Pixologic ®
makers of ZBRUSH

ZScript Command Reference

Contents

Introduction

All the zscript commands are listed below, with brief explanations and examples. If you are new to scripting then it's probably best to read Getting Started with ZScripting before using this list to extend your knowledge. If you are already familiar with scripting then you may find ZScripting for Seasoned Scripters useful.

> ***Note:*** Some commands are indicated as **Top Level** which means that they should not be placed inside other commands. Some are **Sub-Level** only; these commands must *always* be placed within other commands. Failure to follow these rules may result in ZBrush showing an error message after attempting to compile the zscript. The remaining commands can be placed anywhere (either inside or outside other commands).

Buttons and Sliders

IButton	**[IButton,** Button name, Popup info Text, Commands group to execute when button is pressed, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled), Button width as proportion of palette width (e.g. 0.5 half width, 1 full width) or in pixels (0:AutoWidth NonZero:Specified width), Optional hotkey, Optional button icon (.psd .bmp + .pct for Mac Systems), Button height in pixels (0:AutoHeight NonZero:Specified height)**]**

Creates an interactive push button (can be placed anywhere but advise **Top Level**).

Example:

```
[IButton, "Click Me", , [MessageOK, YouClicked]]
```

Creates an interactive button with "Click Me" text which will display a "YouClicked" message when pressed (**Top Level**).

```
[IButton, ???, "This is a macro button", ...commands...]
```

The special Button name **???** indentifies this as a Macro. If the zscript text file is placed in the ZStartup\Macros folder, ZBrush will automatically load the zscript; the file name will show as the Button name.

ISlider	**[ISlider,** Slider name, CurValue, Resolution, MinValue, MaxValue, Popup info Text, Commands group to execute when value is changed, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled), Slider width in pixels (0:AutoWidth NonZero:Specified width)**]**

Creates an interactive slider (**Top Level**).

Example:

```
[ISlider, ChangeMe, 12, 1, 0, 100, , [MessageOk, ThankYou]]
```

Creates an interactive slider with initial value of 12, range of 0 to 100 and "ChangeMe" text which will display a "Thankyou" message when its value is changed

ISwitch	**[ISwitch,** Switch name , Initial state (1:pressed, 0:unpressed), Popup info Text, Commands group to execute when button is pressed , Commands group to execute when button is unpressed, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled), Switch width in

	pixels (0:AutoWidth NonZero:Specified width)**]**

Creates an interactive switch (**Top Level**).

Example:

```
[ISwitch, ClickMe, 1, "Info text", [MessageOK, On], [MessageOK, Off]]
```

Creates an interactive switch with "ClickMe" label which will display an "On" message when pressed and an "Off" message when unpressed

- Note that for plugins it is a good idea to include [IEnable] startup code for each switch.

ISubPalette	**[ISubPalette,** Subpalette name, Title mode? (0:Show Title and minimize button(ByDefault) 1:Show Title without minimize button 2:Hide Title), Optional subpalette gray-scale (8-bits) icon (Standard size of 20x20 pixels), Left Inset (0:default), Right Inset (0:default), Left Top (0:default), Right Bottom (0:default)**]**

Adds a subpalette to ZBrush interface. Output: Returns 1 if subpalette added successfully. Returns 0 if subpalette could not be added or if it already exsists.

Example:

```
[ISubPalette, "ZPlugin:My Plugins"]
```

Creates a "My Plugins" subpalette within the ZPlugin palette. This command is essential for creating a zscript plugin.

ButtonFind	**[ButtonFind,** Interface item path, Button name, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled)**]**

Locates a ZBrush interface item (**Top Level**).

Example:

```
[ButtonFind, Document:Width, Text]
```

Locates the width button in the Document menu

ButtonPress	**[ButtonPress,** Interface item path, Button name, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled)**]**

Locates and presses a ZBrush interface item (**Top Level**).

Example:

[ButtonPress, Tool:Sphere3D, Text]

Locates the Sphere3D button in the Tool menu and presses it making the Sphere3D the active tool

ButtonSet	**[ButtonSet,** Interface item path, Value, Button name, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled)**]**

Locates and sets a new value to a ZBrush interface item (**Top Level**).

Example:

[ButtonSet, Document:Width, 123, Text]

Locates the Width slider in the Document menu and enters "123" as its value

ButtonUnPress	**[ButtonUnPress,** Interface item path, Button name, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled)**]**

Locates and unpresses a ZBrush interface item (**Top Level**).

Example:

[ButtonUnPress, Layer:Modifiers:W, Text]

Locates the W button in the Modifiers submenu of the Layer menu and unpresses it

Notes and Messages

Note	**[Note,** "Text line", Optional path1 of an interface item to be pointed out (default:none), Display Duration (in seconds) (0:wait for user action(default), -1:combine with next note command), Popup background color (0x000000<->0xffffff, default:0x606060, 0:No Background), Prefered distance of the note from the specified interface item (default:48), Prefered Note width (in pixels, default:400), optional marked windows fill color (0x000000<->0xffffff or (blue+(green*256)+(red*65536))(Omitted value:No fill)), Frame horizontal size (1:Max width (default)), Frame vertical size (1:Max height (default)), Frame left side (0:left (default),

	.5:center, 1:right) Omit value for horizontal autocentering, Frame top side (0:top (default), .5:center, 1:bottom)Omit value for vertical auto-centering, Optional icon file name**]**

Displays a note to the user. Output: If the note has UI buttons then the return value of the pressed buttons (1=1st button, 2=2nd button ...), otherwise the return value will be zero (**Sub-Level** only).

Example:

```
[Note, "Hello There"]
```

Displays a message to the user with "Hello There" as the text.

NoteBar	**[NoteBar,** The Message that will be shown (use empty string to clear current note), Optional progress-bar value (0:Min, 1:Max)**]**

Displays a note in progress bar (**Sub-Level** only).

Example:

```
[NoteBar, ”ZScript is calculating, Please wait..."]
```

Displays a progress bar note "ZScript is calculating. Please wait...".

NoteIButton	**[NoteIButton,** Button name, Optional button icon, Initially Pressed? (default:unpressed), Initially Disabled? (default:enabled), Optional button H relative position (Positive value:offset from left, Negative value:offset from right, 0:automatic), Optional button V relative position (Positive value:offset from top, Negative value:offset from bottom, 0:automatic), Optional button width in pixels (default:automatic), Optional button height in pixels (default:automatic), Optional button color (0x000000<->0xffffff or (blue+(green*256)+(red*65536))), Optional text color (0x000000<->0xffffff or (blue+(green*256)+(red*65536))), Optional background opacity (default:1), Optional text opacity (default:1) , Optional image opacity (default:1)**]**

Defines a button to be included within the next Note to be shown (**Sub-Level** only).

Example:

```
[NoteIButton, "OK"]
```

Defines an "OK" note button.

NoteIGet	**[NoteIGet,** Note-button index (1:1st) or its name**]**

Returns the value of am NoteIButton which was shown in the last displayed Note. Output: The item value.

Example:

```
[NoteIGet, 1]
```

Returns the value of the 1st note button or switch.

```
[NoteIGet, "Double"]
```

Returns the value of the note button or switch named "Double".

