;',0)} [See list of classes](#)

Return the Freelance Graphics palette color that is the closest match to the specified [RGB value](#).

Syntax

applicationobject.NearestColorFromRGB(*rgbvalue*)

Parameters

rgbvalue as Long

Any RGB value.

Return values

An instance of the Color class.

Examples

```
Dim MyColor as Color  
Set MyColor = CurrentApplication.NearestColorFromRGB(1598564)
```

Freelance Graphics: NewDocument method

{button ,AL('H_NEWDOCUMENT_METHOD_MEMDEF_RT;H_APPLICATION_CLASS;',0)} [See list of classes](#)

Create a new presentation by showing the user interface, with optional parameters supplying default values or selections.

Syntax

applicationobject.**NewDocument** *name, location, kind, mastername, masterlocation*

Parameters

Note The first three parameters are ignored. You must save a document to name it.

name as String

Optional file name.

location as Variant

Optional location of the file.

kind as String

Optional file type.

mastername as String

Optional content topic set name.

masterlocation as Variant

Optional content topic set location (must contain a string).

Return values

None

Freelance Graphics: OpenDocument method

{button ,AL(^H_OPENDOCUMENT_METHOD_MEMDEF_RT;H_APPLICATION_CLASS;!,0)} [See list of classes](#)

Open a presentation for editing.

Syntax

applicationobject.**OpenDocument** *name, location, kind, readonly, makevisible*

Parameters

name as *String*

File name of the presentation file to be opened.

location as *Variant*

Optional path for the file (must contain a string; defaults to the working directory).

kind as *String*

Optional file type (defaults to PRZ).

readonly as *Integer* (Enumerated)

Optional read-only flag.

<u>Value</u>	<u>Description</u>
TRUE (-1)	Read-only
FALSE (0)	Read-write (default)

makevisible as *Integer* (Enumerated)

Ignored and always treated as TRUE (this parameter is declared for compatibility with Open methods in other Lotus products).

<u>Value</u>	<u>Description</u>
TRUE (-1)	Visible (default)
FALSE (0)	Invisible (ignored)

Return values

None

Freelance Graphics: OpenPresForCopy method

{button ,AL(^H_OPENPRESFORCOPY_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Open a presentation from which pages can be copied (using the presentation browser).

Syntax

documentobject.OpenPresForCopy(*presentationname*)

Parameters

presentationname as String

(Optional) Name of the presentation file to open in the browser.

Return values

None

Freelance Graphics: PasteSelectedPages method

{button ,AL(^H_PASTESELECTEDPAGES_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')0)} [See list of classes](#)

Paste into the document the page(s) copied by the SelectPageForCopy method.

Syntax

documentobject.PasteSelectedPages(startingpagenumber)

Parameters

startingpagenumber as Integer

The starting page number at which the selected pages are to be pasted.

Return values

None

Freelance Graphics: PasteSpecial method

{button ,AL(^H_PASTESPECIAL_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Paste the Clipboard contents onto the page using a special format (bitmap or chart, for example).

Syntax

pageobject.PasteSpecial(clipboardformat)

Parameters

clipboardformat as String

Any of the Clipboard formats displayed in the Paste Special dialog, or *OLE object* for an OLE object on the Clipboard.

Return values

The pasted object (a new instance of the DrawObject class).

Freelance Graphics: Paste method

{button ,AL(^H_PASTE_METHOD_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

Paste the Clipboard contents onto the page or into the document.

Syntax

object.Paste(makevisible)

Parameters

makevisible as Integer (Enumerated)

Ignored and always treated as TRUE (this parameter is declared for compatibility with Paste methods in other Lotus products).

<u>Value</u>	<u>Description</u>
TRUE (-1)	Make the pasted page(s) visible (default)
FALSE (0)	Make the pasted page(s) invisible (ignored)

Return values

If pasting to a Page, the pasted object (a new instance of the DrawObject class); if pasting to a Document, no return value.

Freelance Graphics: Play method

{button ,AL(^H_PLAY_METHOD_MEMDEF_RT;H_MEDIA_CLASS;'0)} [See list of classes](#)

Play a media object (a sound or a movie).

Syntax

mediaobject.**Play**

Parameters

None

Return values

None

Freelance Graphics: PrintOut method

{button ,AL(^H_PRINTOUT_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')0} [See list of classes](#)

Print the presentation.

Syntax

documentobject.Print *frompage*, *topage*, *copies*

Parameters

frompage as *Integer*

Starting page number to print.

topage as *Integer*

Ending page number to print.

copies as *Integer*

Number of copies to print.

Return values

None

Usage

This method is identical to the Print method.

Freelance Graphics: Print method

{button ,AL(^H_PRINT_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Print the presentation.

Syntax

documentobject.Print *frompage*, *topage*, *copies*

Parameters

frompage as *Integer*

Starting page number to print.

topage as *Integer*

Ending page number to print.

copies as *Integer*

Number of copies to print.

Return values

None

Usage

This method is identical to the PrintOut method.

Freelance Graphics: PutIntoPlacementBlock method

{button ,AL('H_PUTINTOPLACEMENTBLOCK_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

Place an object in a "Click here..." block.

Syntax

drawobject.PutIntoPlacementBlock(*placementblockid*)

Parameters

placementblockid as Integer

An integer identifying the "Click here..." block.

Return values

None

Freelance Graphics: Quit method

{button ,AL(^H_QUIT_METHOD_MEMDEF_RT;H_APPLICATION_CLASS;')} [See list of classes](#)

Exit Freelance Graphics.

Syntax

applicationobject.Quit

Parameters

None

Return values

None

Freelance Graphics: RemoveFromSelection method

{button ,AL(^H_REMOVEFROMSELECTION_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;H_PAGESELECTIO
N_CLASS;H_SELECTION_CLASS;!,0)} [See list of classes](#)

Remove the specified object from the set of currently selected objects.

Syntax

objectcollection.**RemoveFromSelection** *object*

Parameters

object as *DrawObject*

The object you want removed from the selection.

Return values

None

Examples

```
Selection.RemoveFromSelection MyRect2
```

Freelance Graphics: Remove method

{button ,AL(^H_REMOVE_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;H_PAGE_CLASS;','0)} [See list of classes](#)

Delete a DrawObject or Page object.

Syntax

object.Remove

Parameters

None

Return values

None

Examples

CurrentSelection.Remove

Freelance Graphics: Replicate method

{button ,AL(^H_REPLICATE_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;H_PAGE_CLASS;','0)} [See list of classes](#)

Replicate an object.

Syntax

drawobject.Replicate

Parameters

None

Return values

A new instance of the replicated object.

Freelance Graphics: Restore method

{button ,AL(^H_RESTORE_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;' ,0)} [See list of classes](#)

Restore the Freelance Graphics application or document window.

Syntax

windowobject.Restore

Parameters

None

Return values

None

Freelance Graphics: RevertToStyle method

{button ,AL(^H_REVERTTOSTYLE_METHOD_MEMDEF_RT;H_BACKGROUND_CLASS;H_BORDER_CLASS;H_FONT_CLASS;H_LINESTYLE_CLASS;H_TEXTBLOCK_CLASS;','0)} [See list of classes](#)

Revert to the named style for a Background, Border, Font, LineStyle, or TextBlock object.

Note For this release, the RevertToStyle method has no effect any object class.

Syntax

object.RevertToStyle(attributename)

Parameters

attributename AS String

Optional input parameter.

Return values

None

Freelance Graphics: RGBtoColor method

{button ,AL(^H_RGBTOCOLOR_METHOD_MEMDEF_RT;H_COLORS_CLASS;','0)} [See list of classes](#)

Return the Color class object that matches the RGB value specified.

Syntax

colorobject.RGBtoColor(*colornumber*)

Parameters

colornumber as Long

An RGB value.

Return values

Color class object matching the RGB value specified.

Examples

```
Dim MyColor as Color
```

```
Set MyColor = CurrentApplication.Colors.RGBToColor(1598564)
```


Freelance Graphics: Rotate method

{button ,AL('H_ROTATE_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

Rotate the object a specified number of degrees around a specified anchor point.

Syntax

drawobject.Rotate *xanchor*, *yanchor*, *degrees*

Parameters

xanchor as Long

Horizontal coordinate of the anchor point in twips.

yanchor as Long

Vertical coordinate of the anchor point in twips.

degrees as Double

Degrees of rotation in a counterclockwise direction.

Return values

None

Freelance Graphics: RunDialog method

{button ,AL(^H_RUNDIALOG_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;';0)} [See list of classes](#)

Launch one of five hard-coded dialogs.

Syntax

documentobject.RunDialog(*dialogtype*, *dialogcaption*, *maintext*, *p4*, *p5*, *p6*, *p7*, *p8*, *p9*, *p10*, *p11*, *p12*)

Parameters and return values (by dialog type)

The first three parameters are the same for all five dialog types. The usage of parameters 4-12 and the return values vary by dialog type. The common parameters are described first, followed by the dialog-specific parameters and return values for each dialog type.

This method always returns a packed 32-bit (integer) value. In the descriptions of the return values, bit 0 is the least significant bit of the integer.

Common parameters

For all dialogs the first three parameters are always the same.

dialogtype as Integer

<u>Value</u>	<u>Description</u>
1	Two radio buttons, each with an associated spinner control
2	Seven check boxes
3	Four radio buttons, optional second text box
4	Three check boxes, optional second text box
5	List box with up to seven items

dialogcaption as String

The text to display in the caption (title bar) of the dialog.

maintext as String

Instructional text for a display-only text box in the dialog.

The use of the remaining parameters varies depending on the value of first parameter (*dialogtype*).

Note Parameters 4-10 are always type String, although for some dialog types they contain string representations of numerics. Parameters 11 and 12 are always type Integer.

Type 1 (two radio buttons, each with an associated spinner control) parameters

<u>Parameter</u>	<u>Description</u>
p4	String for first radio button
p5	String for second radio button
p6	String containing the number of the radio button selected initially
p7, p8	Unused
p9	String containing the first spinner minimum value
p10	String containing the first spinner maximum value
p11	Integer containing the second spinner minimum value
p12	Integer containing the second spinner maximum value

Type 1 return values

<u>Bit(s)</u>	<u>Description</u>
0-7	The value set by the spinner control for the selected radio button when the dialog was closed
8	The radio button selected on return:

- 0 = first
- 1 = second
- 9 How the user closed the box:
 - 0 = selected Cancel
 - 1 = selected OK
- 10-31 Unused

Type 2 (seven check boxes) parameters

<u>Parameter</u>	<u>Description</u>
p4-p10	Strings for check boxes 1-7; each parameter is the string to display with the corresponding check box
p11	Initial state of the check boxes; the low-order seven bits of the integer indicate if box is marked: <ul style="list-style-type: none"> 1 = Marked initially 0 = Not marked initially Bits 7-31 are unused
p12	Unused

Type 2 return values

<u>Bit(s)</u>	<u>Description</u>
0-6	Indicates which boxes were selected on return: <ul style="list-style-type: none"> 0 = not selected 1 = selected
7	Unused
8	How the user closed the box: <ul style="list-style-type: none"> 0 = selected Cancel 1 = selected OK
9-31	Unused

Type 3 (four radio buttons, optional second text box) parameters

<u>Parameter</u>	<u>Description</u>
p4-p7	Strings for radio buttons 1-4; each parameter is the string to display with the corresponding radio button
p8	String for optional second text block
p9-p12	Unused

Type 3 return values

<u>Bit(s)</u>	<u>Description</u>
0-2	A binary number indicating which radio button was selected on return (for example, 110 indicates that button three was selected)

- 3-7 Unused
- 8 How the user closed the box:
0 = selected Cancel
1 = selected OK
- 9-31 Unused

Type 4 (three check boxes, optional secondary text) parameters

<u>Parameter</u>	<u>Description</u>
p4-p6	Strings for check boxes 1-3; each parameter is the string to display with the corresponding check box
p7	String for optional secondary text block
p8-p10	Unused
p11	Initial state of the check boxes; the low-order three bits of the integer indicate if box is marked: 1 = Marked initially 0 = Not marked initially Bits 4-31 are unused
p12	Unused

Type 4 return value

<u>Bit(s)</u>	<u>Description</u>
0-2	Indicates which boxes were selected on return: 0 = not selected 1 = selected
3-7	Unused
8	How the user closed the box: 0 = selected Cancel 1 = selected OK
9-31	Unused

Type 5 (list box with up to seven items) parameters

<u>Parameter</u>	<u>Description</u>
p4-p10	Up to seven strings; each string is a separate entry in the list box
p11-p12	Unused

Type 5 return values

<u>Bit(s)</u>	<u>Description</u>
0-2	A binary number indicating which item in the list box was selected on return (for example, 110 indicates that item three was selected)

- 3-7 Unused
- 8 How the user closed the box:
 - 0 = selected Cancel
 - 1 = selected OK
- 9-31 Unused

Examples

```
' Example of posting Dialog 2 and interpreting results...
DIM PackedVal as Integer
PackedVal = CurrentDocument.RunDialog (2, "DialogTitle", "StaticText", _
"CheckBox 1", "CheckBox 2", "CheckBox 3", "CheckBox 4", "CheckBox 5", _
"CheckBox 6", "CheckBox 7", FALSE, 0)
For I = 1 TO 7
  IF PackedVal AND (2^(I-1)) THEN
    Print "CheckBox "+Str$(I)+" was enabled."
  ELSE
    Print "CheckBox "+Str$(I)+" was disabled."
  END IF
NEXT I
```

Freelance Graphics: SameColor method

{button ,AL(^H_SAMECOLOR_METHOD_MEMDEF_RT;H_COLOR_CLASS;','0)} [See list of classes](#)

Compare the color of two Color class objects to determine if they have the same [RGB value](#).

Syntax

colorobject1.SameColor(*colorobject2*)

Parameters

colorobject2 as Color

An object of class Color.

Return values

Integer (Boolean).

<u>Value</u>	<u>Description</u>
TRUE	Same color
FALSE	Different colors

Examples

```
If Color1.SameColor(Color2) Then
    Print "Color2 is identical to Color1"
End If
```

Freelance Graphics: SaveAs method

{button ,AL(^H_SAVEAS_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;0)} [See list of classes](#)

Save the current presentation in a new file.

Syntax

documentobject.**SaveAs**(*filename*, *location*, *type*, *backup*)

Parameters

filename as *String*

Optional file name.

location as *Variant*

Optional path in which to save the file.

type as *String*

(Optional) File type is ignored for this release. The actual type is always PRZ.

backup as *Integer*

Indicates if the file should be backed up.

<u>Value</u>	<u>Description</u>
TRUE (-1)	Back up the file
FALSE (0)	Do not back up the file

Return values

None

Freelance Graphics: Save method

{button ,AL(^H_SAVE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;'0)} [See list of classes](#)

Save the current presentation.

Syntax

documentobject.Save

Parameters

None

Return values

None

Freelance Graphics: SelectPageForCopy method

{button ,AL(^H_SELECTPAGEFORCOPY_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

Select for copying the specified page of a presentation opened with the OpenPresForCopy method.

Syntax

documentobject.SelectPageForCopy(*pagenumber*)

Parameters

pagenumber as *Integer*

The number of the page to select for copying.

Return values

None

Freelance Graphics: Select method

{button ,AL(^H_SELECT_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;H_PAGESELECTION_CLASS;H_SELECTION_CLASS;!,0)} [See list of classes](#)

Select the specified document, page, or object.

Syntax

selectionclasstype.**Select**(*object*)

Parameters

object as objecttype

A Document, DrawObject, or Page object.

Return values

None

Examples

```
Selection.Select MyRect
```

Freelance Graphics: SetObjectData method

{button ,AL(^H_SETOBJECTDATA_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

Set user-defined object data (persistent, string-valued name and value pair).

Syntax

drawobject.SetObjectData(variablename, variablevalue)

Parameters

variablename as String

The name for the variable.

variablevalue as String

The value for the variable.

Return values

None

Freelance Graphics: SetViewMode method

{button ,AL(^H_SETVIEWMODE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

Set the view mode.

Syntax

documentobject.SetViewMode(mode, pagenumber)

Parameters

mode as Variant

<u>Value</u>	<u>Description</u>
\$ViewDraw	Draw view
\$ViewOutliner	Outliner view
\$ViewSorter	Page Sorter view
\$ViewSlideShow	Screen Show view

pagenumber as Integer

Page number to view.

Return values

None

Freelance Graphics: Show method

{button ,AL(^H_SHOW_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

Run a screen show.

Syntax

documentobject.Show

Parameters

None

Return values

None

Freelance Graphics: StartGuidedTemplate method

{button ,AL(^H_STARTGUIDEDTEMPLATE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

Start using a content topic.

Syntax

documentobject.StartGuidedTemplate

Parameters

None

Return values

None

Freelance Graphics: StopGuidedTemplate method

{button ,AL(^H_STOPGUIDEDTEMPLATE_METHOD_MEMDEF_RT;H_DOCUMENT_CLASS;')0)} [See list of classes](#)

Stop using a content topic.

Syntax

documentobject.StopGuidedTemplate

Parameters

None

Return values

None

Freelance Graphics: StopPlay method

{button ,AL(^H_STOPPLAY_METHOD_MEMDEF_RT;H_MEDIA_CLASS;'0)} [See list of classes](#)

Stop playing the media.

Syntax

mediaobject.**StopPlay**

Parameters

None

Return values

None

Freelance Graphics: Stretch method

{button ,AL('H_STRETCH_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS';0)} [See list of classes](#)

Stretch the object.

Syntax

drawobject.Stretch(xanchor, yanchor, xstart, ystart, xfinish, yfinish, stretchmode)

Parameters

All x and y specifications are twips.

xanchor as Long

Horizontal coordinate of the point from which the object will be stretched.

yanchor as Long

Vertical coordinate of the point from which the object will be stretched.

xstart as Long

Horizontal starting coordinate of the stretch vector.

ystart as Long

Vertical starting coordinate of the stretch vector.

xfinish as Long

Horizontal ending coordinate of the stretch vector.

yfinish as Long

Vertical ending coordinate of the stretch vector.

stretchmode as Integer

Must be zero.

Return values

None

Freelance Graphics: Tile method

```
{button ,AL(^H_TILE_METHOD_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;','0)}
```

[See list of classes](#)

Tile all Freelance Graphics or document MDI windows.

Syntax

windowobject.Tile

Parameters

None

Return values

None

Examples

```
CurrentApplicationWindow.Tile
```

Freelance Graphics: Ungroup method

{button ,AL(^H_UNGROUP_METHOD_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

Ungroup a grouped object.

Syntax

drawobject.Ungroup

Parameters

None

Return values

None

Examples

```
Dim MySel as Selection
Set MySel = CurrentSelection
MySel.Group
MySel.Move(50,50)
MySel.Ungroup
```

Freelance Graphics: ActiveDocument property

{button ,AL(^H_ACTIVEDOCUMENT_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;','0)} [See list of classes](#)

(Read-only) Get the active presentation document.

Data type

Document

Syntax

set *documentobject* = *applicationobject*.**ActiveDocument**

Legal values

Any instance of the Document class.

Freelance Graphics: ActiveDocWindow property

{button ,AL(^H_ACTIVEDOCWINDOW_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;','0)} [See list of classes](#)

(Read-only) Get the document window of the active presentation.

Data type

DocWindow

Syntax

set *documentwindow* = *applicationobject*.**ActiveDocWindow**

Legal values

Any instance of the DocWindow class.

Freelance Graphics: ActivePage property

{button ,AL(^H_ACTIVEPAGE_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the current page, or if in Page Sorter view, the selected page.

Data type

Page

Syntax

set *page* = *documentobject.ActivePage*

set *documentobject.ActivePage* = *page*

Legal values

Any instance of the Page class.

Freelance Graphics: Active property

{button ,AL(^H_ACTIVE_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;!,0)} [See list of classes](#)

(Read-only) Determine if the document is the active presentation.

Data type

Integer (Boolean)

Syntax

value = *documentobject.Active*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Document is the active presentation
FALSE (0)	Document is not the active presentation

Freelance Graphics: ApplicationWindow property

{button ,AL(^H_APPLICATIONWINDOW_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;',0)} [See list of classes](#)

(Read-only) Get the Freelance Graphics application window.

Data type

ApplicationWindow

Syntax

set *applicationwindow* = *applicationobject*.**ApplicationWindow**

Legal values

Any instance of the ApplicationWindow class.

Freelance Graphics: Application property

{button ,AL(^H_APPLICATION_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;H_B
ASEOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the application.

Data type

Application

Document

String

Syntax

set *object* = *objecttype*.**Application**

Legal values

Any instance of the Application or Document class; for the BaseObject class, returns "CurrentApplication."

Freelance Graphics: Author property

{button ,AL(^H_AUTHOR_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;0)} [See list of classes](#)

(Read-write) Get or set the author of the presentation.

Data type

String

Syntax

authorname = *documentobject*.**Author**

documentobject.**Author** = *authorname*

Legal values

Any string value.

Freelance Graphics: AutoSaveInterval property

{button ,AL(^H_AUTOSAVEINTERVAL_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the number of minutes between automatic saves.

Note Used only if AutoSave property is TRUE (non-0).

Data type

Integer

Syntax

autosaveinterval = *preferencesobject*.AutoSaveInterval

preferencesobject.AutoSaveInterval = *autosaveinterval*

Legal values

Any integer.

Examples

```
CurrentApplication.Preferences.AutoSaveInterval = 5
```

Freelance Graphics: AutoSave property

{button ,AL(^H_AUTOSAVE_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Automatic save preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.AutoSave

preferencesobject.AutoSave = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Perform automatic saves
FALSE (0)	Do not perform automatic saves

Usage

The time interval for automatic saves is stored in the AutoSaveInterval property.

Examples

```
CurrentApplication.Preferences.AutoSave = True
```

Freelance Graphics: AutoTime property

{button ,AL(^H_AUTOTIME_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;'0)} [See list of classes](#)

(Read-write) During a screen show, controls whether the page displays for the time interval specified in the *pageobject.Delay* property.

Data type

Integer (Boolean)

Syntax

value = *pageobject.AutoTime*

pageobject.AutoTime = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Page displays for the default time interval specified in <i>pageobject.Delay</i>
FALSE (0)	Page remains displayed until the user selects another page

Freelance Graphics: BackColor property

{button ,AL(^H_BACKCOLOR_PROPERTY_MEMDEF_RT;H_BACKGROUND_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the background color of a drawn object.

Data type

Color

Syntax

set *color* = *backgroundobject*.BackColor

set *backgroundobject*.BackColor = *color*

Legal values

Any instance of the Color class.

Examples

```
Dim MyColor as Color
Set MyColor = MyRect1.Background.BackColor
Set MyRect2.Background.BackColor = MyColor
```

Freelance Graphics: Background property

{button ,AL(^H_BACKGROUND_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Background property of the object.

Data type

Background

Syntax

set *background* = *drawobject*.Background

set *drawobject*.Background = *background*

Legal values

Any instance of the Background class.

Usage

You can set a Background to an existing Background, or you can set the individual properties (BackColor, Color, and Pattern) of the Background class.

Examples

```
Dim MyBackgroundStyle as Background
Set MyBackgroundStyle = MyRect.Background
Set MyOval.Background = MyBackgroundStyle
```

Freelance Graphics: BackupDir property

{button ,AL(^H_BACKUPDIR_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;!,0)} [See list of classes](#)

(Read-write) Get or set the backup directory preference.

Data type

String

Syntax

backupdirectory = *preferencesobject*.BackupDir

preferencesobject.BackupDir = *backupdirectory*

Legal values

Any directory.

Examples

```
CurrentApplication.Preferences.BackUpDir = "c:\backup"
```


Freelance Graphics: BlackWhitePal property

{button ,AL(^H_BLACKWHITEPAL_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;0)} [See list of classes](#)

(Read-write) Get or set the Black-and-white or Color palette flag.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.BlackWhitePal

preferencesobject.BlackWhitePal = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Enable black-and-white palette
FALSE (0)	Disable black-and-white palette

Examples

```
CurrentApplication.Preferences.BlackWhitePal = True
```

Freelance Graphics: Blue property

{button ,AL(^H_BLUE_PROPERTY_MEMDEF_RT;H_COLOR_CLASS;','0)} [See list of classes](#)

(Read-only) Get the amount of blue in a Color object.

Data type

Integer

Syntax

blueamount = *colorobject*.Blue

Legal values

0 (no blue) to 255 (maximum blue).

Examples

```
Dim AmountOfBlue as Integer
AmountOfBlue = MyRect.Background.Color.Blue
```

Freelance Graphics: Bold property

{button ,AL(^H_BOLD_PROPERTY_MEMDEF_RT;H_FONT_CLASS;',0)} [See list of classes](#)

(Read-write) Determine if the font is bold.

Data type

Integer (Boolean)

Syntax

value = *fontobject*.**Bold**

fontobject.**Bold** = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Bold
FALSE (0)	Not bold

Examples

```
MyText.TextBlock.Font.Bold = True
```

Freelance Graphics: BorderDisplay property

{button ,AL(^H_BORDERDISPLAY_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the border display preference.

Data type

Integer (Variant)

Syntax

borderdisplay = *preferencesobject*.**BorderDisplay**

preferencesobject.**BorderDisplay**= *borderdisplay*

Legal values

<u>Value</u>	<u>Description</u>
\$BorderDispMargin	Display drawing area border (recommended)
\$BorderDispPrintableArea	Display printable area border
\$BorderDispNone	Display no border

Examples

`CurrentApplication.Preferences.BorderDisplay = $BorderDispPrintableArea`

Freelance Graphics: Border property

{button ,AL(^H_BORDER_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Border style for a drawing object.

Data type

Border

Syntax

set *border* = *drawobject.Border*

set *drawobject.Border* = *border*

Legal values

Any instance of the Border class.

Usage

You can set a Border to an existing Border, but more commonly you will set the individual properties (Color, Pattern, and Width) of the Border class.

Examples

```
Set MyRect2.Border = MyRect1.Border
```

or

```
MyRect1.Border.Width = $!tsBorderWidthThin
```

Freelance Graphics: BuildBullets property

{button ,AL(^H_BUILDBULLETS_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;';0)} [See list of classes](#)

(Read-write) Determines if a text block is to be made into a bullet build during a screen show.

Data type

Integer (Boolean)

Syntax

value = *drawobject*.Buildbullets

drawobject.BuildBullets = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Make text block into a bullet build during a screen show
FALSE (0)	Do not make text block into a bullet build during a screen show

Freelance Graphics: BulletProperties property

{button ,AL(^H_BULLETPROPERTIES_PROPERTY_MEMDEF_RT;H_TEXTBLOCK_CLASS;',0)} [See list of classes](#)

(Read-only) Get the bullet properties for a text block.

Data Type

BulletProperties

Syntax

set *bulletproperties* = *textblockobject*.**BulletProperties**

Legal values

Any instance of the BulletProperties class.

Usage

You can access or set the individual properties (Color, ShadowColor, ShadowDepth, and so on) of the BulletProperties class.

Freelance Graphics: Case property

{button ,AL(^H_CASE_PROPERTY_MEMDEF_RT;H_FONT_CLASS;',0)} [See list of classes](#)

(Read-write) Determine the font case (upper or lower).

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer

Syntax

fontcase = *fontobject*.**Case**

fontobject.**Case**= *fontcase*

Legal values

Any integer (always returns zero).

Freelance Graphics: Changed property

{button ,AL(^H_CHANGED_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

(Read-write) Determine if the document has changed.

Data type

Integer (Boolean)

Syntax

value = *documentobject.Changed*

documentobject.Changed = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Document changed
FALSE (0)	Document has not been changed

Freelance Graphics: Chart property

{button ,AL(^H_CHART_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;')} [See list of classes](#)

(Read-only) Get the Chart property for a drawing object.

Note The Freelance Graphics Chart class is derived from the DrawObject class, and has all the properties and methods of the LotusChart ChartBase class. For more information about the LotusChart ChartBase class, see the Help contents under LotusScript, LotusChart LotusScript Reference, By Category, Classes.

Data type

Chart

Syntax

set *chart* = *drawobject*.Chart

Legal values

Any instance of the Chart class.

Freelance Graphics: CodePage property

{button ,AL(^H_CODEPAGE_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;')} [See list of classes](#)

(Read-write) Get or set the code page preference.

Data type

Integer

Syntax

codepage = *preferencesobject*.CodePage

preferencesobject.Author = *codepage*

Legal values

<u>Value</u>	<u>Description</u>
0	System setting
437	U.S. English
850	Multilingual (Latin I)
852	Slavic (Latin II)
860	Portuguese
863	Canadian French
865	Norwegian

Freelance Graphics: ColCount property

{button ,AL(^H_COLCOUNT_PROPERTY_MEMDEF_RT;H_TABLE_CLASS;0)} [See list of classes](#)

(Read-only) Get the number of columns in a table.

Data type

Integer

Syntax

columns = *tableobject*.ColCount

Legal values

Any positive integer.

Freelance Graphics: Colors property

{button ,AL(^H_COLORS_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;',0)} [See list of classes](#)

(Read-only) Get the color palette for an application.

Data type

Colors

Syntax

set *colorpalette* = *colorsubject*.Colors

Legal values

Any instance of the Colors class.

Usage

You can use this property to retrieve palettecolors using the RGBToColor, Item, and GetNearestColor methods.

Examples

```
Dim MyPeachColor as Color
Set MyPeachColor = CurrentApplication.Colors.Item(178)
Set MyPeachColor = CurrentApplication.Colors.Item("peach")
Set MyPeachColor = CurrentApplication.Colors.RGBToColor(16764301)
```

Freelance Graphics: Color property

{button ,AL(^H_COLOR_PROPERTY_MEMDEF_RT;H_BACKGROUND_CLASS;H_BORDER_CLASS;H_BULLET_PROPERTIES_CLASS;H_LINESTYLE_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the color for a background, border, bullet, or line style.

Data type

Color

Syntax

set *color* = *object.Color*

set *object.Color* = *color*

Legal values

Any instance of the Color class.

Usage

You can set a Color to an existing Color, or you can use the RGBtoColor method to create a new Color from a combination of red, green, and blue. However, once you define a Color object in Freelance Graphics, you cannot modify its Red, Green, or Blue properties individually.

Examples

```
Dim MyColor as Color
Set MyColor = MyLine.LineStyle.Color
Set MyRect.Background.Color = MyColor
```

Freelance Graphics: Count property

{button ,AL(^H_COUNT_PROPERTY_MEMDEF_RT;H_COLORS_CLASS;H_DOCUMENTS_CLASS;H_OBJECTS_CLASS;H_PAGES_CLASS;'0)} [See list of classes](#)

(Read-only) Get the number of open presentations, the number of objects in a collection, the number of colors in the palette, or the number of pages in a collection.

Data type

Integer

Syntax

count = *object*.Count

Legal values

Any integer.

Examples

```
Print "There are " + str$(CurrentDocument.Pages.Count) + " pages in this document."
```

or

```
Dim NumObjs as Integer  
NumObjs = CurrentPage.Objects.Count
```

Freelance Graphics: CurrentPrinter property

{button ,AL(^H_CURRENTPRINTER_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;')0)} [See list of classes](#)

(Read-only) Get the name of the current printer for an application.

Data type

String

Syntax

printername = *applicationobject*.CurrentPrinter

Legal values

Any printer name.

Freelance Graphics: DefaultFilePath property

{button ,AL(^H_DEFAULTFILEPATH_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;','0)} [See list of classes](#)

(Read-only) Get the application's default file path.

Data type

String

Syntax

defaultfilepath = *applicationobject*.DefaultFilePath

Legal values

Any file path.

Examples

```
Print "The default file path is " + CurrentApplication.DefaultFilePath
```

Freelance Graphics: Delay property

{button ,AL(^H_DELAY_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;H_PAGE_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the time delay (in seconds) for a drawing object or page transition during a screen show. For a page, this delay is used only when *pageobject.AutoTime* is TRUE (non-0).

Data type

Integer

Syntax

secondsdelay = *object.Delay*

object .Delay = *secondsdelay*

Legal values

Any positive integer.

Freelance Graphics: Description property

```
{button ,AL(^H_DESCRIPTION_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;H_BASEOBJECT_CLASS;','0)}
```

[See list of classes](#)

(Read-write) Get or set the description of the presentation stored in the document, or get the class name of an object.

Note Read-only for the BaseObject class.