NoteISwitch	**[NoteISwitch,** Switch name, Optional button icon, Initially Pressed? (default:unpressed), Initially Disabled ? (default:enabled), Optional button H relative position (Positive value:offset from left, Negative value:offset from right, 0:automatic), Optional button V relative position (Positive value:offset from top, Negative value:offset from bottom, 0:automatic), Optional button width in pixels (default:automatic), Optional button height in pixels (default:automatic), Optional button color (0x000000<->0xffffff or (blue+(green*256)+(red*65536))), Optional text color (0x000000<->0xffffff or (blue+(green*256)+(red*65536))), Optional background opacity (default:1), Optional text opacity (default:1), Optional image opacity (default:1)**]**

Define a switch-button to be included within the next Note to be shown (**Sub-Level** only).

Example:

```
[NoteISwitch, "Double Sided"]
```

Defines a "Double Sided" note switch-button.

MessageOK	**[MessageOK,** The Message that will be shown, The Title of the message**]**

Displays a user message with a single OK button

Example:

```
[MessageOK, "Hello There"]
```

Displays a message to the user with "Hello There" as the text, and waits for the user to click the "OK" button (**Sub-Level** only).

MessageOKCancel	**[MessageOKCancel,** The Message that will be shown, The Title of the message**]**

Displays a user message with CANCEL and OK buttons Output: Returns the button that the user clicked. (0=CANCEL, 1=OK)(**Sub-Level** only).

Example:

```
[MessageOkCancel, "Delete this image?"]
```

Displays a message to the user with "Delete this image?" as the text, and waits for the user to click the "OK" or "Cancel" button.

MessageYesNo	**[MessageYesNo,** The Message that will be shown, The Title of the message**]**

Displays a user message with YES and NO buttons Output: Returns the button that the user clicked (0=NO, 1=YES)(**Sub-Level** only).

Example:

```
[MessageYesNo, "Are you sure?"]
```

Displays a message to the user with "Are you sure?" as the text, and waits for the user to click the "YES" or "NO" button.

MessageYesNoCancel	**[MessageYesNoCancel,** The Message that will be shown, The Title of the message**]**

Displays a user message with YES, NO and CANCEL buttons Output: Returns the button that the user clicked (0=NO, 1=YES CANCEL=-1)(**Sub-Level** only).

Example:

```
[MessageYesNoCancel, "Are you sure?"]
```

Displays a message to the user with "Are you sure?" as the text, and waits for the user to click the "YES", "NO" or "CANCEL" button.

Working in the Canvas

CanvasClick	**[CanvasClick,** X1, Y1, X2, Y2, X3, Y3, X4, Y4, X5, Y5, X6, Y6, X7, Y7, X8, Y8**]**

Emulates a click within the current canvas area

Example:

```
[CanvasClick, 10, 10, 20, 20]
```

Emulates a canvas click at 10, 10 with a drag to 20, 20 before releasing the mouse button

CanvasGyroHide	**[CanvasGyroHide]**

Hides the Transformation Gyro

Example:

```
[CanvasGyroHide]
```

Hides the Transformation Gyro until the next `[CanvasGyroShow]` **is encountered**

CanvasGyroShow	**[CanvasGyroShow]**

Shows the Transformation Gyro

Example:

```
[CanvasGyroShow]
```

Shows the Transformation Gyro hidden by a previous `[CanvasGyroHide]`

CanvasPanGetH	**[CanvasPanGetH]**

Returns the H pan value of the active document view Output: The current H Pan value.

Example:

```
[CanvasPanGetH]
```

Returns the Horizontal pan value of the active document view

	[CanvasPanGetV]

CanvasPanGetV	

Returns the V pan value of the active document view Output: The current V Pan value.

Example:

```
[CanvasPanGetV]
```

Returns the Vertical pan value of the active document view

CanvasPanSet	**[CanvasPanSet,** H value (0:left of document), V value (0:top of document)**]**

Pans (Scrolls) the active document view

Example:

```
[CanvasPanSet, 320, 240]
```

Scrolls the center of a 640x480 canvas to be at the center of the document view

```
[CanvasPanSet, 0, 0]
```

Scrolls the top left corner of the canvas to be at the center of the document view

CanvasStroke	**[CanvasStroke,** StrokeData, Delayed update until end of stroke, Rotation, HScale, VScale, HOffset, VOffset**]**

Emulates a brush stroke within the current canvas area

Example:

```
[CanvasStroke, [StrokeGetLast]]
```

Replays the last stroke

CanvasStrokes	**[CanvasStrokes,** StrokesData, Delayed update until end of stroke, Rotation, HScale, VScale, HOffset, VOffset, HRotateCenter, VRotateCenter**]**

Emulates multiple brush strokes within the current canvas area

Example:

```
[CanvasStrokes, [Var, loadedStrokes]]
```

Replays "loadedStrokes" in the canvas area

CanvasZoomGet	**[CanvasZoomGet]**

Returns the zoom value of the active document view Output: The current zoom value.

Example:

```
[CanvasZoomGet]
```

Returns the zoom value of the active document view

CanvasZoomSet	**[CanvasZoomSet,** Zoom factor**]**

Sets the zoom factor of the active document view

Example:

```
[CanvasZoomSet, 2]
```

Sets the zoom value to 2 (each Pixol is shown twice as large)

```
[CanvasZoomSet, .5]
```

Sets the zoom value to .5 (half-antialiased zoom mode)

PixolPick	**[PixolPick,** Component Index: 0:CompositeColor (0x000000<->0xffffff or (red*65536+green*256+blue)); 1:Z(-32576 to 32576); 2:Red(0 to 255); 3:Green(0 to 255); 4:Blue(0 to 255); 5:MaterialIndex(0 to 255); 6:XNormal(-1 to 1); 7:YNormal(-1 to 1); 8:ZNormal(-1 to 0), H Position, V Position**]**

Retrieves information about a specified Pixol Output: The value of the specified Pixol

Example:

```
[PixolPick, 1, 10, 20]
```

Returns the Z(depth) value at 10, 20 canvas position.

StrokeGetInfo	**[StrokeGetInfo,** Stroke-type Variable, Info number, Point index (0 based)**]**

Retrieves the information from a specified Stroke-type Variable Output: StrokeInfo result

Info number: 0=PointsCount 1=IndexedHPos 2=IndexedVPos 3=IndexedPressure 4=MinH 5=MinV 6=MaxH 7=MaxV 8=MaxRadius 9=MaxRadiusPointIndex 10=MaxDeltaH 11=MaxDeltaV 12=Total Distance 13=Twirl Count 14=DeducedZValue 15=IndexedkeyPress

Example:

```
[StrokeGetInfo, [StrokeGetLast], 0]
```

Returns the number of points in the last drawn brush stroke

StrokeGetLast	**[StrokeGetLast]**

Retrieves the last drawn brush stroke Output: StrokeData

Example:

```
[CanvasStroke, [StrokeGetLast]]
```

Replays the last drawn brush stroke

```
[CanvasStroke, [StrokeGetLast], 0, 90, 2, 2]
```

Replays the last drawn brush stroke rotated 90 degrees and scaled x2.

StrokeLoad	**[StrokeLoad,** FileName(.txt)**]**

Loads a brush-stroke text file Output: StrokeData

Example:

```
[VarSet, Strokel, [StrokeLoad, "Star.txt"]]
```

Loads the "Star.txt" file, creates a BrushStroke object and assigns it to "Strokel" variable.

```
[CanvasStroke, [StrokeLoad, "Star.txt"]]
```

Loads the "Star.txt" file, creates a BrushStroke object and applies it to the canvas.