Data type

String

Syntax

description = *object*.**Description**

documentobject.**Description** = *description*

Legal values

Any string value (for the Document class) or the name of a class (for the BaseObject class).

Freelance Graphics: DimPrevious property

{button ,AL(^H_DIMPREVIOUS_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;0)} [See list of classes](#)

(Read-write) Determines if the previous items in a bulleted list are to be dimmed during a bullet build (during a screen show).

Data type

Integer (Boolean)

Syntax

value = *drawobject*.DimPrevious

drawobject.DimPrevious = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Dim the previous bullets
FALSE (0)	Do not dim the previous bullets

Freelance Graphics: DisplayCoords property

{button ,AL(^H_DISPLAYCOORDS_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;!,0)} [See list of classes](#)

(Read-write) Get or set the Display coordinates preference.

Data type

Integer (Boolean)

Syntax

displaycoordinates = *preferencesobject*.DisplayCoords

preferencesobject.DisplayCoords = *displaycoordinates*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Display the coordinates
FALSE (0)	Do not display the coordinates

Examples

```
CurrentApplication.Preferences.DisplayCoords = True
```

Freelance Graphics: DisplayDrawRuler property

{button ,AL(^H_DISPLAYDRAWRULER_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;!,0)} [See list of classes](#)

(Read-write) Get or set the Display drawing ruler preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.DisplayDrawRuler

preferencesobject.DisplayDrawRuler = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Display the drawing ruler
FALSE (0)	Do not display the drawing ruler

Examples

```
CurrentApplication.Preferences.DisplayDrawRuler = True
```

Freelance Graphics: DisplayGrid property

{button ,AL(^H_DISPLAYGRID_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Display grid preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.DisplayGrid

preferencesobject.DisplayGrid = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Display grid
FALSE (0)	Do not display grid

Examples

```
CurrentApplication.Preferences.DisplayGrid = True
```

Freelance Graphics: DisplayTextRuler property

{button ,AL(^H_DISPLAYTEXTRULER_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the Display text ruler preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.DisplayTextRuler

preferencesobject.DisplayTextRuler = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Display text ruler
FALSE (0)	Do not display text ruler

Examples

```
CurrentApplication.Preferences.DisplayTextRuler = True
```


Freelance Graphics: DocName property

{button ,AL(^H_DOCNAME_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the document (file) name for the presentation.

Data type

String

Syntax

documentname = *documentobject.DocName*

documentobject.DocName = *documentname*

Legal values

Any string value.

Freelance Graphics: Documents property

{button ,AL(^H_DOCUMENTS_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;',0)} [See list of classes](#)

(Read-only) Get the collection of open presentations.

Data type

Documents

Syntax

set *documentcollection* = *applicationobject*.Documents

Legal values

Any instance of the Documents class (a collection of Document objects).

Examples

```
ForAll doc in CurrentApplication.Documents
    Print doc.DocName
End ForAll
```

or

```
Dim num as Integer
num = CurrentApplication.Documents.Count
Print "There are " + str$(num) + " presentations open."
```

Freelance Graphics: Document property

{button ,AL(^H_DOCUMENT_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;'0)} [See list of classes](#)

(Read-only) Get the document containing the page.

Data type

Document

Syntax

set *documentobject* = *pageobject*.**Document**

Legal values

Any instance of the Document class.

Usage

You can set a Document object to an existing Document, or you can set the individual properties (Author, Description, Location, and so on) of the Document class.

Freelance Graphics: DocWindow property

{button ,AL(^H_DOCWINDOW_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the document window size (width and height).

Data type

DocWindow

Syntax

set *documentwindow* = *documentobject*.**DocWindow**

Legal values

Any instance of the DocWindow class.

Usage

You can set a DocWindow to an existing DocWindow, or you can set the individual properties (Height or Width) of the DocWindow class.

Freelance Graphics: DoubleUnderline property

{button ,AL(^H_DOUBLEUNDERLINE_PROPERTY_MEMDEF_RT;H_FONT_CLASS;','0)} [See list of classes](#)

(Read-write) Determine the Double underline (two lines under both words and spaces) font property.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

flag = *fontobject*.DoubleUnderline

fontobject.DoubleUnderline = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Double underline
FALSE (0)	No double underline

Freelance Graphics: Embedded property

{button ,AL(^H_EMBEDDED_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

(Read-only) Determine if the object is embedded.

Data type

Integer (Boolean)

Syntax

flag = *documentobject*.Embedded

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Embedded object
FALSE (0)	Not an embedded object

Freelance Graphics: Exclude property

{button ,AL(^H_EXCLUDE_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

(Read-write) Determine if the page is excluded from the screen show.

Data type

Integer (Boolean)

Syntax

value = *pageobject*.Exclude

pageobject.Exclude = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Excluded from screen show
FALSE (0)	Not excluded from screen show

Freelance Graphics: ExeName property

{button ,AL(^H_EXENAME_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;')0} [See list of classes](#)

(Read-write) Determine the executable to launch from an object during a screen show.

Data type

String

Syntax

executablename = *drawobject*.ExeName

drawobject.ExeName = *executablename*

Legal values

Any legal executable name.

Freelance Graphics: FirstIndent property

{button ,AL(^H_FIRSTINDENT_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the First indent property for the text (the number of [twips](#) the first line of text is to be indented).

Data type

Integer

Syntax

indentation = *textobject*.**FirstIndent**

textobject.**FirstIndent** = *indentation*

Legal values

Any integer.

Usage

Use the ParaIndent property to set the amount of space you want each paragraph indented.

Freelance Graphics: FontColor property

{button ,AL(^H_FONTCOLOR_PROPERTY_MEMDEF_RT;H_FONT_CLASS;';0)} [See list of classes](#)

(Read-write) Determine the color for a font.

Data type

Color

Syntax

set *color* = *fontobject*.FontColor

set *fontobject*.Color = *color*

Legal values

Any instance of the Color class.

Usage

You can set a FontColor to an existing Color, or you can use the Colors class to set a FontColor to a Color from the Freelance Graphics palette.

Examples

```
Set MyTextObject1.TextBlock.Font.FontColor = MyColor
```

```
Set MyTextObject2.TextBlock.Font.FontColor = CurrentApplication.Colors.Item(178)
```

Freelance Graphics: **FontName** property

{button ,AL(^H_FONTNAME_PROPERTY_MEMDEF_RT;H_FONT_CLASS;','0)} [See list of classes](#)

(Read-write) Determine the Font name for a Font class.

Data type

String

Syntax

fontname = *fontobject*.**FontName**

fontobject.**FontName** = *fontname*

Legal values

Any valid font name.

Examples

```
MyTextObject.TextBlock.Font.FontName = "Arial"
```

Freelance Graphics: FontUnit property

{button ,AL(^H_FONTUNIT_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;','0)} [See list of classes](#)

(Read-write) Determine the font units of measurement for an application. For Freelance Graphics, this is always points.

Data type

Variant

Syntax

fontunit = *applicationobject*.FontUnit

applicationobject.FontUnit = *fontunit*

Legal values

The only legal value is \$ltsScaleModePoint.

Freelance Graphics: Font property

{button ,AL(^H_FONT_PROPERTY_MEMDEF_RT;H_TEXTBLOCK_CLASS;H_TEXTPROPERTIES_CLASS;';0)} [See list of classes](#)

(Read-write) Determine the font property for the TextProperties or TextBlock class.

Data type

Font

Syntax

set *font* = *object*.Font

set *object*.Font = *font*

Legal values

Any instance of the Font class.

Usage

You can set a Font to an existing Font, or you can set the individual properties (Bold, Case, FontColor, and so on) of the Font class.

Examples

```
Dim MyFont as Font
Set MyFont = MyTextObject.TextBlock.Font
MyFont.Bold = True
Set MyTextObject2.TextBlock.Font = MyFont
```

Freelance Graphics: FullName property

{button ,AL(^H_FULLNAME_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the full application or document path name (the full path name includes the executable file name).

Data type

String

Syntax

fullname = *object*.FullName

Legal values

Any valid full path name.

Freelance Graphics: Green property

{button ,AL('H_GREEN_PROPERTY_MEMDEF_RT;H_COLOR_CLASS;','0)} [See list of classes](#)

(Read-only) Get the amount of green in a Color object.

Data type

Integer

Syntax

greenamount = *colorobject*.Green

colorobject.Green = *greenamount*

Legal values

0 (no green) to 255 (maximum green).

Examples

```
Dim AmountOfGreen as Integer
```

```
AmountOfGreen = MyRect.Background.Color.Green
```

Freelance Graphics: Height property

{button ,AL(^H_HEIGHT_PROPERTY_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the application window, document window, or drawing object height, in [twips](#).

Data type

Integer (AppWindow, DocWindow classes)

Long (DrawObject class)

Syntax

twipshigh = *object*.Height

Legal values

Any positive value.

Freelance Graphics: HorizontalAlignment property

{button ,AL(^H_HORIZONTALALIGNMENT_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;';0)} [See list of classes](#)

(Read-write) Get or set the horizontal alignment property for the text.

Data type

Variant (Enumerated)

Syntax

horizontalalignment = *textobject*.**HorizontalAlignment**

HorizontalAlignment = *horizontalalignment*

Legal values

<u>Value</u>	<u>Description</u>
\$ItsAlignmentLeft	Left aligned
\$ItsAlignmentRight	Right aligned
\$ItsAlignmentHorizCenter	Centered horizontally
\$ItsAlignmentJustify	Fit between margins

Freelance Graphics: ID property

{button ,AL(^H_ID_PROPERTY_MEMDEF_RT;H_PLACEMENTOBJECT_CLASS;0)} [See list of classes](#)

(Read-only) Get the ID for a "Click here..." block.

Data type

Integer

Syntax

idnumber = *placementblockobject.ID*

Legal values

Any "Click here..." block number.

Freelance Graphics: Image property

{button ,AL(^H_IMAGE_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only)Get the image properties for a drawing object. This property is valid only if *DrawImageRef.IsImage* is TRUE (non-0).

Data type

Image

Syntax

set *image* = *drawobject*.Image

Legal values

Any instance of the Image class.

Usage

You can set an Image to an existing Image, or you can set the individual properties (Brightness, Contrast, Sharpness, and so on) of the Image class.

Freelance Graphics: Interactive property

{button ,AL(^H_INTERACTIVE_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;'0)} [See list of classes](#)

(Read-only) Determine if the application is interactive. For Freelance Graphics, this value is always TRUE (non-0).

Data type

Integer (Boolean)

Syntax

value = *applicationobject*.Interactive

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Interactive
FALSE (0)	Not interactive

Freelance Graphics: IsChart property

{button ,AL(^H_ISCHART_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;!,0)} [See list of classes](#)

(Read-only) Determine if the object is a chart object.

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsChart

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Chart
FALSE (0)	Not a chart

Freelance Graphics: IsDraggable property

{button ,AL(^H_ISDRAGGABLE_PROPERTY_MEMDEF_RT;H_BASEOBJECT_CLASS;')0)} [See list of classes](#)

(Read-only) Determine if the object is draggable.

Note Not implemented for this release.

Data type

Integer (Boolean)

Syntax

value = *object*.IsDraggable

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Draggable
FALSE (0)	Not draggable

Freelance Graphics: IsGroup property

{button ,AL(^H_ISGROUP_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

(Read-only) Determine if the object is a grouped object.

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsGroup

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Grouped
FALSE (0)	Not grouped

Examples

```
If Selection.IsGroup Then
    Print "Selected DrawObject is a Grouped object."
    Selection.UnGroup
End If
```

Freelance Graphics: IsImage property

{button ,AL(^H_ISIMAGE_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;'0)} [See list of classes](#)

(Read-only) Determine if the object is an image object (a bitmap or graphics metafile, for example).

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsImage

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Image
FALSE (0)	Not an image

Freelance Graphics: IsMedia property

{button ,AL(^H_ISMEDIA_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;0)} [See list of classes](#)

(Read-only) TRUE if object is a media object (a movie or sound, for example).

Data type

Integer (Boolean)

Syntax

value = *drawobject.IsMedia*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Movie or sound object
FALSE (0)	Not a movie or sound object

Freelance Graphics: IsOleObj property

{button ,AL(^H_ISOLEOBJ_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;')} [See list of classes](#)

(Read-only) Determine if the object is an OLE object.

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsOleObj

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	OLE object
FALSE (0)	Not an OLE object

Freelance Graphics: IsOpen property

{button ,AL(^H_ISOPEN_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;',0)} [See list of classes](#)

(Read-only) Determine if the presentation document is currently open. In Freelance Graphics, this is always TRUE (non-0).

Data type

Integer (Boolean)

Syntax

value = *documentobject*.IsOpen

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Open
FALSE (0)	Not open

Freelance Graphics: IsPlacementBlock property

{button ,AL(^H_ISPLACEMENTBLOCK_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

(Read-only) Determine if the object is a "Click here..." block.

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsPlacementBlock

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	"Click here..." block
FALSE (0)	Not a "Click here..." block

Freelance Graphics: IsSelectable property

{button ,AL(^H_ISSELECTABLE_PROPERTY_MEMDEF_RT;H_BASEOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Determine if the object can be selected.

Data type

Integer (Boolean)

Syntax

value = *object*.IsSelectable

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Selectable (objects of the DrawObject and Page classes)
FALSE (0)	Not selectable (all other classes)

Freelance Graphics: IsTable property

{button ,AL(^H_ISTABLE_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Determine if the object is a table object.

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsTable

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Table
FALSE (0)	Not a table

Freelance Graphics: IsText property

{button ,AL(^H_IStEXT_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;0)} [See list of classes](#)

(Read-only) Determine if the object is a text object

Data type

Integer (Boolean)

Syntax

value = *drawobject*.IsText

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Text object
FALSE (0)	Not a text object

Freelance Graphics: IsValid property

{button ,AL(^H_ISVALID_PROPERTY_MEMDEF_RT;H_BASEOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Determine if the object is valid (still available).

Data type

Integer (Boolean)

Syntax

value = *object*.IsValid

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Valid
FALSE (0)	Not valid

Examples

The following example prints the value zero (FALSE):

```
MyRect.Cut  
Print MyRect.IsValid
```


Freelance Graphics: Italic property

{button ,AL(^H_ITALIC_PROPERTY_MEMDEF_RT;H_FONT_CLASS;';0)} [See list of classes](#)

(Read-write) Determine the italic attribute for the font.

Data type

Integer (Boolean)

Syntax

value = *font.Italic*

font.italic = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Italicized
FALSE (0)	Not italicized

Examples

```
MyTextObject.TextBlock.Font.Italic = True
```

Freelance Graphics: Layout property

{button ,AL(^H_LAYOUT_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

(Read-write) Determine the layout used by the page.

Data type

String

Syntax

layoutname = *pageobject*.Layout

pageobject.Layout = *layoutname*

Legal values

Any existing page layout name. Use a zero-length string ("") to refer to the [Blank Page] layout.

Tip Use the Freelance Graphics user interface to display all page layout names.

Freelance Graphics: Left property

{button ,AL(^H_LEFT_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

(Read-only) Get the left edge of the object, in [twips](#).

Data type

Long

Syntax

lefttwip = *drawobject*.Left

Legal values

Any positive value.

Freelance Graphics: LineLead property

{button ,AL(^H_LINELEAD_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;'0)} [See list of classes](#)

(Read-write) Get or set the line leading for the text (the number of points between line skips).

Data type

Integer

Syntax

lineleading = *textobject*.LineLead

textobject.LineLead= *lineleading*

Legal values

Any positive integer.

Freelance Graphics: LineStyle property

{button ,AL(^H_LINestyle_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the line style property.

Data type

LineStyle

Syntax

set *linestyleobject* = *drawobject.LineStyle*

set *drawobject.LineStyle* = *linestyleobject*

Legal values

You can set a LineStyle to an existing LineStyle, or you can set the individual properties (Color, Pattern, and Width) of the LineStyle class.

Examples

```
MyLine1.LineStyle.Width = $!tsBorderWidthThin
```

```
Set MyLine2.LineStyle = MyLine1.LineStyle
```

Freelance Graphics: Location property

{button ,AL(^H_LOCATION_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the default application or document path.

Data type

String

Syntax

path = *object*.Location

object.Location = *path*

Legal values

Any valid path name.

Freelance Graphics: Media property

{button ,AL(^H_MEDIA_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the Media class properties for an object. This property is valid only if *drawobject.isMedia* is TRUE (non-0).

Data type

Media

Syntax

set *media* = *drawobject.Media*

set *drawobject.Media* = *media*

Legal values

Any instance of the Media class.

Freelance Graphics: Name property

{button ,AL('H_NAME_PROPERTY_MEMDEF_RT;H_COLOR_CLASS;H_DRAWOBJECT_CLASS;H_PAGE_CLASS ;H_BASEOBJECT_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the page or drawing object name; or get the name of any other type of Freelance Graphics object.

Note Read-only except for the Page and DrawObject classes.

Data type

String

Syntax

name = *object.Name*

object.Name = *name*

Legal values

Any ASCII string value, but may not contain a semi-colon (;) or an equal sign (=).

Examples

```
Dim MyRect as DrawObject
Selection.name = "purple rectangle"
Set MyRect = CurrentPage.FindObject("purple rectangle")
CurrentPage.name = "My Title Page"
Print CurrentPage.Name
```


Freelance Graphics: Number property

{button ,AL(^H_NUMBER_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;0)} [See list of classes](#)

(Read-write) Get or set the page number.

Data type

Integer

Syntax

pagenumber = *pageobject*.**Number**

pageobject.**Number** = *pagenumber*

Legal values

From 1 to the number of pages in the presentation.

Usage

To move a page, set the page number to the new page number.

Freelance Graphics: Objects property

{button ,AL(^H_OBJECTS_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;!,0)} [See list of classes](#)

(Read-only) Get a collection of drawing objects on a page.

Data type

ObjectCollection

Syntax

set *objectcollection* = *pageobject*.Objects

Legal values

Any collection of objects.

Examples

```
ForAll objs in CurrentPage.Objects
    Print obj.name
End ForAll

Dim num as Integer
num = MyPage2.Objects.Count
Print "There are " + str$(num) + " objects on page 2."
```

Freelance Graphics: OffsetReplicate property

{button ,AL(^H_OFFSETREPLICATE_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Offset replicate preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.OffsetReplicate

preferencesobject.OffsetReplicate = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Offset copy from original
FALSE (0)	Place copy on top of original

Examples

```
CurrentApplication.Preferences.OffsetReplicate = True
```

Freelance Graphics: OleObject property

{button ,AL(^H_OLEOBJ_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

(Read-only) Get an OLE object.

Data type

Variant

Syntax

set *object* = *drawobject.OleObject*

Legal values

As defined by the application that created the object.

Freelance Graphics: Overstrike property

{button ,AL(^H_OVERSTRIKE_PROPERTY_MEMDEF_RT;H_FONT_CLASS;'0)} [See list of classes](#)

(Read-write) Determine the overstrike attribute for the font. The overstrike characters are typed over existing characters, usually indicating a deletion.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

value = *FontRef*.**Overstrike**

FontRef.**Overstrike** = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Overstrike on
FALSE (0)	Overstrike off

Freelance Graphics: PageSelection property

```
{button ,AL(^H_PAGSELECTION_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;',  
  0)} See list of classes
```

(Read-only) Get the set of pages currently selected in an application or document.

Data type

PageSelection

Syntax

set *pageselection* = *object*.PageSelection

Legal values

Any page selection.

Freelance Graphics: Pages property

{button ,AL(^H_PAGES_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;0)} [See list of classes](#)

(Read-only) Get pages in the presentation document.

Data type

PageCollection

Syntax

set *pagecollection* = *documentobject*.Pages

Legal values

Any collection of pages in the document.

Examples

```
ForAll page in CurrentDocument.Pages
    Print page.name
End ForAll
Dim num as Integer
num = MyDocument.Pages.Count
Print "There are " + str$(num) + " pages in MyDocument."
```

Freelance Graphics: PageTransitionDelay property

{button ,AL(^H_PAGETRANSITIONDELAY_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;';0)} [See list of classes](#)

(Read-write) Get or set the default delay for each page in the presentation when running a screen show.

Data type

Integer

Syntax

secondsdelay = *documentobject*.PageTransitionDelay

documentobject.PageTransitionDelay = *secondsdelay*

Legal values

Any positive integer.

Freelance Graphics: PageTransitionEffect property

{button ,AL('H_PAGETRANSITIONEFFECT_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the screen show page transition effect for each page in the presentation.

Data type

Variant

Syntax

pagetransitioneffect = *documentobject*.PageTransitionEffect

documentobject.PageTransitionEffect = *pagetransitioneffect*

Legal values

<u>Value</u>	<u>Description</u>
\$SSPageReplace	Instant replacement
\$SSPageBottom	From bottom
\$SSPageLeft	From left
\$SSPageRight	From right
\$SSPageBlinds	Blinds opening
\$SSPageLouvers	Louvers opening
\$SSPageBlocks	Block replacement
\$SSPageCenter	From center out
\$SSPageBoxIn	From outer box in
\$SSPageZigZag	Zigzag replacement
\$SSPageHorzIn	Horizontal lines
\$SSPageHVertIn	Vertical lines
\$SSPageTop	From top
\$SSPageBoxOut	From inner box out
\$SSPageHorizOut	Slide out horizontally
\$SSPageVertOut	Slide out vertically
\$SSPageFade	Fade to new
\$SSPageDiagL	Diagonal left
\$SSPageDiagR	Diagonal right
\$SSPagePanL	Pan left
\$SSPagePanR	Pan right
\$SSPageScrollT	Scroll from the top
\$SSPageScrollB	Scroll from the bottom
\$SSPageDraw	Draw new page
\$SSPageRain	Rain new page
\$SSPagePBrush	Paintbrush new page
\$SSPageShade	Shade new page
\$SSPageCurtain	Open curtain
\$SSPageBMPCol	Bitmap by colors

Freelance Graphics: Page property

{button ,AL(^H_PAGE_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the page on which the drawing object exists.

Data type

Page

Syntax

set *pageobject* = *drawobject*.Page

Legal values

Any instance of the Page class.

Freelance Graphics: ParaIndent property

{button ,AL(^H_PARAINDENT_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;!,0)} [See list of classes](#)

(Read-write) Get or set the Paragraph indent property for the text (the number of [twips](#) to indent the paragraph).

Data type

Integer

Syntax

indentation = *textobject*.**ParaIndent**

textobject.**ParaIndent** = *indentation*

Legal values

Any integer.

Usage

Use the FirstIndent property to set the amount of space you want first line of the paragraph indented.

Freelance Graphics: Parent property

{button ,AL(^H_PARENT_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;H_BASEOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Identify the parent application from which the current application or document was launched; or return the string "CurrentApplication" if the object is not a member of the Application or Document class.

Data type

Application

Document

Syntax

set parent = object.Parent

Legal values

Any instance of the Application or Document class, or the string "CurrentApplication."

Freelance Graphics: Path property

{button ,AL(^H_PATH_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the path of the application or document.

Data type

String

Syntax

path = *applicationobject*.Path

Legal values

Any legal path.

Freelance Graphics: Pattern property

{button ,AL(^H_PATTERN_PROPERTY_MEMDEF_RT;H_BACKGROUND_CLASS;H_BORDER_CLASS;H_LINESTYLE_CLASS;";0)} [See list of classes](#)

(Read-write) Get or set a background, border, or line style pattern.

Data type

Variant

Syntax

pattern = *object*.**Pattern**

object.**Pattern** = *pattern*

Legal values

There are separate values for each class. You can view the patterns in the Properties box of the Freelance Graphics user interface for the appropriate object type, and match them with the names below.

Border class

<u>Value</u>	<u>Description</u>
\$ItsBorderPatternNone	
\$ItsBorderPatternSolid	
\$ItsBorderPatternDashDot	
\$ItsBorderPatternDashDotDot	
\$ItsBorderPatternLongDash	
\$ItsBorderPatternDashed	same as \$ItsBorderPatternDot
\$ItsBorderPatternDot	same as \$ItsBorderPatternDashed

LineStyle class

<u>Value</u>	<u>Description</u>
\$ItsLineStyleNone	
\$ItsLineStyleSolid	
\$ItsLineStyleDashDot	
\$ItsLineStyleDashDotDot	
\$ItsLineStyleLongDash	
\$ItsLineStyleMediumDash	same as \$ItsLineStyleDot
\$ItsLineStyleDot	same as \$ItsLineStyleMediumDash

Background values

<u>Value</u>	<u>Description</u>
\$ItsFillNone	
\$ItsFillSolid	
\$ItsFillGray1	
\$ItsFillGray2	
\$ItsFillGray3	
\$ItsFillGray4	
\$ItsFillGray5	
\$ItsFillGray6	
\$ItsFillGray7	
\$ItsFillGray8	
\$ItsFillGray9	same as \$ItsFillGray10
\$ItsFillGray10	same as \$ItsFillGray9

\$ltsFillLeftDiagonal
\$ltsFillRightDiagonal
\$ltsFillDiagonalHatch
\$ltsFillHorizontal
\$ltsFillVertical
\$ltsFillRegularHatch
\$ltsFillLeftRightGrad
\$ltsFillBottomTopGrad
\$ltsFillNeToSwGrad
\$ltsFillNwToSeGrad
\$ltsFillCenterBoxGrad
\$ltsFillLowBoxGrad
\$ltsFillCenterCircleGrad
\$ltsFillLowCircleGrad
\$ltsFillNeToSwDiagonalStripGrad
\$ltsFillNwToSeDiagonalStripGrad

Examples

```
MyRect.Border.Pattern = $ltsBorderPatternDashed  
MyRect.Background.Pattern = $ltsFillGray8  
MyLine.LineStyle.Pattern = $ltsLineStyleDashDot
```

Freelance Graphics: PlacementBlock property

{button ,AL(^H_PLACEMENTBLOCK_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;')0)} [See list of classes](#)

(Read-only) Get the "Click here..." block properties for a drawing object. These properties are valid only when *drawobject.isPlacementBlock* is TRUE (non-0).

Data type

PlacementBlock

Syntax

set *placementblock* = *placementblockobject.PlacementBlock*

Legal values

Any instance of the PlacementBlock class.

Freelance Graphics: PlayPriority property

{button ,AL(^H_PLAYPRIORITY_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the priority with which the object transitions onto the page during a screen show.

Data type

Integer

Syntax

playpriority = *drawobject*.**PlayPriority**

drawobject.**PlayPriority** = *playpriority*

Legal values

1=first object, 2=second, and so on

Freelance Graphics: Preferences property

{button ,AL(^H_PREFERENCES_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;!,0)} [See list of classes](#)

(Read-only) Get the preferences for the application.

Data type

Preferences

Syntax

set *preferences* = *applicationobject.Preferences*

Legal values

Any instance of the Preferences class.

Usage

You can get the set of preferences in a Preferences object, or you can get or set the individual properties (AutoSave, AutoSaveInterval, BackupDir, and so on) of the Preferences class.

Examples

```
CurrentApplication.Preferences.AutoSave = True
```

Freelance Graphics: Priority property

{button ,AL(^H_PRIORITY_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the drawing priority of a drawing object.

Data type

Integer

Syntax

priority = *drawobject*.Priority

drawobject.Priority = *priority*

Legal values

1=first, 2=second, and so on

Freelance Graphics: PromptText property

{button ,AL(^H_PROMPTTEXT_PROPERTY_MEMDEF_RT;H_PLACEMENTBLOCK_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the prompt text for a "Click here..." block.

Data type

String

Syntax

prompttext = *placementblockobject*.**PromptText**

placementblockobject.**PromptText** = *prompttext*

Legal values

Any string value.

Freelance Graphics: ReadOnly property

{button ,AL(^H_READONLY_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;'0)} [See list of classes](#)

(Read-write) Determine if the presentation is read-only.

Data type

Integer (Boolean)

Syntax

value = *documentobject.ReadOnly*

documentobject.ReadOnly = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Read-only
FALSE (0)	Read-write

Freelance Graphics: Red property

{button ,AL('H_RED_PROPERTY_MEMDEF_RT;H_COLOR_CLASS';0)} [See list of classes](#)

(Read-only) Get the amount of red in a Color object.

Data type

Integer

Syntax

redamount = *colorobject*.Red

Legal values

0 (no red) to 255 (maximum red).

Examples

```
Dim AmountOfRed as Integer  
AmountOfRed = MyRect.Background.Color.Red
```

Freelance Graphics: RemoveMedia property

{button ,AL(^H_REMOVEMEDIA_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;0)} [See list of classes](#)

(Read-only) Determine if the object is on removable media.

Data type

Integer (Boolean)

Syntax

removablemedia = *drawobject*.RemoveMedia

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	On removable media
FALSE (0)	Not on removable media

Freelance Graphics: RightIndent property

{button ,AL(^H_RIGHTINDENT_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;')0)} [See list of classes](#)

(Read-write) Get or set the Right indent property for the text (the number of twips to indent the text).

Data type

Integer

Syntax

indentation = *textobject*.RightIndent

textobject.RightIndent = *indentation*

Legal values

Any integer.

Freelance Graphics: RowCount property

{button ,AL(^H_ROWCOUNT_PROPERTY_MEMDEF_RT;H_TABLE_CLASS;')} [See list of classes](#)

(Read-only) Get the number of rows in a table.

Data type

Integer

Syntax

rows = *tableobject*.RowCount

Legal values

Any positive integer.

Freelance Graphics: ScanSpeed property

{button ,AL(^H_SCANSPEED_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the Scan speed preference (the number of seconds delay between displaying images when scanning in the browser).

Data type

Long

Syntax

secondsdelay = *preferencesobject.ScanSpeed*

preferencesobject.ScanSpeed = *secondsdelay*

Legal values

Any value between .1 and 100.0.

Examples

`CurrentApplication.Preferences.ScanSpeed = 36.5`

Freelance Graphics: SelectedObjects property

{button ,AL(^H_SELECTEDOBJECTS_PROPERTY_MEMDEF_RT;H_SELECTION_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the collection of currently selected objects.

Data type

Objects

Syntax

set *selection* = *objectsobject.SelectedObjects*

set *objectsobject.SelectedObjects* = *selection*

Legal values

Any selection of objects.

Usage

This property is useful for saving and then restoring a selection.

Examples

```
Dim MySelection as Objects 'clears and restores selection
Set MySelection = Selection.SelectedObjects
Selection.ClearSelection
Set Selection.SelectedObjects = MySelection
```

Freelance Graphics: SelectionCount property

{button ,AL(^H_SELECTIONCOUNT_PROPERTY_MEMDEF_RT;H_PAGESELECTION_CLASS;H_SELECTION_CL
ASS;';0)} [See list of classes](#)

(Read-only) Get the number of selected pages or objects.

Data type

Integer

Syntax

selectioncount = *object*.SelectionCount

Legal values

Any integer.

Examples

```
Dim num as Integer
num = Selection.SelectionCount
Print "There are " + str$(num) + " selected objects."
```

Freelance Graphics: Selection property

{button ,AL(^H_SELECTION_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;H_DOCUMENT_CLASS;H_PAGE_CLASS;0)} [See list of classes](#)

(Read-only) Get a selection of objects in the application or document, or on the currently active page.

Data type

Selection

Syntax

set *selection* = *object*.**Selection**

Legal values

Any selection of objects.

Examples

```
Dim MySelection as Selection
Set MySelection = CurrentPage.Selection
MySelection.Group
```

Freelance Graphics: ShadowColor property

{button ,AL(^H_SHADOWCOLOR_PROPERTY_MEMDEF_RT;H_BULLET_PROPERTIES_CLASS;H_TEXTPROPERTIES_CLASS;'0)} [See list of classes](#)

(Read-write) Get or set the Shadow color for a bullet or text.

Data type

Color

Syntax

set *color* = *shadowobject*.ShadowColor

set *shadowobject*.ShadowColor = *color*

Legal values

Any instance of the Color class.

Freelance Graphics: ShadowDepth property

{button ,AL(^H_SHADOWDEPTH_PROPERTY_MEMDEF_RT;H_BULLET_PROPERTIES_CLASS;H_TEXTPROPERTIES_CLASS;0)} [See list of classes](#)

(Read-write) Get or set the Shadow depth for a bullet or text.