StrokesLoad	**[StrokesLoad,** FileName(.txt)**]**

Loads a brush-strokes text file Output: StrokesData

Example:

```
[VarSet, Strokel, [StrokesLoad, "Star.txt"]]
```

Loads the "Star.txt" file, creates a BrushStrokes object and assigns it to "Strokel" variable.

```
[CanvasStroke, [StrokesLoad, "Star.txt"]]
```

Loads the "Star.txt" file, creates a BrushStrokes object and applies it to the canvas.

TransformGet	[TransformGet, xPosition, yPosition, zPosition, xScale, yScale, zScale, xRotate, yRotate, zRotate]

Gets current transformation values (**Sub-Level** only).

Example:

```
[TransformGet, xPos, yPos, zPos, xSc, ySc, zSc, xRot, yRot, zRot]
```

sets the variables xPos, yPos, zPos, xSc, ySc, zSc, xRot, yRot and zRot to the 3D Position, Scale and Rotation values of the last drawn object or current floating object.

TransformSet	[TransformSet, xPosition, yPosition, zPosition, xScale, yScale, zScale, xRotate, yRotate, zRotate]

Sets new transformation values (**Sub-Level** only).

Example:

```
[TransformSet, (Document:Width*.5), (Document:Height*.5), 0, 100, 100, 100, 0, 0, 0]
```

sets the 3D values of the last drawn object or current floating object to: Position - XY center of the canvas and Z depth 0 Scale - uniform scale of 100 Rotation - default rotation of 0, 0, 0

Working with ZBrush

IClick	[IClick, Interface item path, X1, Y1, X2, Y2, X3, Y3, X4, Y4, X5, Y5, X6, Y6, X7, Y7]

Emulates a click within a specified ZBrush interface item (**Sub-Level** only).

Example:

```
[IClick, LIGHT:Intensity, 55, 10]
```

Emulates a click at 55, 10 position

[IClick, LIGHT:Intensity, 55, 10, 10, 20, 10]

Emulates a click at 10, 10 with a drag to 20, 10 before releasing the mouse button

IClose	**[IClose,** Interface item path, Show Zoom Rectangles?, Target parent window?**]**

Closes an interface item.

Example:

[IClose, ZScript:Play]

deletes the ZScript:Play window

IColorSet	**[IColorSet,** Red (0-255), Green (0-255), Blue (0-255)**]**

Sets the active color to a new value

Example:

[IColorSet, 255, 0, 0]

sets the main interface active color to red

IConfig	**[IConfig,** ZBrush version-configuration]

Sets ZBrush internal version-configuration

Example:

[IConfig, 2.0]

sets the interface to 2.0 configuration

[IConfig, 3.1]

sets the interface to 3.1 configuration

IDialog	

Command currently disabled

IDisable	**[IDisable,** Window path, Window ID or relative windowID(-100<->100)**]**

Disables a ZScript interface item (can only be used for ZScript-generated interface items)

Example:

```
[IDisable, Zscript:DoIt]
```

Disables the "DoIt" ZScript interface item

```
[IDisable, l]
```

Disables the next interface item in the ZScript window

IEnable	**[IEnable,** Window path, Window ID or relative windowID(-100<->100)**]**

Enables a ZScript interface item (can only be used for ZScript-generated interface items)

Example:

```
[IEnable, ZScript:DoIt]
```

Enables the "DoIt" ZScript interface item

```
[IEnable, 1]
```

Enables the next interface item in the ZScript window

IExists	**[IExists,** Interface item path**]**

Verifies that a specified interface item exists. Output: 1 if item exists, 0 otherwise

Example:

```
[IExists, TOOL:Sphere3D]
```

returns 1 if TOOL:Sphere3D exists, returns 0 otherwise.

IFadeIn	**[IFadeIn,** Fade in speed in secs. (default:.5 secs)**]**

Fades in ZBrush window from black.

Example:

```
[IFadeIn, .35]
```

fade in from black, speed = 0.35 seconds .

IFadeOut	[IFadeOut, Fade out speed in secs. (default:.5 secs)]

Fades out ZBrush window to black.

Example:

```
[IFadeOut, .35]
```

fade out to black, speed = 0.35 seconds .

IGet	[IGet, Interface item path]

Returns the current value of a ZBrush or ZScript interface item Output: The item value

Example:

```
[IGet, Draw:Width]
```

Returns the current value of the Width slider in the Draw menu

IGetFlags	[IGetFlags, Interface item path]

Returns the status flags of the specified interface item Output: The flags

Example:

```
[IGetFlags, windowID]
```

Returns the info of specified windowID interface item

IGetHotkey	[IGetHotkey, Interface item path]

Returns the hotkey of the specified interface item Output: The Hotkey

Example:

```
[IGetHotkey, windowID]
```

Returns the hotkey of specified windowID interface item

IGetID	[IGetID, Interface item path]

Returns the window ID code of the specified interface item Output: The Title

Example:

```
[IGetID, Tool:LoadTool]
```

Returns the id code of the Tool:LoadTool interface item

IGetInfo	[IGetInfo, Interface item path]

Returns the info (popup info) of the specified interface item Output: The info

Example:

```
[IGetInfo, windowID]
```

Returns the info of specified windowID interface item

IGetMax	[IGetMax, Interface item path]

Returns the maximum possible value of a ZBrush or ZScript interface item Output: The item maximum value

Example:

```
[IGetMax, Draw:Width]
```

Returns the maximum value of the Width slider in the Draw menu

IGetMin	[IGetMin, Interface item path]

Returns the minimum possible value of a ZBrush or ZScript interface item Output: The item minimum value

Example:

```
[IGetMin, Draw:Width]
```

Returns the minimum value of the Width slider in the Draw menu

IGetSecondary	[IGetSecondary, Interface item path]

Returns the the secondary value of a 2D interface item Output: The item value

Example:

```
[IGetSeconday, Light:LightPlacement]
```

Returns the secondary value of the Light:LightPlacement control

IGetStatus	`[IGetStatus, Interface item path]`

Returns the Enabled/Disabled status of a ZBrush or ZScript interface item Output: The item status 0=Disabled 1=Enabled

Example:

```
[IGetStatus, Transform:Move]
```

Returns the current status of the Move button in the Transform menu

IGetTitle	`[IGetTitle, Interface item path, Return full path? (0:no nonZero:yes)]`

Returns the title of the specified interface item Output: The Title of the button

Example:

```
[IGetTitle, windowID]
```

Returns the title of specified windowID interface item

IHeight	`[IHeight, Interface item path]`

Returns the pixel-height of an interface item. Output: The height of the interface item.

Example:

```
[IHeight, Tool:SimpleBrush]
```

Returns the height of the "Tool:SimpleBrush" interface item

IHide	`[IHide, Interface item path, Show Zoom Rectangles?, Target parent window?]`

Hides an interface item.

Example:

[IHide, Draw:Width]

Hides the "Draw:Width" window

IHPos	**[IHPos,** Interface item path , Global coordinates?(set value to non-zero for global coordinates; default:Canvas coordinates)**]**

Returns the H position of the interface item in Canvas or Global coordinates. Output: The H position of the interface item.