Data type

Variant (Enumerated)

Syntax

shadowdepth = *shadowobject*.ShadowDepth

shadowobject.ShadowDepth = *shadowdepth*

Legal values

<u>Value</u>	<u>Description</u>
\$ItsShadowDepthShallow	Short shadow
\$ItsShadowDepthNormal	Normal shadow
\$ItsShadowDepthDeep	Long shadow

Freelance Graphics: ShadowDirection property

{button ,AL(^H_SHADOWDIRECTION_PROPERTY_MEMDEF_RT;H_BULLET_PROPERTIES_CLASS;H_TEXTPROPERTIES_CLASS;'0)} [See list of classes](#)

(Read-write) Get or set the Shadow direction for a bullet or text.

Data type

Variant (Enumerated)

Syntax

shadowdirection = *shadowobject*.ShadowDirection

shadowobject.ShadowDirection = *shadowdirection*

Legal values

<u>Value</u>	<u>Description</u>
\$ItsShadowNone	No shadow
\$ItsShadowBottomRight	Shadow on bottom right
\$ItsShadowBottomLeft	Shadow on bottom left
\$ItsShadowTopRight	Shadow on top right
\$ItsShadowTopLeft	Shadow on top left

Freelance Graphics: Shadow property

{button ,AL(^H_SHADOW_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;',0)} [See list of classes](#)

(Read-write) Determine if the text has a shadow.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

flag = *textobject*.Shadow

textobject.Shadow = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Text shadow
FALSE (0)	No text shadow

Freelance Graphics: Size property

{button ,AL(^H_SIZE_PROPERTY_MEMDEF_RT;H_BULLET_PROPERTIES_CLASS;H_FONT_CLASS;','0)} [See list of classes](#)

(Read-write) Determine the font or bullet size (in points).

Data type

Double

Syntax

points = *object*.**Size**

object.**Size** = *points*

Legal values

Any positive integer.

Freelance Graphics: SkipWelcome property

{button ,AL(^H_SKIPWELCOME_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the Skip welcome preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.SkipWelcome

preferencesobject.SkipWelcome = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Skip Welcome on entry
FALSE (0)	Display Welcome on entry

Examples

```
CurrentApplication.Preferences.SkipWelcome = True
```

Freelance Graphics: SmallCaps property

{button ,AL(^H_SMALLCAPS_PROPERTY_MEMDEF_RT;H_FONT_CLASS;','0)} [See list of classes](#)

(Read-write) Determine if the font displays as Small caps (smaller point size, but all uppercase).

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

value = *fontobject*.SmallCaps

fontobject.SmallCaps = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Small caps
FALSE (0)	No small caps

Freelance Graphics: SmartLook property

{button ,AL(^H_SMARTLOOK_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;!,0)} [See list of classes](#)

(Read-write) Get or set the SmartMaster look for the current presentation.

Data type

String

Syntax

smartlook = *documentobject*.SmartLook

documentobject.SmartLook = *smartlook*

Legal values

Any existing SmartMaster look.

Note You can use the LotusScript Dir command to list the *.MAS files in the \SMASTERS\FLG directory.

Freelance Graphics: SnapToGrid property

{button ,AL(^H_SNAPTOGRID_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Snap to grid preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.SnapToGrid

preferencesobject.SnapToGrid = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Snap objects to grid
FALSE (0)	Do not snap objects to grid

Examples

```
CurrentApplication.Preferences.SnapToGrid = True
```

Freelance Graphics: Sound property

{button ,AL(^H_SOUND_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;')} [See list of classes](#)

(Read-write) Get or set the file name for a sound associated with the page.

Data type

String

Syntax

soundfilename = *pageobject*.**Sound**

pageobject.**Sound** = *soundfilename*

Legal values

Any existing sound file name.

Freelance Graphics: SpeakerNoteText property

{button ,AL(^H_SPEAKERNOTETEXT_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the speaker note text displayed on the page.

Data type

String

Syntax

speakernotetext = *pageobject*.**SpeakerNoteText**

pageobject.**SpeakerNoteText** = *speakernotetext*

Legal values

Any string value.

Freelance Graphics: StartNumber property

{button ,AL(^H_STARTNUMBER_PROPERTY_MEMDEF_RT;H_BULLET_PROPERTIES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the starting number for numbered bullets.

Data type

Integer

Syntax

startnumber = *bulletobject*.**StartNumber**

bulletobject.**StartNumber** = *startnumber*

Legal values

Any integer.

Freelance Graphics: StartupView property

{button ,AL(^H_STARTUPVIEW_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Startup view preference.

Data type

Variant (Enumerated)

Syntax

startupview = *preferencesobject.StartupView*

preferencesobject.StartupView = *startupview*

Legal values

<u>Value</u>	<u>Description</u>
\$ViewDraw	Current Page view
\$ViewOutliner	Outliner view
\$ViewSorter	Page Sorter view
\$ViewSlideShow	Screen Show view

Examples

`CurrentApplication.Preferences.StartupView = $ViewSorter`

Freelance Graphics: StrikeThrough property

{button ,AL(^H_STRIKETHROUGH_PROPERTY_MEMDEF_RT;H_FONT_CLASS;’,0)} [See list of classes](#)

(Read-write) Determine the strikethrough attribute for the font. Strikethrough is a horizontal line that prints through the middle of existing characters.

Data type

Integer (Boolean)

Syntax

value = *fontobject*.StrikeThrough

fontobject.StrikeThrough = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Strikethrough on (for example)
FALSE (0)	Strikethrough off

Examples

```
MyTextObject.TextBlock.Font.StrikeThrough = True
```

Freelance Graphics: Style property

{button ,AL(^H_STYLE_PROPERTY_MEMDEF_RT;H_BULLET_PROPERTIES_CLASS;0)} [See list of classes](#)

(Read-write) Determine the Bullet style.

Data type

Variant (Enumerated)

Syntax

bulletstyle = *bulletpropertiesobject.Style*

bulletpropertiesobject.Style = *bulletstyle*

Legal values

<u>Value</u>	<u>Description</u>
\$ItsBulletNone	None
\$ItsBulletSmallLetters	
\$ItsBulletCapLetters	
\$ItsBulletRomanNums	
\$ItsBulletDecimalNums	
\$ItsBulletSmallDot	
\$ItsBulletLargeDot	
\$ItsBulletSmallSquare	
\$ItsBulletLargeSquare	
\$flwBulletDash	
\$ItsBulletArrowhead	
\$ItsBulletCheck	
\$ItsBulletX	
\$ItsBulletStar	
\$ItsBulletPlus	
\$flwBulletCurvedArrowhead	
\$ItsbulletRndSquare	
\$ItsBulletSmallDiamond	
\$ItsBulletLargeDiamond	
\$\$ItsBulletSmallArrowhead	

Freelance Graphics: SubScript property

{button ,AL(^H_SUBSCRIPT_PROPERTY_MEMDEF_RT;H_FONT_CLASS;'0)} [See list of classes](#)

(Read-write) Determine the subscript attribute for the font. Subscripted characters are placed just below the surrounding text.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

value = *fontobject*.SubScript

fontobject.SubScript = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Subscript on
FALSE (0)	Subscript off

Freelance Graphics: SuperScript property

{button ,AL(^H_SUPERSCRIPT_PROPERTY_MEMDEF_RT;H_FONT_CLASS;'0)} [See list of classes](#)

(Read-write) Determine the superscript attribute for the font. Superscripted characters are placed just above the surrounding text.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

value = *fontobject*.SuperScript

fontobject.SuperScript = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Superscript on
FALSE (0)	Superscript off

Freelance Graphics: Table property

{button ,AL(^H_TABLE_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;!,0)} [See list of classes](#)

(Read-write) Get table properties for a drawing object. These properties are valid only when *drawobject.IsTable* is TRUE (non-0).

Data type

Table

Syntax

set table = drawobject.Table

Legal values

Any instance of the Table class.

Freelance Graphics: TemplateDir property

{button ,AL(^H_TEMPLATEDIR_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;')0)} [See list of classes](#)

(Read-write) Get or set the content topic directory preference.

Data type

String

Syntax

templatedirectory = *preferencesobject*.**TemplateDir**

preferencesobject.**TemplateDir** = *templatedirectory*

Legal values

Any directory.

Examples

```
CurrentApplication.Preferences.TemplateDir = "c:\lotus\smasters\flg"
```


Freelance Graphics: TemplatePageCount property

{button ,AL(^H_TEMPLATEPAGECOUNT_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;')} [See list of classes](#)

(Read-only) Get the content topic page count.

Data type

Integer

Syntax

pagecount = *documentobject*.**TemplatePageCount**

Legal values

Any integer.

Freelance Graphics: TextBlock property

{button ,AL(^H_TEXTBLOCK_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;','0)} [See list of classes](#)

(Read-only) Get the text block properties for a drawing object. These properties are valid only if *drawobject.isText* is TRUE (non-0).

Data type

TextBlock

Syntax

set *textblock* = *drawobject*.**TextBlock**

Legal values

Any instance of the TextBlock class.

Freelance Graphics: TextProperties property

{button ,AL(^H_TEXTPROPERTIES_PROPERTY_MEMDEF_RT;H_TEXTBLOCK_CLASS;','0)} [See list of classes](#)

(Read-only) Get the text properties for a text block.

Data type

TextProperties

Syntax

set *textproperties* = *textblockobject*.**TextProperties**

Legal values

Any instance of the TextProperties class.

Usage

You can get a set of TextProperties, or you can get or set the individual properties (FirstIndent, Font, HorizontalAlignment, and so on) of the TextProperties class.

Freelance Graphics: TextTightness property

{button ,AL(^H_TEXTTIGHTNESS_PROPERTY_MEMDEF_RT;H_FONT_CLASS;',0)} [See list of classes](#)

(Read-write) Determine the Text tightness for the font.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer

Syntax

tightness = *fontobject*.TextTightness

fontobject.TextTightness = *tightness*

Legal values

Any integer (always returns zero).

Freelance Graphics: Text property

{button ,AL(^H_TEXT_PROPERTY_MEMDEF_RT;H_TEXTBLOCK_CLASS;'0)} [See list of classes](#)

(Read-write) Get or set the text contained in a text block.

Data type

String

Syntax

text = *textblockobject*.Text

textblockobject.Text = *text*

Legal values

Any string value.

Examples

```
MyTextObject.TextBlock.Text = "This text will appear in the text block"
```

Freelance Graphics: Title property

{button ,AL(^H_TITLE_PROPERTY_MEMDEF_RT;H_PAGE_CLASS;',0)} [See list of classes](#)

(Read-only) Get the title text block on a page.

Data type

DrawObject

Syntax

set *titleobject* = *pageobject*.Title

Legal values

Any instance of a page title text block.

Usage

You cannot set this property directly, but you can change the page title by changing the underlying Text object (*pageobject*.Title.Text).

Freelance Graphics: Top property

{button ,AL(^H_TOP_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;',0)} [See list of classes](#)

(Read-only) Get the top edge of an object, in [twips](#).

Data type

Long

Syntax

toptwip = *DrawObject*.Top

Legal values

Any positive value.

Freelance Graphics: TransitionEffect property

```
{button ,AL(^H_TRANSITIONEFFECT_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;H_PAGE_CLASS;',0)}
```

[See list of classes](#)

(Read-write) Get or set the transition effect for an object or page during a screen show.

Data type

Variant

Syntax

transitioneffect = *object*.TransitionEffect

object.TransitionEffect = *transitioneffect*

Legal values

The legal values vary by class.

Legal values for the Page class

<u>Value</u>	<u>Description</u>
\$SSPageReplace	Instant replacement
\$SSPageBottom	From bottom
\$SSPageLeft	From left
\$SSPageRight	From right
\$SSPageBlinds	Blinds opening
\$SSPageLouvers	Louvers opening
\$SSPageBlocks	Block replacement
\$SSPageCenter	From center out
\$SSPageBoxIn	From outer box in
\$SSPageZigZag	Zigzag replacement
\$SSPageHorzIn	Horizontal lines
\$SSPageHVertIn	Vertical lines
\$SSPageTop	From top
\$SSPageBoxOut	From inner box out
\$SSPageHorizOut	Slide out horizontally
\$SSPageVertOut	Slide out vertically
\$SSPageFade	Fade to new
\$SSPageDiagL	Diagonal left
\$SSPageDiagR	Diagonal right
\$SSPagePanL	Pan left
\$SSPagePanR	Pan right
\$SSPageScrollT	Scroll from the top
\$SSPageScrollB	Scroll from the bottom
\$SSPageDraw	Draw new page
\$SSPageRain	Rain new page
\$SSPagePBrush	Paintbrush new page
\$SSPageShade	Shade new page
\$SSPageCurtain	Open curtain
\$SSPageBMPCol	Bitmap by colors

Legal values for the DrawObject class

Value	Description
\$SSObjReplace	Instant replacement
\$SSObjBottom	From bottom
\$SSObjLeft	From left
\$SSObjRight	From right
\$SSObjBlinds	Blinds opening
\$SSObjLouvers	Louvers opening
\$SSObjBlocks	Block replacement
\$SSObjCenter	From center out
\$SSObjBoxIn	From outer box in
\$SSObjZigZag	Zigzag replacement
\$SSObjHorzIn	Horizontal lines
\$SSObjVertIn	Vertical lines
\$SSObjTop	From top
\$SSObjBoxOut	From inner box out
\$SSObjHorzOut	Slide out horizontally
\$SSObjVertOut	Slide out vertically
\$SSObjFade	Fade to new
\$SSObjDiagL	Diagonal left
\$SSObjDiagR	Diagonal right
\$SSObjScrollL	Pan left
\$SSObjScrollR	Pan right
\$SSObjScrollT	Scroll from the top
\$SSObjScrollB	Scroll from the bottom
\$SSObjDraw	Draw new page
\$SSObjRain	Rain new page
\$SSObjPBrush	Paintbrush new page
\$SSObjShade	Shade new page
\$SSObjCurtain	Open curtain
\$SSObjBMPCol	Bitmap by colors

Freelance Graphics: Type property

{button ,AL(^H_TYPE_PROPERTY_MEMDEF_RT;H_PLACEMENTBLOCK_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the "Click here..." block type.

Data type

Variant

Syntax

type = *placementblockobject.Type*

placementblockobject.Type = *type*

Legal values

<u>Value</u>	<u>Description</u>
pbTypeText	Click here to enter text
pbTypeSymbol	Click here to add a symbol
pbTypeChart	Click here to create a chart
pbTypeOrgChart	Click here to create an organization chart
pbTypeTable	Click here to create a table
pbTypeButton	Click here to add a button

Freelance Graphics: Underline property

{button ,AL(^H_UNDERLINE_PROPERTY_MEMDEF_RT;H_FONT_CLASS;'0)} [See list of classes](#)

(Read-write) Determine the Underline attribute for the font. This underline style is a solid line under both words and spaces.

Data type

Integer (Boolean)

Syntax

value = *fontobject*.Underline

fontobject.Underline = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Underline on
FALSE (0)	Underline off

Examples

```
MyText.TextBlock.Font.Underline = True
```

Freelance Graphics: UndoEnabled property

{button ,AL(^H_UNDOENABLED_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;0)} [See list of classes](#)

(Read-write) Get or set the Undo enabled preference.

Data type

Integer (Boolean)

Syntax

flag = *preferencesobject*.UndoEnabled

preferencesobject.UndoEnabled = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Undo command enabled
FALSE (0)	Undo command disabled

Examples

```
CurrentApplication.Preferences.UndoEnabled = True
```

Freelance Graphics: UnitOfMeasure property

{button ,AL(^H_UNITOFMEASURE_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;'0)} [See list of classes](#)

(Read-write) Get or set the unit of measure used in the application. In Freelance Graphics, this is always twips.

Data type

Variant

Syntax

unitofmeasure = *applicationobject*.UnitOfMeasure

applicationobject.UnitOfMeasure = *unitofmeasure*

Legal values

The only legal value is \$ltsScaleModeTwip.

Examples

CurrentApplication.UnitofMeasure = \$ltsScaleModeTwip

Freelance Graphics: VersionID property

{button ,AL(^H_VERSIONID_PROPERTY_MEMDEF_RT;H_BASEOBJECT_CLASS;',0)} [See list of classes](#)

(Read-only) Determine the version of the code implementing an object. The version number changes for any release that is not 100% compatible with the previous release.

Data type

Long

Syntax

value = *object*.VersionID

Legal values

Any numeric value.

Freelance Graphics: VerticalAlignment property

{button ,AL('H_VERTICALALIGNMENT_PROPERTY_MEMDEF_RT;H_TEXTPROPERTIES_CLASS;',0)} [See list of classes](#)

(Read-write) Get or set the vertical alignment property for the text.