Example:

[IHPos, Draw:Width, 1]

Returns the H position of the "Draw:Width" interface Item in Global coordinates

IKeyPress	**[IKeyPress,** The key to press (with an optional CTRL/CMD, ALT/OPT, SHIFT or TAB combination.) , Commands group to execute while the key is pressed , Optional H cursor position prior to key press , Optional V cursor position prior to key press**]**

Simulates a key press

Example:

[IKeyPress, 'x']

Simulates "x" key press

[IKeyPress, CTRL+'z']

Simulates "Ctrl+z" key press

ILock	**[ILock,** Window path, Window ID or relative windowID(-100<->100)**]**

Locks an interface item.

Example:

[ILock, ZScript:DoIt]

locks the "DoIt" Zscript window interface item .

[ILock, 1]

locks the next interface item .

IMaximize	**[IMaximize,** Interface item path, Maximizeall sub palettes? (0:no, NonZero:yes)**]**

Locates an interface item and (if possible) maximize its size.

Example:

```
[IMaximize, Tool:, 1]
```

Expand the TOOL palette and all of its sub palettes

IMinimize	**[IMinimize,** Interface item path, Minimize all sub palettes? (0:no, NonZero:yes)**]**

Locates an interface item and (if possible) minimize its size.

Example:

```
[IMinimize, Tool:, 1]
```

Closes the TOOL palette and all of its sub palettes

IModGet	**[IModGet,** Interface item path**]**

Returns the current modifiers binary state of a ZBrush or ZScript interface item Output: The item value

Example:

```
[IModGet, Tool:Modifiers:Deformation:Rotate]
```

Returns the current modifiers of the Rotate slider in the Tool menu. Each modifier is identified by its binary value such as lst=1, 2nd=2, 3rd=4, 4th=8

IModSet	**[IModSet,** Interface item path, value**]**

Sets the modifiers binary value of a ZBrush or a ZScript interface item

Example:

```
[IModSet, Tool:Modifiers:Deformation:Rotate, 2]
```

Sets the modifiers of the Rotate slider in the Tool menu to 2. Each modifier is identified by its binary value such as lst=1, 2nd=2, 3rd=4, 4th=8

IPress	[IPress, Interface item path]

Presses a ZBrush or ZScript interface item (**Sub-Level** only).

Example:

```
[IPress, Tool:Cube3D]
```

Presses the Cube3D button in the Tool menu making the Cube3D the active tool

IReset	[IReset, Optional item to reset (default:All). (0:All, 1:Interface, 2:Document, 3:Tools, 4:Lights, 5:Materials, 6:Stencil) , Optional ZBrush version-configuration]

Interface Reset. Output: Returns the button that the user clicked (0=NO, 1=YES) (**Sub-Level** only).

Example:

```
[IReset]
```

Resets the interface to a default state

IsDisabled	[IsDisabled, Interface item path]

Returns 1 if the specified ZBrush or ZScript interface item is currently disabled, returns 0 otherwise Output: The item 'Disabled' status (1=Disabled 0=Enabled)

Example:

```
[IsDisabled, Transform:Move]
```

Returns 1 if the "Transform:Move" interface item is currently disabled, returns 0 otherwise

IsEnabled	[IsEnabled, Interface item path]

Returns 1 if the specified ZBrush or ZScript interface item is currently enabled, returns 0 otherwise Output: The item 'Enabled' status (1=Enabled 0=Disabled)

Example:

```
[IsEnabled, Transform:Move]
```

Returns 1 if the "Transform:Move" interface item is currently enabled, returns 0 otherwise

ISet	**[ISet,** Interface item path, value, Secondary value**]**

Sets a new value to a ZBrush or ZScript interface item

Example:

```
[ISet, Draw:Width, 50]
```

Sets the Width slider in the Draw menu to 50

ISetHotkey	**[ISetHotkey,** Interface item path, Hotkey(0:no Hotkey)**]**

Sets the hotkey of the specified interface item

Example:

```
[ISetHotkey, windowID, 'k']
```

Sets "k" as the hotkey for the specified windowID interface item

ISetMax	**[ISetMax,** Interface item path, New max value**]**

Sets the maximum value for an ISlider interface item (can only be used for ZScript-generated interface items)

Example:

```
[ISetMax, Zscript:Counter, 10]
```

Sets the maximum value of "ZScript:Counter" interface item to 10

ISetMin	**[ISetMin,** Interface item path, New min value**]**

Sets the minimum value for an ISlider interface item (can only be used for ZScript-generated interface items)

Example:

```
[ISetMin, Zscript:Counter, 10]
```

Sets the minimum value of "ZScript:Counter" interface item to 10

IShow	**[IShow,** Interface item path, Show Zoom Rectangles?, Target parent window?**]**

Locates an interface item and makes it visible.

Example:

```
[IShow, Draw:Width]
```

Makes the "Draw:Width" item visible. If necessary the "Draw" palette will be opened

IShowActions	**[IShowActions,** The ShowActions status.(0:Disable ShowActions, Positive value:enable show actions, Negative value:Reset ShowActions)**]**

Temorarily sets the status of ShowActions

Example:

```
[IShowActions, 0]
```

Temporarily disables ShowActions

IsLocked	**[IsLocked,** Interface item path**]**

Returns 1 if the specified ZBrush or ZScript interface item is currently locked, returns 0 otherwise

Output: The item 'Locked' status (1=Locked 0=Unlocked)

Example:

```
[IsLocked, Transform:Move]
```

returns 1 if the "Transform:Move" interface item is locked, returns 0 otherwise .

IStroke	**[IStroke,** Interface item path, StrokeData**]**

Emulates a brush stroke within an interface item

Example:

```
[IStroke, [StrokeLoad, "CurvePoints.txt"]]
```

Loads the "Curvepoints.txt" file, creates a BrushStroke and applies it to the interface item

IsUnlocked	**[IsUnlocked,** Interface item path**]**

Returns 1 if the specified ZBrush or ZScript interface item is currently unlocked, returns 0 otherwise

Output: The item 'Unlocked' status (1=Unlocked 0=locked)

Example:

```
[IsUnLocked, Transform:Move]
```

returns 1 if the "Transform:Move" interface item is unlocked, returns 0 otherwise .

IToggle	**[IToggle,** Interface item path**]**

Toggles the state of a ZBrush or ZScript interface item

Example:

```
[IToggle, Draw:ZAdd]
```

Toggles the ZAdd button in the Draw menu turning ZAdd mode on and off

IUnlock	**[IUnlock,** Window path, Window ID or relative windowID(-100<->100)**]**

Unlocks an interface item

Example:

```
[IUnLock, ZScript:DoIt]
```

unlocks the "DoIt" Zscript window interface item .

```
[IUnLock, 1]
```

unlocks the next interface item .

IUnPress	**[IUnPress,** Interface item path**]**

Unpresses a ZBrush or ZScript interface item

Example:

```
[IUnPress, Layer:Modifiers:w]
```

Unpresses the W button in the Modifiers submenu of the Layer menu

IUpdate	**[IUpdate,** Repeat count (default:1), Redraw UI? (default:no, 1:yes)**]**

Updates the ZBrush interface.