Data type

Variant (Enumerated)

Syntax

verticalalignment = *textobject*.VerticalAlignment

textobject.VerticalAlignment = *verticalalignment*

Legal values

<u>Value</u>	<u>Description</u>
\$ltsAlignmentTop	Align at top
\$ltsAlignmentVertCenter	Center
\$ltsAlignmentBottom	Align at bottom

Examples

```
MyText.TextBlock.TextProperties.VerticalAlignment = $ltsAlignmentTop
```

Freelance Graphics: ViewMode property

{button ,AL(^H_VIEWMODE_PROPERTY_MEMDEF_RT;H_DOCUMENT_CLASS;!,0)} [See list of classes](#)

(Read-write) Get or set the view mode.

Data type

Variant (Enumerated)

Syntax

viewmode = *documentobject.ViewMode*

documentobject.ViewMode = *viewmode*

Legal values

<u>Value</u>	<u>Description</u>
\$ViewDraw	Page view
\$ViewOutliner	Outliner view
\$ViewSorter	Page Sorter view
\$ViewSlideShow	Screen Show view

Freelance Graphics: Visible property

{button ,AL(^H_VISIBLE_PROPERTY_MEMDEF_RT;H_APPLICATION_CLASS;')0)} [See list of classes](#)

(Read-write) Get or set the visible attribute for the application.

Note For this release, this is effectively read-only, and always TRUE; sets are ignored.

Data type

Integer (Boolean)

Syntax

value = *applicationobject.Visible*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Visible
FALSE (0)	Not visible

Freelance Graphics: WaitForClick property

{button ,AL(^H_WAITFORCLICK_PROPERTY_MEMDEF_RT;H_DRAWOBJECT_CLASS;0)} [See list of classes](#)
(Read-write) Determine if the object needs a mouse click in order to draw the object on the page during a screen show.

Data type

Integer (Boolean)

Syntax

value = *drawobject.WaitForClick*

drawobject.WaitForClick = *value*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	Wait for click to draw the object
FALSE (0)	Do not wait for click to draw the object

Freelance Graphics: Width property

{button ,AL(^H_WIDTH_PROPERTY_MEMDEF_RT;H_APPLICATIONWINDOW_CLASS;H_DOCWINDOW_CLASS;
H_DRAWOBJECT_CLASS;H_LINESTYLE_CLASS;';0)} [See list of classes](#)

(Read-write) Get or set the width of a border or line.

(Read-only) Get the width of the application window, document window, or drawing object.

Data type

Integer (ApplicationWindow and DocWindow classes)

Variant (Border and LineStyle classes)

Long (DrawObject classes)

Syntax

width = *object.Width*

object.Width = *width*

Legal values

ApplicationWindow, DocWindow, and DrawObject classes: any positive value representing the width in twips.

Border and LineStyle classes:

<u>Value</u>	<u>Description</u>
<code>\$ltsBorderWidthNone</code>	Thinnest
<code>\$ltsBorderWidthVeryThin</code>	
<code>\$ltsBorderWidthThin</code>	
<code>\$ltsBorderWidthModeratelyThin</code>	
<code>\$ltsBorderWidthMedium</code>	
<code>\$ltsBorderWidthModeratelyThick</code>	
<code>\$ltsBorderWidthThick</code>	
<code>\$ltsBorderWidthVeryThick</code>	
<code>\$ltsBorderWidthExtremelyThick</code>	Thickest

Examples

```
MyRect.Border.Width = $ltsBorderWidthThin
```

Freelance Graphics: WordDoubleUnderline property

{button ,AL(^H_WORDDOUBLEUNDERLINE_PROPERTY_MEMDEF_RT;H_FONT_CLASS;')0)} [See list of classes](#)

(Read-write) Determine the Word double underline attribute for the font. This is a double line under words but not spaces.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

flag = *fontobject*.WordDoubleUnderline

fontobject.WordDoubleUnderline = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	WordDoubleUnderline on
FALSE (0)	WordDoubleUnderline off

Freelance Graphics: WordUnderline property

{button ,AL(^H_WORDUNDERLINE_PROPERTY_MEMDEF_RT;H_FONT_CLASS;'0)} [See list of classes](#)

(Read-write) Determine the Word underline attribute for the font. This is a single line under words but not spaces.

Note This property is ignored by Freelance Graphics, and is provided for compatibility with other Lotus products.

Data type

Integer (Boolean)

Syntax

flag = *fontobject*.WordUnderline

fontobject.WordUnderline = *flag*

Legal values

<u>Value</u>	<u>Description</u>
TRUE (non-0)	WordUnderline on
FALSE (0)	WordUnderline off

Freelance Graphics: WorkDir property

{button ,AL(^H_WORKDIR_PROPERTY_MEMDEF_RT;H_PREFERENCES_CLASS;','0)} [See list of classes](#)

(Read-write) Get or set the Work directory preference.

Data type

String

Syntax

workdirectory = *preferencesobject*.**WorkDir**

preferencesobject.**WorkDir** = *workdirectory*

Legal values

Any directory.

Examples

```
CurrentApplication.Preferences.WorkDir = "c:\lotus\work\flg"
```

Freelance Graphics: Application class

Controls a Freelance Graphics session.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Document	Application

Usage

You can use the CurrentApplication predefined variable to reference the properties and methods of the current Application object.

Freelance Graphics: Application class members

Properties

[ActiveDocument](#)
[ActiveDocWindow](#)
[Application](#) AS [Application class](#)
[ApplicationWindow](#)
[Colors](#) AS [Colors class](#)
[CurrentPrinter](#)
[DefaultFilePath](#)
[Documents](#) AS [Documents class](#)
[FontUnit](#)
[FullName](#)
[Interactive](#)
[Location](#)
[PageSelection](#) AS [PageSelection class](#)
[Parent](#)
[Path](#)
[Preferences](#) AS [Preferences class](#)
[Selection](#) AS [Selection class](#)
[UnitOfMeasure](#)
[Visible](#)

Methods

[CloseWindow](#)
[GetEnum](#)
[NearestColorFromRGB](#)
[NewDocument](#)
[OpenDocument](#)
[Quit](#)

Functions

None

Events

None

Freelance Graphics: ApplicationWindow class

The application's main window.

Base classes

BaseObject

Contained by

None

Usage

You can use the CurrentApplicationWindow predefined variable to reference the properties and methods of the currently selected ApplicationWindow object.

Freelance Graphics: ApplicationWindow class members

Properties

Height

Width

Methods

Cascade

Close

GotoNotes

Maximize

Minimize

Restore

Tile

Functions

None

Events

None

Freelance Graphics: Background class

Properties (color, fill pattern, etc.) of an object's background.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Background

Freelance Graphics: Background class members

Properties

BackColor

Color AS Color class

Pattern

Methods

RevertToStyle

Functions

None

Events

None

Freelance Graphics: BaseObject class

Abstract class used as the base class for all Freelance Graphics objects - no instances of this class ever exist.

Base classes

None

Contained by

All Freelance Graphics LotusScript classes. See [Freelance Graphics LotusScript Classes A-Z](#).

Freelance Graphics: BaseObject class members

The BaseObject class is an abstract class used as the base class for all Freelance Graphics objects - no instances of this class ever exist. It contains the following properties.

Properties

Application
Description
IsDraggable
IsSelectable
IsValid
Name
Parent
VersionID

Methods

None

Functions

None

Events

None

Freelance Graphics: Border class

Properties of an object's edges.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Border

Freelance Graphics: Border class members

Properties

Color AS Color class

Pattern

Width

Methods

RevertToStyle

Functions

None

Events

None

Freelance Graphics: BulletProperties class

Level-specific bullet properties.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
TextBlock	BulletProperties

Freelance Graphics: BulletProperties class members

Properties

Color AS Color class

ShadowColor

ShadowDepth

ShadowDirection

Size

StartNumber

Style

Methods

None

Functions

None

Events

None

Freelance Graphics: Chart class

A chart of any type. The Freelance Graphics Chart class is derived from the DrawObject class and contains all methods and properties of the ChartBase class. You create a chart using the CreateChart method of the Page class.

Note For more information about the LotusChart ChartBase class, see the Help contents under LotusScript, LotusChart LotusScript Reference, By Category, Classes.

Base classes

ChartBase

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Chart

Freelance Graphics: Color class

A color. You can specify a color using either the Item or RGBToColor methods of the Colors class.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Background	Color, BackColor
Border	Color
BulletProperties	Color
LineStyle	Color
Font	FontColor
TextProperties	ShadowColor

Freelance Graphics: Color class members

Properties

Blue

Green

Red

Methods

GetNearestIndex

GetRGB

SameColor

Functions

None

Events

None

Freelance Graphics: Colors class

The color library.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Application	Colors

Freelance Graphics: Colors class members

Properties

Count

Methods

ColorToRGB

GetIndex

GetNearestColor

IsEmpty

Item

RGBtoColor

Functions

None

Events

None

coordinates (defined)

The Freelance Graphics coordinate system has its origin (0,0) at the bottom left corner of the page. Coordinates are measured in twips (1/1440 of an inch, or 1/567 of a centimeter). In LotusScript syntax, the horizontal coordinate is usually referred to as *x*, and the vertical coordinate as *y*.

Freelance Graphics: Document class

Models a Freelance Graphics presentation. You can create an object of the Document class using the NewDocument method of the Application class.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Page	Document
Application	ActiveDocument

Usage

You can use the CurrentDocument predefined variable to reference the properties and methods of the currently selected Document object.

Freelance Graphics: Document class members

Properties

[Active](#)
[ActivePage](#)
[Application](#) AS [Application class](#)
[Author](#)
[Changed](#)
[Description](#)
[DocName](#)
[DocWindow](#) AS [DocWindow class](#)
[Embedded](#)
[FullName](#)
[IsOpen](#)
[Location](#)
[Pages](#) AS [Pages class](#)
[PageSelection](#) AS [PageSelection class](#)
[PageTransitionDelay](#)
[PageTransitionEffect](#)
[Parent](#)
[Path](#)
[ReadOnly](#)
[Selection](#) AS [Selection class](#)
[SmartLook](#)
[TemplatePageCount](#)
[ViewMode](#)

Methods

[Activate](#)
[AddToPageSelection](#)
[AddToSelection](#)
[Close](#)
[CopySelection](#)
[CreatePage](#)
[CreatePageFromTemplate](#)
[CutSelection](#)
[DeletePage](#)
[DeleteReviewer](#)
[DistributeForComment](#)
[FindObject](#)
[GotoPage](#)
[OpenPresForCopy](#)
[Paste](#)
[PasteSelectedPages](#)
[Print](#)
[PrintOut](#)
[RemoveFromSelection](#)
[RunDialog](#)
[Save](#)
[SaveAs](#)
[Select](#)
[SelectPageForCopy](#)
[SetViewMode](#)
[Show](#)
[StartGuidedTemplate](#)
[StopGuidedTemplate](#)

Functions

None

Events
None

Freelance Graphics: Documents class

A collection of Documents.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Application	Documents

Freelance Graphics: Documents class members

Properties

Count

Methods

GetIndex

IsEmpty

Item

Functions

None

Events

None

Freelance Graphics: DocWindow class

A document window.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Document	DocWindow

Usage

You can use the CurrentDocWindow predefined variable to reference the properties and methods of the currently selected DocWindow object.

Freelance Graphics: DocWindow class members

Properties

Height

Width

Methods

Cascade

Close

Maximize

Minimize

Restore

Tile

Functions

None

Events

None

Freelance Graphics: DrawObject class

Any selectable object on a page in a presentation.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Page	Title

Usage

You can use the Selection or CurrentSelection predefined variable to reference the properties and methods of the currently selected DrawObject object.

Freelance Graphics: DrawObject class members

Properties

[Background](#) AS [Background class](#)
[Border](#) AS [Border class](#)
[BuildBullets](#)
[Chart](#) AS [Chart class](#)
[Delay](#)
[DimPrevious](#)
[ExeName](#)
[Height](#)
[Image](#) AS [Image class](#)
[IsChart](#)
[IsGroup](#)
[IsImage](#)
[IsMedia](#)
[IsOleObj](#)
[IsPlacementBlock](#)
[IsTable](#)
[IsText](#)
[Left](#)
[LineStyle](#) AS [LineStyle class](#)
[Media](#) AS [Media class](#)
[Name](#)
[OleObj](#)
[Page](#) AS [Page class](#)
[PlacementBlock](#) AS [PlacementBlock class](#)
[PlayPriority](#)
[Priority](#)
[Table](#) AS [Table class](#)
[TextBlock](#) AS [TextBlock class](#)
[Top](#)
[TransitionEffect](#)
[WaitForClick](#)
[Width](#)

Methods

[AddPoint](#)
[ConvertTo](#)
[Copy](#)
[Cut](#)
[Delete](#)
[Flip](#)
[GetObjectData](#)
[Move](#)
[PutIntoPlacementBlock](#)
[Remove](#)
[Replicate](#)
[Rotate](#)
[SetObjectData](#)
[Stretch](#)
[Ungroup](#)

Functions

None

Events

None

Freelance Graphics: Font class

The style of a selection or block of text.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
TextBlock	Font
TextProperties	Font

Freelance Graphics: Font class members

Properties

Bold

Case

DoubleUnderline

FontColor

FontName

Italic

Overstrike

Size

SmallCaps

StrikeThrough

SubScript

SuperScript

Underline

WordDoubleUnderline

WordUnderline

Methods

RevertToStyle

Functions

None

Events

None

Freelance Graphics: Image class

Derived from DrawObject - a bitmap image.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Image

Freelance Graphics: Image class members

Properties

Brightness

Contrast

FileName

IsColor

IsInverted

Sharpness

Transparency

Methods

Crop

Functions

None

Events

None

Freelance Graphics: LineStyle class

The style of a line, arrow, or connector.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	LineStyle

Freelance Graphics: LineStyle class members

Properties

Color AS Color class

Pattern

Width

Methods

RevertToStyle

Functions

None

Events

None

Freelance Graphics: Media class

Derived from DrawObject - a movie or animation. To create a Media object, use the CreateMovie method of the Page class.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Media

Freelance Graphics: Media class members

Properties

FileName

Methods

Play

StopPlay

Functions

None

Events

None

Freelance Graphics: Objects class

A collection of DrawObjects.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Page	Objects

Freelance Graphics: Objects class members

Properties

Count

Methods

GetIndex

IsEmpty

Item

Functions

None

Events

None

Freelance Graphics: Page class

One page or slide in a presentation. To create a Page object, you can use either the CreatePage method or the CreatePageFromTemplate method of the Document class.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Page
Document	ActivePage

Usage

You can use the CurrentPage predefined variable to reference the properties and methods of the currently selected Page object.

Freelance Graphics: Page class members

Properties

[AutoTime](#)
[Delay](#)
[Document](#) AS [Document class](#)
[Exclude](#)
[Layout](#)
[Name](#)
[Number](#)
[Objects](#) AS [Objects class](#)
[Selection](#) AS [Selection class](#)
[Sound](#)
[SpeakerNoteText](#)
[Title](#)
[TransitionEffect](#)

Methods

[CopyPage](#)
[CreateArrow](#)
[CreateChart](#)
[CreateComment](#)
[CreateLine](#)
[CreateMovie](#)
[CreateOval](#)
[CreatePlacementBlock](#)
[CreateRect](#)
[CreateSymbol](#)
[CreateTable](#)
[CreateText](#)
[CutPage](#)
[DeleteSpeakerNote](#)
[FindNextObject](#)
[FindObject](#)
[GetSpeakerNoteMarkup](#)
[Move](#)
[Paste](#)
[PasteSpecial](#)
[Remove](#)
[Replicate](#)

Functions

None

Events

None

Freelance Graphics: Pages class

A collection of Pages.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Document	Pages

Freelance Graphics: Pages class members

Properties

Count

Methods

GetIndex

IsEmpty

Item

Functions

None

Events

None

Freelance Graphics: PageSelection class

The currently selected pages; derives methods and properties from Page.