Example:

```
[IUpdate, 5]
```

Execute 5 interface-update cycles

IVPos	**[IVPos,** Interface item path , Global coordinates? (set value to non-zero for global coordinates, default:Canvas coordinates)**]**

Returns the V position of the interface item in Canvas or Global coordinates. Output: The V position of the interface item.

Example:

```
[IVPos, Draw:Width, 1]
```

Returns the V position of the "Draw:Width" interface Item in Global coordinates

IWidth	**[IWidth,** Interface item path**]**

Returns the pixel-width of an interface item. Output: The width of the interface item.

Example:

```
[IWidth, Tool:SimpleBrush]
```

with Returns the width of the "Tool:SimpleBrush" interface item.

ZBrushInfo	**[ZBrushInfo,** The info type. (0:version number 1:Demo/Beta/Full 2:Runtime seconds 3:Mem use 4:VMem Use 5:Free Mem 6:operating system (0:PC 1:Mac 2:MacOSX) 7:Unique session ID 8:Total RAM)**]**

Returns ZBrush info. Output: Result value

Example:

```
[ZBrushInfo, 6]
```

returns the user's operating system. Useful if there are different requirements for running the zscript on different operating systems.

ZBrushPriorityGet	[ZBrushPriorityGet]

Returns the task-priority of ZBrush. Output: The current task-priority

Example:

```
[ZBrushPriorityGet]
```

Returns current task-priority.

ZBrushPrioritySet	[ZBrushPrioritySet , The priority. -2:Low, -1:BelowNormal, 0:normal, 1:Above Normal, 2:High]

Sets the task-priority of ZBrush.

Example:

```
[ZBrushPrioritySet, 1]
```

sets ZBrush priority to above normal.

Reading the Mouse

MouseHPos	[MouseHPos, Global coordinates? (set value to non-zero for global coordinates, default:Canvas coordinates)]

Returns the current H position of the mouse in Canvas or Global coordinates. Output: The H position of the mouse

Example:

```
[MouseHPos]
```

Returns the current H position of the mouse

MouseLButton	[MouseLButton]

Returns the current state of the left mouse button Output: Returns 1 if mouse button is pressed, returns zero if unpressed

Example:

[MouseLButton]

Returns 1 if mouse button is pressed, returns zero if unpressed

MouseVPos	**[MouseVPos,** `Global coordinates? (set value to non-zero for global coordinates, default:Canvas coordinates)`**]**

Returns the current V position of the mouse in Canvas or Global coordinates. Output: The V position of the mouse

Example:

[MouseVPos]

Returns the current V position of the mouse

Display in the ZScript Tutorial Window

BackColorSet	**[BackColorSet,** `Red, Green, Blue`**]**

Sets the pen background color (**Top Level**).

Example:

[BackColorSet, 255, 255, 0]

sets the pen background color to yellow

Caption	**[Caption,** `Text`**]**

Displays a text line using the current Caption settings (**Top Level**).

Example:

[Caption, This Is A Caption]

displays "ThisIsACaption" using the Caption settings

FontSetColor	**[FontSetColor,** `Red, Green, Blue`**]**

Sets the color of the zscript window font (**Top Level**).

Example:

`[FontSetColor, 255, 0, 0]`

sets zscript window font to red

`[FontSetColor, 255, 255, 255]`

sets zscript window font to white

FontSetOpacity	**`[FontSetOpacity,` `Opacity``]`**

Sets the opacity of the zscript window font (**Top Level**).

Example:

`[FontSetOpacity, .25]`

sets zscript window font opacity to 25%

`[FontSetOpacity, 1]`

sets zscript window font opacity to 100%

FontSetSize	**`[FontSetSize,` `Size: 1:Small 2:Med 3:Large``]`**

Sets the intensity of the zscript window font (**Top Level**).

Example:

`[FontSetSize, 2]`

sets zscript window font size to medium

FontSetSizeLarge	**`[FontSetSizeLarge]`**

Sets the size of the zscript window font to large (**Top Level**).

Example:

`[FontSetSizeLarge]`

sets zscript window font size to large

FontSetSizeMedium	**[FontSetSizeMedium]**

Sets the size of the zscript window font to medium (**Top Level**).

Example:

```
[FontSetSizeMedium]
```

sets zscript window font size to medium

FontSetSizeSmall	**[FontSetSizeSmall]**

Sets the size of the zscript window font to small.

Example:

```
[FontSetSizeSmall]
```

sets zscript window font size to small

FrontColorSet	**[FrontColorSet,** Description Text, Red, Green, Blue, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled)**]**

Sets the main interface color to a new value (**Top Level**).

Example:

```
[FrontColorSet, Text, 0, 0, 0]
```

sets the ZBrush interface main front color to black

HotKeyText	**[HotKeyText,** Interface item path**]**

Displays a hot-key for the specified interface item (**Top Level**).

Example:

```
[HotKeyText, DOCUMENT:UNDO]
```

displays the hot key of the DOCUMENT:UNDO button using the Hotkey text settings

Image	**[Image,** FileName (.psd .bmp + .pct for Mac Systems), Align (0:center 1:left 2:right), Resized Width**]**

Loads and displays an image (**Top Level**).

Example:

```
[Image, TestImage.psd]
```

Loads and displays TestImage.psd in the zscript window, center-aligned (default) in its original size

ISetStatus	**[ISetStatus,** Interface item path, New status (0:Disable NotZero:Enable)**]**

Enables or Disables a ZScript interface item (can only be used for ZScript-generated interface items).

Example:

```
[ISetStatus, ZScript:DoIt, 1]
```

Enables the "DoIt" ZScript interface item

```
[ISetStatus, ZScript:DoIt, 0]
```

Disables the "DoIt" ZScript interface item

PageSetWidth	**[PageSetWidth,** Preferred PageWidth**]**

Sets the width of the page (**Top Level**).

Example:

```
[PageSetWidth, 300]
```

sets the zscript window display page width to a maximum of 300 pixels

PaintBackground	**[PaintBackground,** Red, Green, Blue**]**

Paints the background using the current background color (**Top Level**).

Example:

```
[PaintBackground, 10, 10, 10]
```

Fills the zscript window background with a dark gray color

PaintBackSliver	**[PaintBackSliver,** height, Red, Green, Blue**]**

Draws a full page-width rectangle using the current background color (**Top Level**).

Example:

```
[PaintBackSliver, 20, 255, 255, 0]
```

Draws a yellow rectangle in the zscript window, full page-width and 20 pixels tall.

PaintPageBreak	**[PaintPageBreak]**

Draws a visual page-break (**Top Level**).

Example:

```
[PaintPageBreak]
```

Draws a default page break in the zscript window, which is a special case of the PaintBackSliver command

PaintRect	**[PaintRect,** Width, height, Red, Green, Blue**]**

Draws a rectangle (in the ZScript window) using the current pen color (**Top Level**).

Example:

```
[PaintRect, 80, 100]
```

Draws a rectangle in the zscript window, 80 pixels wide and 100 pixels tall using the current pen color.

PaintTextRect	**[PaintTextRect,** Width, Height, Text**]**

Draws a rectangle with imbedded text (**Top Level**).

Example:

```
[PaintTextRect, 200, 100, "This is an Example"]
```

Draws a rectange in the zscript window, 200 pixels wide and 100 tall with "This is an Example" inside it.

PD	**[PD]**

Moves the pen position to the beginning of the next line (Same as PenMoveDown).

Example:

```
[PD]
```

moves the pen to the beginning of the next line of the zscript window.

PenMove	**[PenMove,** `Horizontal Offset, Vertical Offset`**]**

Moves the pen a relative distance.

Example:

```
[PenMove, 40, 80]
```

moves the pen 40 pixels to the right and 80 pixels down in the zscript window. Note: useful for arranging interface items in the zscript window.

PenMoveCenter	**[PenMoveCenter]**

Moves the pen position to the horizontal center of the page .

Example:

```
[PenMoveCenter]
```

moves the pen to the center of the zscript window.

PenMoveDown	**[PenMoveDown]**

Moves the pen position to the beginning of the next line .

Example:

```
[PenMoveDown]
```

moves the pen to the beginning of the next line in the zscript window.

PenMoveLeft	**[PenMoveLeft]**

Moves the pen position to the left side of the page .

Example:

```
[PenMoveLeft]
```

moves the pen to the left of the zscript window.

PenMoveRight	[PenMoveRight]

Moves the pen position to the right side of the page .

Example:

```
[PenMoveRight]
```

moves the pen to the right of the zscript window.

PenSetColor	[PenSetColor, Red, Green, Blue]

Sets the pen main color .

Example:

```
[PenSetColor, 128, 128, 128]
```

sets pen color to medium gray.

PropertySet	[PropertySet, The base command name (Title,SubTitle,Caption), Property Index, The new Value]

Modifies the setting of Title, SubTitle and Caption text.

Properties Indexes:

0:FontSetSize(1:small,2:medium,3:large)

1:Alignment(0:center,1:left,2:right)

2:Opacity(0-1)

3:Color(0x000000<->0xffffff)

4:BorderSize

5:BackColor1(0x000000<->0xffffff)

6:GradientMode(-1:None 0:flat 1:HGrad 2:VGrad 3:DHGrad 4:DVGrad)

7:BackColor2(0x000000<->0xffffff)

Example:

```
[PropertySet, Title, 1, 1]
```

sets left alignment for Title settings.

SectionBegin	**[SectionBegin,** Section Title, Initial state (1:Expanded, 0:Collapsed), Popup Info Text, Commands group to execute when expanding to reveal content, Commands group to execute when collapsing to hide content, Initially Disabled? (0:Enabled(ByDefault) NonZero:Disabled)**]**

Begins a collapsible section .

Example:

```
[SectionBegin, Chapterl2]
```

Begins a collapsible section in the zscript window with "Chapterl2" as the Title which will expand/collapse to reveal/hide its contents when pressed.

SectionEnd	**[SectionEnd]**

Ends a collapsible section .

Example:

```
[SectionEnd]
```

ends a previously defined collapsible section in the zscript window.

SubTitle	**[SubTitle,** Text**]**

Displays a text line using the current SubTitle settings .

Example:

```
[SubTitle, "Chapter 23"]
```

displays "Chapter 23" in the zscript window using the SubTitle settings.

TextCalcWidth	**[TextCalcWidth,** The text to be evaluated**]**

Calculates the pixel-width of the specified string Output: Width of text in pixels .

Example:

```
[TextCalcWidth, "Test"]
```

returns the width in pixels of the string "Test".

Title	[Title, Text]

Displays a text line using the current Title settings .

Example:

```
[Title, "Hello"]
```

displays "Hello" in the zscript window using the Title settings.

Variables

VarDef	[VarDef, Variable name, Variable defaultValue]

Defines a variable (advised Top Level).

Example:

```
[VarDef, xPos, 1]
```

Defines a variable with the name "xPos" and initializes it to 1.

```
[VarDef, xPos(100)]
```

Defines a list variable named "xPos" with 100 items. Note: the list index starts at 0, so xPos(99) is the hundredth item.

VarSet	[VarSet, Variable name, New Value]

Sets the value of a named variable (can be placed **anywhere**).

Example:

```
[VarSet, xPos, 42]
```

sets variable "xPos" to 42.

```
[VarSet, xPos, (Document:Width*.5)]
```

sets variable "xPos" to Document:Width multiplied by 0.5.

VarListCopy	[VarListCopy, Destination list, Destination initial index, Source list, Source initial index, Number of items to copy.(if omitted

	`or it is 0, then all items will be copied)]`

Copies items from a source list to a destination list

Example:

```
[VarListCopy, destList, 0, sourceList, 4, 3]
```

copies items 4-6 from sourceList to items 0-2 of destList.

VarLoad	`[VarLoad, Variable name, FileName, Verify only (1:Only Verify that a proper saved variable file exists, 0:(default)Verifies and loads values)]`

Loads variable/s from a file Output: Number of loaded or verfied values

Example:

```
[VarLoad, userData, tempFile]
```

Sets variable named "userData" to value/s loaded from the "tempFile.zvr" file.

VarSave	`[VarSave, Variable name, FileName]`

Saves variable value/s to file Output: Number of saved values

Example:

```
[VarSave, userData, tempFile]
```

Saves the current value/s of "userData" variable to "tempFile.zvr" file.

Strings

StrAsk	`[StrAsk, Initial string, Title]`

Asks user to input a string. Output: Returns the text typed by user or an empty string if canceled.

Example:

```
[StrAsk, "Type text in here", "Please enter a file name"]
```

Displays a text input dialog and returns the string typed by user. Note: include both initial string and title.

StrExtract	**[StrExtract,** Input string, Start character index (0:left), End character index (0:left)**]**

Returns specified portion of the input string Output: The extracted portion of the input string.

Example:

```
[StrExtract, "abcdefgh", 3, 5]
```

Returns the "def" portion of the input string.

StrFind	**[StrFind,** find this string, in this string, Optional start search index (default:0)**]**

Locate a string within a string. Output: Returns the starting index of the 1st string within the 2nd string. returns -1 if not found.

Example:

```
[StrFind, "Br", "ZBrush"]
```

Searches for "Br" within "ZBrush" and returns 1.

```
[StrFind, "Ba", "ZBrush"]
```

Searches for "Br" within "ZBrush" and returns -1 (not found).

StrFromAsc	**[StrFromAsc,** Input Ascii value**]**

Returns the character of the specified Ascii value. Output: The character of the specified Ascii value.

Example:

```
[StrFromAsc, 65]
```

returns "A".

StrLength	**[StrLength,** String to evaluate**]**

Returns the number of characters in the input string. Output: Number of characters in the input string.

Example:

```
[StrLength, "Hello"]
```

Returns 5 as the number of characters in "Hello".

StrLower	**[StrLower,** Input string**]**

Returns the lowercase version of the input string. Output: The lowercase version of the input string.

Example:

```
[StrLower, "ZBrush"]
```

returns "zbrush".

StrMerge	**[StrMerge,** Str 1, Str 2, Optional Str 3, Opt Str 4, Opt Str 5, Opt Str 6, Opt Str 7, Opt Str 8, Opt Str 9, Opt Str 10, Opt Str 11, Opt Str 12**]**

Combines two (or more) strings into one string. Output: The combined string. Note: result string will not exceed 255 characters in length

Example:

```
[StrMerge, "Texture number ", "15", " is selected"]
```

returns the merged string: "Texture number 15 is selected".

```
[StrMerge, ZTool, 27, .ztl]
```

returns the merged string: "ZTool27.ztl".