Base classes

Page

Contained by

<u>Class</u>	<u>Property</u>
Application	PageSelection
Document	PageSelection

Freelance Graphics: PageSelection class members

Properties

SelectionCount

Methods

AddToSelection

ClearSelection

GetSelection

RemoveFromSelection

Select

Functions

None

Events

None

Freelance Graphics: PlacementBlock class

Derived from DrawObject - a "Click here..." block. To create a "Click here..." block, use the CreatePlacementBlock method of the Page class.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	PlacementBlock

Freelance Graphics: PlacementBlock class members

Properties

ID

PromptText

Type

Methods

Activate

BrowseDiagrams

BrowseSymbols

Insert

Functions

None

Events

None

Freelance Graphics: Preferences class

Freelance Graphics preferences.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Application	Preferences

Freelance Graphics: Preferences class members

Properties

AutoSave
AutoSaveInterval
BackupDir
BlackWhitePal
BorderDisplay
DisplayCoords
DisplayDrawRuler
DisplayGrid
DisplayTextRuler
OffsetReplicate
ScanSpeed
SkipWelcome
SnapToGrid
StartupView
TemplateDir
UndoEnabled
WorkDir

Methods

None

Functions

None

Events

None

RGB value (defined)

A type Long number specifying the amount of red, green, and blue tint in a 24-bit color. The high-order byte is 0 and is unused. The three low-order bytes each contain a binary value from 0 (no tint) to 255 (maximum tint), with the lowest-order byte representing blue, the next byte green, and the next byte red.

Freelance Graphics: Selection class

The currently selected draw objects; derives methods and properties from DrawObject.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
Application	Selection
Page	Selection

Freelance Graphics: Selection class members

Properties

SelectedObjects

SelectionCount

Methods

AddToSelection

Align

ClearSelection

Connect

GetSelection

Group

RemoveFromSelection

Select

Functions

None

Events

None

Freelance Graphics: Table class

Derived from DrawObject - a Freelance Graphics table. To create a Table object, use the CreateTable method of the Page class.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	Table

Freelance Graphics: Table class members

Properties

ColCount
RowCount

Methods

DeleteCol
DeleteRow
GetCell
InsertCol
InsertRow

Functions

None

Events

None

Freelance Graphics: TextBlock class

Derived from DrawObject - a text block or text shape.

Base classes

DrawObject

Contained by

<u>Class</u>	<u>Property</u>
DrawObject	TextBlock

Freelance Graphics: TextBlock class members

Properties

BulletProperties AS BulletProperties class

Font AS Font class

Text

TextProperties AS TextProperties class

Methods

ApplyStyle

CreateStyle

EnterEditMode

GetBulletCount

GetMarkup

GetNthBullet

LeaveEditMode

RevertToStyle

Functions

None

Events

None

Freelance Graphics: TextProperties class

Level-specific text properties.

Base classes

BaseObject

Contained by

<u>Class</u>	<u>Property</u>
TextBlock	TextProperties

Freelance Graphics: TextProperties class members

Properties

FirstIndent

Font AS Font class

HorizontalAlignment

LineLead

ParaIndent

RightIndent

Shadow

ShadowColor AS Color class

ShadowDepth

ShadowDirection

VerticalAlignment

Methods

None

Functions

None

Events

None

twips (defined)

A twip is a screen-independent unit of measurement (unlike a pixel, which is screen-dependent). There are 1440 twips to an inch and 567 twips to a centimeter. In LotusScript, all Freelance Graphics coordinates and sizes are specified in twips.

Freelance Graphics LotusScript Classes A-Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Application class](#)
[ApplicationWindow class](#)

B

[Background class](#)
[BaseObject class](#)
[Border class](#)
[BulletProperties class](#)

C

[Chart class](#)
[Color class](#)
[Colors class](#)

D

[Document class](#)
[Documentsclass](#)
[DocWindow class](#)
[DrawObject class](#)

E

(None)

F

[Font class](#)

G, H, I, J, K

(None)

L

[LineStyle class](#)

M

Media class

N

(None)

O

Objects class

P

Page class

Pages class

PageSelection class

PlacementBlock class

Preferences

Q, R

(None)

S

Selection class

T

Table class

TextBlock class

TextProperties class

U, V, W, X, Y, Z

(None)

Overview: Scripting in Freelance Graphics

Freelance Graphics has an object-oriented Application Programming Interface (API) to Freelance Graphics functionality. It is an extension of LotusScript--a cross-product Basic scripting language.

Use the API to automate Freelance Graphics. For example, you can use the API to create programs that make it easy to edit a frequently used presentation format. You can also use LotusScript to share information from various Lotus applications to perform tasks that you do on a regular basis.

You use the Script Editor and Debugger to create, edit, and debug scripts. With the Freelance Graphics API, you can create "scripts" that execute when you click a button or a "Click here..." block in a SmartMaster content topic, or when an icon is clicked. You can also execute a script when you choose Edit - Script - Run.

Sources of information

There is information in the Help system about the Freelance Graphics API (look under Help - Help Topics - How Do I - Script). There is also Help for the Script Editor and Debugger and for the overall LotusScript language.

For more information about LotusScript, use the coupon enclosed with Freelance Graphics to receive the *LotusScript Programmer's Guide*.

The structure of LotusScript

The structure of the Freelance Graphics API is as follows: There is always one Freelance Graphics application that contains any number of documents; each document contains any number of pages; and each page contains any number of objects.

The API and LotusScript use the principles of object-oriented programming. For information about object-oriented programming and LotusScript in Lotus applications, see the *LotusScript Programmer's Guide*.

Classes

The Freelance Graphics API consists of approximately 25 pre-defined object classes (custom LotusScript data types). Each class definition includes a set of properties and methods.

In Freelance Graphics the API is object oriented; the classes in the API follow the structure of the Freelance Graphics application. That is, the application consists of one or more application windows; an application window consists of documents; and documents consist of pages. Each page may contain a variety of objects, such as circles and squares, charts, OLE objects, and so on.

When you use the Freelance Graphics LotusScript API, an application window, for example, is an instance of the AppWindow class. Individual objects on a page, such as a rectangle or a text block, are instances of the DrawObject class. For a list of classes, as well as reference information, see [Freelance Graphics LotusScript Classes](#).

There are several classes in the Freelance Graphics API that are collection classes, that is, collections of other objects in the presentation. For example, to get from one page to another you would look through all numbers of pages in the document class. Classes that are collection classes are identified as such in the class reference Help.

Note In Freelance Graphics script syntax, the index of the first item in a collection or table is one. In other Lotus applications (WordPro and Approach, for example) the index of the first item is zero. In the future, script indexing may be standardized across all Lotus applications. If zero becomes the standard for the first collection or table item, you will need to adjust all existing Freelance Graphics script statements containing collection or table references.

Properties

Properties define the appearance and behavior of objects. Many object classes have properties defining visual attributes such as background color, size, and location. Some properties apply to only one object class. For example, the Title property is unique to the Page class. On the other hand, the Width property is a property of the AppWindow, Border, DocWindow, and DrawObject classes.

Some classes may also act as a property on another class. For example, an instance of the Color class can be a property for an instance of the Font class. When this is the case, it is noted in the property description.

For a list of properties, see [Freelance Graphics LotusScript Properties](#).

Methods

Methods are subprograms that you use to manipulate objects. For example, you can use the Move method to move an instance of the DrawObject class (a rectangle, for example) or you can use the Copy method to copy an instance of the DrawObject class.

For a list of methods, see [Freelance Graphics LotusScript Methods](#).

Freelance Graphics API rules

Every document, page, and object has a name. You can assign names to pages and objects. The document name is read-only. Freelance Graphics stores the names given to pages and objects in the presentation, so that they are available in future script sessions. You can find a draw object if you know its name by using the FindObject method, see [FindObject method](#). You can also use Bind to find the name of a named object of any class. For example,

```
Dim p as Page
Set p = Bind("Page 1")
Print p.number
```

Whenever you do anything to an object, that object becomes the selected object.

LotusScript also includes predefined variables that you can use to reference Freelance Graphics objects:

- CurrentApplication--operates on the application and uses the properties and methods of the Application class.
- CurrentApplicationWindow--operates on the application and uses the properties and methods of the ApplicationWindow class.
- CurrentDocument--operates on the current document and uses the properties and methods of the Document class.
- CurrentDocWindow--operates on the application and uses the properties and methods of the DocWindow class.
- CurrentPage--operates on the current page and uses the properties and methods of the Page class.
- Selection--operates on the currently selected object or objects and uses the properties and methods of the DrawObject class.

How scripting works

You can write scripts that operate on documents, pages, and objects. You can also attach scripts to a page, a "Click here..." block, or SmartIcons. These are slightly different processes.

To operate on a document, page, or an object, you use the Script Editor to write a script and then save it as an LSS (or LSO) file. (An LSO file is a compiled LSS file.) You run the script by choosing Edit - Script - Run, and typing the LSS (or LSO) file name.

To attach (or assign) a script to a page or a "Click here..." block, you must be in a content topic (see [To open a content topic](#)). If the script is attached to a "Click here..." block or a page, it is saved in the content topic itself. The script runs "automatically" when the user opens the page or clicks on the "Click here..." block.

See [Attaching a script to a "Click here" block](#), [Attaching a script to a content page](#), or [Creating a script button](#).

For information about attaching scripts to SmartIcons, see [Attaching a script to an icon](#).

For information about creating, editing, and debugging a script, see [Overview: Creating, editing, and debugging a script](#).

{button ,AL(`H_FLW_SCRIPT_RUN_STEPS;H_FLW_SCRIPT_CR_ED_DEB_OVER',0)} [See related topics](#)

Overview: Creating, editing, and debugging a script

The Script Editor and Debugger in which you write, run, and debug a script has Help. Press F1 when you want information. To launch the IDE in Freelance Graphics, choose Edit - Script - Show Script Editor. For more information about working in the Script Editor and Debugger, look under Script Editor or Script Debugger in the Freelance Graphics Help index.

Tip You can also review the list of available Freelance Graphics classes, methods, and properties by using the browser in the IDE. You can get help on a highlighted item by pressing F1.

The TransScript window

When you run a script using Edit - Script - Run in Freelance Graphics, you can see error messages and output, however, the transcript window must be visible. To view the transcript window choose Edit - Script - Show TransScript Window.

Creating objects and assigning object references

You can access Freelance Graphics objects, including draw objects, pages, and whole presentations, by assigning a reference to that object. References can be assigned to objects that are created within a script or to already existing objects.

To assign an existing Freelance Graphics object a reference variable, use the Set statement. Set must be used any time a reference variable is assigned to an instance of a class. For instance:

```
Dim MyRect as DrawObject
Dim MyPage as Page
'// -----
'// set MyPage equal to the third Page in the presentation, My Rect to the second
item on MyPage
'// -----
Set MyPage = CurrentDocument.Pages.Item(3)
Set MyRect = MyPage.Objects.Item(2)
'// -----
'// set MyRect equals the object with name My Rectangle
'// -----
Set MyRect = CurrentPage.FindObject("My Rectangle")
```

To create a Freelance Graphics object, you must use the appropriate method. In general, object creation is done by methods on the container. The methods for creating draw objects belong to the Page class. For instance, to create a rectangle:

```
Dim MyRect as DrawObject
Set MyRect = CurrentPage.CreateRect (2000, 3000, 3000, 4000)
```

The method for creating a page belongs to the Document class:

```
Dim MyPage as Page
Set MyPage = CurrentDocument.CreatePage("Title of Page", 2)
```

The method for creating a document, NewDocument, belongs to the Application class. Also, OpenDocument opens an existing document (in that sense it "creates" a document object). For example,

```
Set MyDoc = CurrentApplication.OpenDocument("test.prz")
```

Using LotusScript predefined variables

You can use the LotusScript predefined variables to write scripts that operate on the currently selected object, page, document, document window, application window, and application.

A sample script that changes the pattern of the currently selected object is as follows:

```
Sub Main
Selection.Background.Pattern=$LtsFillGray2
End Sub
```

A sample script that gives a page transition delay of ten seconds to the currently selected page is as follows:

```
Sub Main
```

```
CurrentPage.PageTransitionDelay=10
```

```
End Sub
```

Launching a script from the command line

You can launch a script from the command line by clicking Start in the taskbar, choosing Run, and typing:

```
C:\Freelancepath\F32main /r lsscript.lss filename.prz
```

(*lsscript.lss* is the name of the script you want to run and *filename* is the name of the presentation you want to run the script in.)

Using scripts and content topics

In addition to creating a script by choosing Edit - Script - Show Script Editor, you can create and attach a script to a placement block or a page when you have a content topic open. For more information on content topics, see [Overview: What is a content topic](#).

For information about creating and attaching a script to a "Click here" block, see [Attaching a script to a "Click here" block](#).

Note Whenever you attach a script to a placement block or to a page, you must name the subroutine "Main."

Sample scripts

For examples of working code and of the object-oriented syntax used in the Freelance Graphics API, review the scripts that are used by SmartMaster content (SMC) files in Freelance Graphics. Scripts in these files refer to the source code contained in the file GTSCRPT.LSS, located in the \LOTUS\SMARTERS\FLG directory.

Caution Modifying script code in this file may cause problems with Freelance Graphics content topics. Make a copy of the file to experiment with.

```
{button ,AL(';H_AV_ATTACHING_A_SCRIPT_TO_AN_ICON_STEPS;H_FLW_SCRIPT_IDE_STEPS;H_FLW_SCRIP  
T_OVER;H_FLW_SCRIPT_RUN_STEPS;H_SMDESIGN_SCRIPT_BUTTON_STEPS',0)} See related topics
```

Opening the Script Editor and Debugger

To open the Script Editor and Debugger, choose Edit - Script - Show Script Editor.

{button ,AL(`H_FLW_SCRIPT_OVER;H_FLW_SCRIPT_RUN_STEPS',0)} [See related topics](#)

Running a script

Follow these steps to run a script that is not attached to a page or a "Click here..." block.

1. Choose Edit - Script - Run.
2. Type the script file name in the File name box.
3. Click Open.

Note To see error messages and output, open the transcript window; choose Edit - Script - Show TransScript Window.

{button ,AL(`H_FLW_SCRIPT_IDE_STEPS;H_FLW_SCRIPT_OVER',0)} [See related topics](#)

Setting basic preferences in the Script Editor

You can set the following preferences in the Basics panel of the Preferences dialog box:

- Tab or Indent Script text
- Display error counts

{button ,AL('H_IDEMENU_INDENTING_SCRIPT_TEXT_STEPS;H_IDEMENU_SETTING_AND_CHANGING_TABS_IN_A_SCRIPT_STEPS;H_IDEMENU_DISPLAYING_ERROR_COUNTS_IN_A_SCRIPT_STEPS;',0)} [See related topics](#)

Creating a new function

1. In the Script Editor, choose Script - New Function.
2. Type a name in the Name text box.
3. Click OK.

Note For more information about creating scripts, see "Scripts, creating" in the Help index.

Creating a new sub

1. In the Script Editor, choose Script - New Sub.
2. Type a name in the Name text box.
3. Click OK.

Note For more information about creating scripts, see "Scripts, creating" in the Help index.

Displaying error counts in a script

The Show Error Count Dialog check box lets you display the number of errors found after a script has been compiled.

Before you compile your script

1. In the Script Editor, choose File - Script Editor Preferences.
2. Click the Basics tab and select the Show Error Count Dialog check box.
3. Click OK.

Compiling your script

In the Script Editor, choose either Script - Check Scripts for Object. This command executes your uncompiled scripts and displays error counts.

{button ,AL(`H_IDEMENU_BASIC_PREFERENCES_CS;`,`0`)} [See related topics](#)

Exporting a script

Use this procedure to export (save) a script for the current object.

1. In the Script Editor, choose File - Export Script.
2. Select the script name or type a new name in the File name list box.
3. Click OK.

{button ,AL('H_IDEMENU_IMPORTING_A_SCRIPT_STEPS',0)} [See related topics](#)

Importing a script

1. In the Script Editor, choose File - Import Script.
2. Select the script name in the File name list box.
3. Click OK.

{button ,AL(`H_IDEMENU_EXPORTING_A_SCRIPT_STEPS',0)} [See related topics](#)

Indenting script text

1. In the Script Editor, choose File - Script Editor Preferences.
2. Click the Basics tab and select the Smart Indenting check box to automatically indent a script.
3. Click OK.

{button ,AL(`H_IDEMENU_BASIC_PREFERENCES_CS;',0)} [See related topics](#)

Script Editor Debug Menu

Step

Execute the current statement and enter the procedure it calls.