StrToAsc	**[StrToAsc,** Input string, Optional character offset (default:0)**]**

Returns the Ascii value of a character. Output: The Ascii value of a character.

Example:

```
[StrToAsc, "ZBrush"]
```

returns the Ascii value of "Z".

```
[StrToAsc, "ZBrush", 2]
```

returns the Ascii value of "r".

StrUpper	**[StrUpper,** Input string**]**

Returns the uppercase version of the input string. Output: The uppercase version of the input string.

Example:

```
[StrUpper, "ZBrush"]
```

returns "ZBRUSH".

Files and Filenames

FileExecute	**[FileExecute,** File name including the extension (such as plugin.dll), Routine to call, Optional text input, Optional number input, Optional memory block input, Optional memory block output**]**

Executes the specified plugin file (DLL). Output: Returns the result value which was returned by the executed routine. Returns zero if error

Example:

```
[FileExecute, PluginTest.dll, ShowMsg, "Hi There"]
```

Executed the "ShowMsg" routine of the "PluginTest.dll" plugin

FileExists	**[FileExists,** File name including the extension (such as brush1.ztl)**]**

Check if a specific file exists. Output: Returns 1 if file exists. Returns zero if does not exists

Example:

```
[FileExists, LargeImage.psd]
```

Returns 1 if "LargeImage.psd" exists or zero if file does not exists

FileNameAdvance	**[FileNameAdvance,** Base file name, Number of digits (0-4) (i.e. 3: 001), Add 'Copy' string?(0:no, NonZero:yes)**]**

Increments the index value contained within a filename string Output: Updated file Name.

Example:

```
[FileNameAdvance, "image01.psd"]
```

Adds 1 to the index of the string, i.e. if filename was image0l.psd it will be modified to image02.psd

FileNameAsk	**[FileNameAsk,** Extension list (up to 3 extensions), Default fileName for SaveDialog. Name should be omitted for OpenDialog, Optional dialog title**]**

Asks user for a file name Output: Result file name or an empty string if user canceled operation (**Sub-Level** only).

Example:

```
[FileNameAsk, "DXF(*.dxf)|*.dxf|OBJ(*.obj)|*.obj||", , "Please select a file to load..."]
```

Activates OpenDialog for a *.dxf or *.obj file to load. Sets the dialog title to "Please select a file to load..."

```
[FileNameAsk, *.zvr, tempFile]
```

Activates SaveDialog with default "tempFile.zvr" file name.

FileNameExtract	**[FileNameExtract,** File name (Full path), Component specifier (1:path, 2:name, 4:ext)**]**

Extracts filename components. Output: The extracted filename component/s.

Example:

```
[FileNameExtract, fullFilePath, 2]
```

returns the name component of the input file path

```
[FileNameExtract, fullFilePath, 2+4]
```

returns the name+extension components of the input file path

FileNameGetLastTyped	**[FileNameGetLastTyped]**

Retrieves the latest file name that was typed by the user in a Save/Load action Output: Latest file name that was typed by the user. Returned string will be empty if the user has canceled the action.

Example:

```
[FileNameGetLastTyped]
```

Returns a string with the latest typed filename

FileNameGetLastUsed	[FileNameGetLastUsed]

Retrieves the latest file name that was used (by the user or by ZBrush) in a Save/Load action Output: Latest file name that was used. Returned string will be empty if the user has canceled the action.

Example:

```
[FileNameGetLastUsed]
```

Returns a string with the latest used filename

FileNameMake	[FileNameMake, Base file name, Index, Number of numeric digits to use]

Combines a base filename with an index number Output: Combined file name Variable

Example:

```
[FileNameMake, Image.psd, 12]
```

Creates a string with "Image12.psd" as its value

FileNameResolvePath	[FileNameResolvePath, Local File Name]

Resolves local path to full path Output: Full path.

Example:

```
[FileNameResolvePath, LargeImage.psd]
```

returns the full path of the "LargeImage.psd" file which is located within the same directory as the executing ZScript

```
[FileNameResolvePath, ]
```

returns the full path to the executing ZScript

FileNameSetNext	[FileNameSetNext, File name including the extension (such as .psd). If omitted the stored file name will be cleared.]

Pre-sets the file name that will be used in the next Save/Load action

Example:

```
[FileNameSetNext, LargeImage.psd]
```

sets "LargeImage.psd" as the next file name that will be used in a Save/Load action

```
[FileNameSetNext, [FileNameMake, Image.psd, 23, 4]]
```

sets "Image0023.psd" as the next file name that will be used in a Save/Load action

Calculations

Interpolate	**[Interpolate,** Time (0:AtStart 0.5:half 1:AtEnd), Value1 (Num, VarName or ListName), Value2 (Num, VarName or ListName), Value3 (Num, VarName or ListName), Value4 (Num, VarName or ListName), Angle interpolation (0:no(default), 1:yes)**]**

Performs time-based interpolation Output: Interpolated value or list

Example:

```
[Interpolate, 0.25, startx, endx]
```

returns an interpolated value (startX*(1.0-0.25))+(endX*0.25)

```
[Interpolate, 0.25, list1, list2, list3]
```

returns an interpolated list calculated as a spline at t=.25

Math Operators	**[- * / ^^**

`+` Plus
`-` Minus
`*` Multiplied by
`/` Divided by
`^^` To the Power of

Note: Unlike some programming languages, calculations are always evaluated from left to right. That means

```
2 + 3 * 4
```

evaluates to 20.

To make sure certain parts of the calculation are evaluated first, place them inside parentheses:

```
2 + (3 * 4)
```

evaluates to 14.

Logical Operators	&& !

`&&` AND
`||` OR
`!` NOT

Note: When evaluating several conditions, group using parentheses:

```
(myVar < 8)&&(myVar >=2)
```

```
((myVarA < 8)&&(myVarA >=2)) || (myVarA == (myVarB - 1))
```

Math Functions	

`INT(value)` Integer Portion of value; removes everything after decimal point
`FRAC(value)` Fractional Portion of value; removes everything before decimal point
`ABS(value)` Absolute Value (ignores + or - sign)
`NEG(value)` Changes the + or - sign of value
`MIN(value1, value2)` Finds the lesser of two values
`MAX(value1, value2)` Finds the greater of two values
`SQRT(value)` Square Root of the value
`RAND(value)` Random Number between 0 and value
`IRAND(value)` Random Integer between 0 and value
`SIN(angle)` Trig Sine of the angle, in degrees
`COS(angle)` Trig cosine of the angle, in degrees
`TAN(angle)` Trig Tangent of the angle, in degrees
`ASIN(value)` Trig ArcSine of the value
`ACOS(value)` Trig ArcCosine of the value
`ATAN(value)` Trig ArcTangent of the value
`ATAN2(value, value)` Trig ArcTangent of the value
`LOG(value)` Natural Log of the value
`LOG10(value)` Base 10 Log of the value
`BOOL(value)` Boolean Evaluation

Randomize	**[Randomize,** `Optional seed value (0 to 32767)`**]**

Resets the Rand generator.

Example:

[Randomize]

Resets Random generator.

RGB	**[RGB,** Red, Green, Blue**]**

Combines 3 color-components into one RGB value Output: Combined RGB

Example:

[RGB, 20, 40, 80]

calculates and returns (20*65536 + 40*256 + 80) as a combined RGB value to be used in functions that need combined RGB input.