Step Over

Execute the current statement and the procedure it calls.

Step Exit

Exit the remainder of the current procedure.

Continue Execution

Continue executing the current script.

Stop Execution

Stop executing scripts.

Set Breakpoint

Set breakpoints in the selected lines.

Disable Breakpoint

Disable breakpoints in the selected lines.

Script Editor Edit Menu

Undo

Undo the last edit.

Cut

Cut to the Clipboard.

Copy

Copy to the Clipboard.

Paste

Paste from the Clipboard.

Clear

Erase the current selection without copying anything to the Clipboard.

Script Editor File Menu

Save Presentation

Save the active presentation (but not any scripts).

Import Script File

Import a script for the current object.

Export Script File

Export (save) a script for the current object.

Script Editor Preferences

Set preferences for the Script Editor.

Close Script Editor

Close the active window.

Script Editor Help Menu

Help Topics

List Help topics for the Script Editor.

Script Editor Script Menu

Run Current Sub

Run the script for the current sub and report errors.

Check Scripts for Object

Check the structure and syntax of all uncompiled scripts for the current object and report errors.

New Sub

Create a sub.

New Function

Create a function.

Previous Script

Move to the previous sub.

Next Script

Move to the next sub.

Script Pane

Return focus to the script pane.

Switch Panes

Switch focus between the upper and lower panes.

Browser

Switch to the Browser panel to display script keywords and component names.

Output

Switch to the Output panel to display output for LotusScript print commands.

Breakpoints

Switch to the Breakpoints panel to display breakpoints.

Variables

Switch to the Variables panel to display variables.

Setting tabs in a script

1. In the Script Editor, choose File - Script Editor Preferences.
2. Click the Basics tab.
3. Specify the number of spaces you want between tab stops in the Characters list box.
4. Click OK.

{button ,AL(^H_IDEMENU_INDENTING_SCRIPT_TEXT_STEPS;H_IDEMENU_BASIC_PREFERENCES_CS;',0)}
[See related topics](#)

Freelance Graphics LotusScript Methods A-Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Activate method](#)
[AddPoint method](#)
[AddToPageSelection method](#)
[AddToSelection method](#)
[Align method](#)
[ApplyStyle method](#)

B

[BrowseDiagrams method](#)
[BrowseSymbols method](#)

C

[Cascade method](#)
[ClearSelection method](#)
[Close method](#)
[CloseWindow method](#)
[ColorToRGB method](#)
[Connect method](#)
[ConvertTo method](#)
[Copy method](#)
[CopyPage method](#)
[CopySelection method](#)
[CreateArrow method](#)
[CreateChart method](#)
[CreateComment method](#)
[CreateLine method](#)
[CreateMovie method](#)
[CreateOval method](#)
[CreatePage method](#)
[CreatePageFromTemplate method](#)

[CreatePlacementBlock method](#)
[CreateRect method](#)
[CreateStyle method](#)
[CreateSymbol method](#)
[CreateTable method](#)
[CreateText method](#)
[Cut method](#)
[CutPage method](#)
[CutSelection method](#)

D

[DeleteCol method](#)
[DeletePage method](#)
[DeleteReviewer method](#)
[DeleteRow method](#)
[DeleteSpeakerNote method](#)
[DistributeForComment method](#)

E

[EnterEditMode method](#)

F

[FindNextObject method](#)
[FindObject method](#)
[Flip method](#)

G

[GetBulletCount method](#)
[GetCell method](#)
[GetEnum method](#)
[GetIndex method](#)
[GetMarkup method](#)
[GetNearestColor method](#)
[GetNearestIndex method](#)
[GetNthBullet method](#)
[GetObjectData method](#)
[GetRGB method](#)
[GetSelection method](#)
[GetSelection method](#)
[GetSpeakerNoteMarkup method](#)
[GotoNotes method](#)
[GotoPage method](#)
[Group method](#)

H

(None)

I

[Import method](#)
[Insert method](#)
[InsertCol method](#)
[InsertRow method](#)
[IsEmpty method](#)
[Item method](#)

J, K

(None)

L

LeaveEditMode method

M

Maximize method

Minimize method

Move method

N

NearestColorFromRGB method

NewDocument method

O

OpenDocument method

OpenPresForCopy method

P

Paste method

PasteSelectedPages method

PasteSpecial method

Play method

Print method

PrintOut method

PutIntoPlacementBlock method

Q

Quit method

R

Remove method

RemoveFromSelection method

Replicate method

Restore method

RevertToStyle method

RGBToColor method

Rotate method

RunDialog method

S

SameColor method

Save method

SaveAs method

Select method

SelectPageForCopy method

SetObjectData method

SetViewMode method

Show method

StartGuidedTemplate method

StopGuidedTemplate method

StopPlay method

Stretch method

T

Tile method

U

Ungroup method

V, W, X, Y, Z

(None)

Freelance Graphics LotusScript Properties A-Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Active property](#)
[ActiveDocument property](#)
[ActiveDocWindow property](#)
[ActivePage property](#)
[Application property](#)
[ApplicationWindow property](#)
[Author property](#)
[AutoSave property](#)
[AutoSaveInterval property](#)
[AutoTime property](#)

B

[BackColor property](#)
[Background property](#)
[BackupDir property](#)
[BlackWhitePal property](#)
[Blue property](#)
[Bold property](#)
[Border property](#)
[BorderDisplay property](#)
[BuildBullets property](#)
[BulletProperties property](#)

C

[Case property](#)
[Changed property](#)
[Chart property](#)
[CodePage property](#)
[ColCount property](#)
[Color property](#)

[Colors property](#)
[Count property](#)
[CurrentPrinter property](#)

D

[DefaultFilePath property](#)
[Delay property](#)
[Description property](#)
[DimPrevious property](#)
[DisplayCoords property](#)
[DisplayDrawRuler property](#)
[DisplayGrid property](#)
[DisplayTextRuler property](#)
[DocName property](#)
[Document property](#)
[Documents property](#)
[DocWindow property](#)
[DoubleUnderline property](#)

E

[Embedded property](#)
[Exclude property](#)
[ExeName property](#)

F

[FirstIndent property](#)
[Font property](#)
[FontColor property](#)
[FontName property](#)
[FontUnit property](#)
[FullName property](#)

G

(None)

H

[Height property](#)
[HorizontalAlignment property](#)

I

[ID property](#)
[Image property](#)
[Interactive property](#)
[IsChart property](#)
[IsDraggable property](#)
[IsGroup property](#)
[IsImage property](#)
[IsMedia property](#)
[IsOleObj property](#)
[IsOpen property](#)
[IsPlacementBlock property](#)
[IsSelectable property](#)
[IsTable property](#)
[IsText property](#)
[IsValid property](#)
[Italic property](#)

J, K

(None)

L

[Layout property](#)
[Left property](#)
[LineLead property](#)
[LineStyle property](#)
[Location property](#)

M

[Media property](#)

N

[Name property](#)
[Number property](#)

O

[Objects property](#)
[OffsetReplicate property](#)
[OleObj property](#)
[Overstrike property](#)

P

[Page property](#)
[Pages property](#)
[PageSelection property](#)
[PageTransitionDelay property](#)
[PageTransitionEffect property](#)
[ParaIndent property](#)
[Parent property](#)
[Path property](#)
[Pattern property](#)
[PlacementBlock property](#)
[PlayPriority property](#)
[Preferences property](#)
[Priority property](#)
[PromptText property](#)

Q

(None)

R

[ReadOnly property](#)
[Red property](#)
[RemoveMedia property](#)
[RightIndent property](#)
[RowCount property](#)

S

[ScanSpeed property](#)
[SelectedObjects property](#)
[Selection property](#)
[SelectionCount property](#)
[Shadow property](#)
[ShadowColor property](#)
[ShadowDepth property](#)
[ShadowDirection property](#)
[Size property](#)
[SkipWelcome property](#)
[SmallCaps property](#)
[SmartLook property](#)

[SnapToGrid property](#)
[Sound property](#)
[SpeakerNoteText property](#)
[StartNumber property](#)
[StartupView property](#)
[StrikeThrough property](#)
[Style property](#)
[SubScript property](#)
[SuperScript property](#)

T

[Table property](#)
[TemplateDir property](#)
[TemplatePageCount property](#)
[Text property](#)
[TextBlock property](#)
[TextProperties property](#)
[TextTightness property](#)
[Title property](#)
[Top property](#)
[TransitionEffect property](#)
[Type property](#)

U

[Underline property](#)
[UndoEnabled property](#)
[UnitOfMeasure property](#)

V

[VersionID property](#)
[VerticalAlignment property](#)
[ViewMode property](#)
[Visible property](#)

W

[WaitForClick property](#)
[Width property](#)
[WordDoubleUnderline property](#)
[WordUnderline property](#)
[WorkDir property](#)

X, Y, Z

(None)

Freelance Graphics LotusScript A-Z

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

A

[Activate method](#)
[Active property](#)
[ActiveDocument property](#)
[ActiveDocWindow property](#)
[ActivePage property](#)
[AddPoint method](#)
[AddToPageSelection method](#)
[AddToSelection method](#)
[Align method](#)
[Application class](#)
[Application property](#)
[ApplicationWindow class](#)
[ApplicationWindow property](#)
[ApplyStyle method](#)
[Author property](#)
[AutoSave property](#)
[AutoSaveInterval property](#)
[AutoTime property](#)

B

[BackColor property](#)
[Background class](#)
[Background property](#)
[BackupDir property](#)
[BaseObject class](#)
[BlackWhitePal property](#)
[Blue property](#)
[Bold property](#)
[Border class](#)
[Border property](#)

[BorderDisplay property](#)
[BrowseDiagrams method](#)
[BrowseSymbols method](#)
[BuildBullets property](#)
[BulletProperties class](#)
[BulletProperties property](#)

C

[Cascade method](#)
[Case property](#)
[Changed property](#)
[Chart property](#)
[ClearSelection method](#)
[Close method](#)
[CloseWindow method](#)
[CodePage property](#)
[ColCount property](#)
[Color class](#)
[Color property](#)
[Colors class](#)
[Colors property](#)
[ColorToRGB method](#)
[Connect method](#)
[ConvertTo method](#)
[Copy method](#)
[CopyPage method](#)
[CopySelection method](#)
[Count property](#)
[CreateArrow method](#)
[CreateChart method](#)
[CreateComment method](#)
[CreateLine method](#)
[CreateMovie method](#)
[CreateOval method](#)
[CreatePage method](#)
[CreatePageFromTemplate method](#)
[CreatePlacementBlock method](#)
[CreateRect method](#)
[CreateStyle method](#)
[CreateSymbol method](#)
[CreateTable method](#)
[CreateText method](#)
[CurrentPrinter property](#)
[Cut method](#)
[CutPage method](#)
[CutSelection method](#)

D

[DefaultFilePath property](#)
[Delay property](#)
[DeleteCol method](#)
[DeletePage method](#)
[DeleteReviewer method](#)
[DeleteRow method](#)
[DeleteSpeakerNote method](#)
[Description property](#)
[DimPrevious property](#)
[DisplayCoords property](#)

[DisplayDrawRuler property](#)
[DisplayGrid property](#)
[DisplayTextRuler property](#)
[DistributeForComment method](#)
[DocName property](#)
[Document class](#)
[Document property](#)
[Documents class](#)
[Documents property](#)
[DocWindow class](#)
[DocWindow property](#)
[DoubleUnderline property](#)
[DrawObject class](#)

E

[Embedded property](#)
[EnterEditMode method](#)
[Exclude property](#)
[ExeName property](#)

F

[FindNextObject method](#)
[FindObject method](#)
[FirstIndent property](#)
[Flip method](#)
[Font class](#)
[Font property](#)
[FontColor property](#)
[FontName property](#)
[FontUnit property](#)
[FullName property](#)

G

[GetBulletCount method](#)
[GetCell method](#)
[GetEnumerator method](#)
[GetIndex method](#)
[GetMarkup method](#)
[GetNearestColor method](#)
[GetNearestIndex method](#)
[GetNthBullet method](#)
[GetObjectData method](#)
[GetRGB method](#)
[GetSelection method](#)
[GetSelection method](#)
[GetSpeakerNoteMarkup method](#)
[GotoNotes method](#)
[GotoPage method](#)
[Green property](#)
[Group method](#)

H

[Height property](#)
[HorizontalAlignment property](#)

I

[ID property](#)
[Image property](#)
[Import method](#)

[Insert method](#)
[InsertCol method](#)
[InsertRow method](#)
[Interactive property](#)
[IsChart property](#)
[IsDraggable property](#)
[IsEmpty method](#)
[IsGroup property](#)
[IsImage property](#)
[IsMedia property](#)
[IsOleObj property](#)
[IsOpen property](#)
[IsPlacementBlock property](#)
[IsSelectable property](#)
[IsTable property](#)
[IsText property](#)
[IsValid property](#)
[Italic property](#)
[Item method](#)

J, K

(None)

L

[Layout property](#)
[LeaveEditMode method](#)
[Left property](#)
[LineLead property](#)
[LineStyle class](#)
[LineStyle property](#)
[Location property](#)

M

[Maximize method](#)
[Media class](#)
[Media property](#)
[Minimize method](#)
[Move method](#)

N

[Name property](#)
[NearestColorFromRGB method](#)
[NewDocument method](#)
[Number property](#)

O

[Objects class](#)
[Objects property](#)
[OffsetReplicate property](#)
[OleObj property](#)
[OpenDocument method](#)
[OpenPresForCopy method](#)
[Overstrike property](#)

P

[Page class](#)
[Page property](#)
[Pages class](#)
[Pages property](#)

[PageSelection class](#)
[PageSelection property](#)
[PageTransitionDelay property](#)
[PageTransitionEffect property](#)
[ParaIndent property](#)
[Parent property](#)
[Paste method](#)
[PasteSelectedPages method](#)
[PasteSpecial method](#)
[Path property](#)
[Pattern property](#)
[PlacementBlock class](#)
[PlacementBlock property](#)
[Play method](#)
[PlayPriority property](#)
[Preferences class](#)
[Preferences property](#)
[Print method](#)
[PrintOut method](#)
[Priority property](#)
[PromptText property](#)
[PutIntoPlacementBlock method](#)

Q

[Quit method](#)

R

[ReadOnly property](#)
[Red property](#)
[Remove method](#)
[RemoveFromSelection method](#)
[RemoveMedia property](#)
[Replicate method](#)
[Restore method](#)
[RevertToStyle method](#)
[RGBToColor method](#)
[RightIndent property](#)
[Rotate method](#)
[RowCount property](#)
[RunDialog method](#)

S

[SameColor method](#)
[Save method](#)
[SaveAs method](#)
[ScanSpeed property](#)
[Select method](#)
[SelectedObjects property](#)
[Selection class](#)
[Selection property](#)
[SelectionCount property](#)
[SelectPageForCopy method](#)
[SetObjectData method](#)
[SetViewMode method](#)
[Shadow property](#)
[ShadowColor property](#)
[ShadowDepth property](#)
[ShadowDirection property](#)

[Show method](#)
[Size property](#)
[SkipWelcome property](#)
[SmallCaps property](#)
[SmartLook property](#)
[SnapToGrid property](#)
[Sound property](#)
[SpeakerNoteText property](#)
[StartGuidedTemplate method](#)
[StartNumber property](#)
[StartupView property](#)
[StopGuidedTemplate method](#)
[StopPlay method](#)
[Stretch method](#)
[StrikeThrough property](#)
[Style property](#)
[SubScript property](#)
[SuperScript property](#)

T

[Table class](#)
[Table property](#)
[TemplateDir property](#)
[TemplatePageCount property](#)
[Text property](#)
[TextBlock class](#)
[TextBlock property](#)
[TextProperties class](#)
[TextProperties property](#)
[TextTightness property](#)
[Tile method](#)
[Title property](#)
[Top property](#)
[TransitionEffect property](#)
[Type property](#)

U

[Underline property](#)
[UndoEnabled property](#)
[Ungroup method](#)
[UnitOfMeasure property](#)

V

[VersionID property](#)
[VerticalAlignment property](#)
[ViewMode property](#)
[Visible property](#)

W

[WaitForClick property](#)
[Width property](#)
[WordDoubleUnderline property](#)
[WordUnderline property](#)
[WorkDir property](#)

X, Y, Z

(None)