Val	**[Val,** Variable name**]**

Evaluates the input and returns a numerical value Output: Value of the named variable

Example:

[Val, (xPos*2)+4]

returns the value of variable "xPos" multiplied by 2, then added to 4.

Var	**[Var,** Variable name**]**

Gets the value of a named variable Output: Value of the named variable

Example:

[Var, myString]

returns the value of variable "myString". Useful for clearly specifying when a variable name is being used. The special character # can also be used as in #myString.

VarAdd	**[VarAdd,** Variable name, Value To Add**]**

Adds a value to an existing variable

Example:

[VarAdd, xPos, 42]

Adds 42 to the "xPos" variable.

```
[VarAdd, xPos, (Document:Width*.5)]
```

Adds the value of Document:Width multiplied by 0.5 to the "xPos" variable.

VarDec	`[VarDec, Variable name]`

Subtracts 1 from the value of an existing variable

Example:

```
[VarDec, loopCounter]
```

subtracts 1 from the value of the "loopCounter" variable.

VarDiv	`[VarDiv, Variable name, Value to Divide By]`

Divides an existing variable by a value

Example:

```
[VarDiv, xPos, 42]
```

Divides xPos variable by 42.

```
[VarDiv, xPos, (Document:Width*.5)]
```

Divides xPos variable by the value of Document:Width multiplied by 0.5.

VarInc	`[VarInc, Variable name]`

Adds 1 to the value of an existing variable

Example:

```
[VarInc, loopCounter]
```

adds 1 to the "loopCounter" variable.

VarMul	`[VarMul, Variable name, Value To Multiply]`

Multiplies an existing variable by a value

Example:

```
[VarMul, myVar, 5]
```

Multiplies the "myVar" variable by 5.

VarSize	[VarSize, Variable name]

Returns the number of items in a variable or in a list Output: The number of items in a list or 1 if it is a simple variable

Example:

```
[VarSize, list1]
```

returns the number of items in list1 variable.

VarSub	[VarSub, Variable name, Value To Subtract]

Subtracts a value from an existing variable

Example:

```
[VarSub, xPos, 42]
```

Subtracts 42 from the xPos variable.

```
[VarSub, xPos, (Document:Width*.5)]
```

Subtracts value of Document:Width multiplied by 0.5 from the "xPos" variable.

Controlling Flow

Assert	[Assert, True Or False Evaluation, Message that will be shown if the first input is false (zero)]

(ZScript debugging helper) aborts execution if specified condition is not true

Example:

```
[Assert, [Var, a]<10, "Something is wrong"]
```

Checks the value of variable "a" and if it is smaller than 10 then displays a message "Something is wrong" and aborts the execution of the ZScript

Delay	**[Delay,** Delay (in seconds)**]**

Delays execution of ZScript for specified amount of time (**Sub-Level** only).

Example:

```
[Delay, 1]
```

Delays 1 second

```
[Delay, 0.010]
```

Delays 10 milliseconds

Exit	**[Exit]**

Aborts execution and exits the current ZScript

Example:

```
[Exit]
```

Current executing ZScript will be aborted and exited

If	**[If,** True Or False Evaluation , Commands group to be executed if true (not zero) , Commands group to be executed if false (is zero)**]**

Provides conditional execution of a commands group (can be placed **anywhere**).

Example:

```
[If, MyVariable < 10, [MessageOk, LessThanl0], [MessageOk, l0orMore]]
```

Checks the variable "MyVariable" and displays a message "LessThanl0" if the value is less than 10 and displays "l0orMore" otherwise

IFreeze	**[IFreeze,** Commands group to be executed without updating the interface**]**

Disables interface updates.

Example:

```
[IFreeze, ...]
```

Execute inner commands without updating the interface to increase execution speed. ***Note that if using a loop within IFreeze, [IUpdate] should be put at the start (and inside) the loop to avoid interface issues.

Loop	**[Loop,** RepeatCount, Commands group, Optional loop-counter variable (starts at Zero)**]**

Repeats execution of the specified commands group

Example:

```
[Loop, 5, [MessageOK, Hi]]
```

Displays a message box 5 times

```
[Loop, n, [VarSet, a, a+1]]
```

repeats the process of adding 1 to variable a, n times

LoopContinue	**[LoopContinue]**

Continues execution from the beginning of the current Loop

Example:

```
[LoopContinue]
```

loops to the first command within the current Loop

LoopExit	**[LoopExit]**

Exits the current Loop

Example:

```
[LoopExit]
```

Exits the current Loop. Useful when searching for a particular value which has been found.

RoutineCall	**[RoutineCall,** Name of the routine to be called, Input Var01, Input Var02, Input Var03, Input Var04, Input Var05, Input Var06, Input Var07, Input Var08, Input Var09, Input Var10**]**

Executes the specified defined routine (can be placed **anywhere**).

Example:

```
[RoutineCall, testing]
```

Executes a routine named "testing"

RoutineDef	**[RoutineDef,** Name of the routine, Commands group that will be executed when the routine is called, Input Var01, Input Var02, Input Var03, Input Var04, Input Var05, Input Var06, Input Var07, Input Var08, Input Var09, Input Var10**]**

Defines a named commands group (can be placed **anywhere** but generally **Top Level**).

Example:

```
[RoutineDef, testing, [MessageOk, Hi][MessageOk, There]]
```

Creates a routine named "testing" that when called will display 2 messages to the user ("Hi" and then "There").

Sleep	**[Sleep ,** Sleep amount in seconds, Commands group to execute when awaken, Optional event (default:1) (1:Timer 2:Mouse Moved 4:LButton down 8:LButton up 16:KeyDown 32:keyUp 64:ModifierKeyDown 128:ModifierKeyUp 256:Startup 512:Shut down 1024:InterfaceItem pressed/unpressed 2048:tool selected 4096:texture selected 8192:alpha selected), Optional output variable which will contain the event code that has awaken the ZScript, Optional output variable which will contain the ID of the window pointed by the mouse**]**

Exists ZScript and be awaken by specified event (can be placed **anywhere**).

Example:

```
[Sleep, 100, [Note, "LButton pressed"], 4]
```

Sleeps until awakened when left mouse button clicked.

SleepAgain	**[SleepAgain ,** Optional new Sleep amount in seconds (default:unchanged), Optional event (default:unchanged) (1:Timer 2:Mouse Moved 4:LButton down 8:LButton up 16:KeyDown 32:keyUp 64:ModifierKeyDown 256:Startup 512:Shut down 1024:InterfaceItem post pressed/unpressed 2048:tool selected 4096:texture selected 8192 alpha selected)**]**

Exists ZScript and continues the Sleep command (**Sub-Level** only).

Example:

```
[SleepAgain]
```

Sleeps again.

<zscriptinsert>	**<zscriptinsert,** "Filename.txt" **>**

Inserts all the text and commands of an entire ZScript file. Not strictly a zscript command, it is the only one that does not have square brackets.

Example:

```
<zscriptinsert, "MyZscript.txt">
```

Inserts the entire contents of *MyZScript.txt* at that point of the zscript. When the zscript is compiled the inserted zscript is included and no further reference is made to the separate file. Useful in some circumstances but can make code difficult to understand. Also note that commenting out can have unpredictable results.

Memory Blocks

MemCopy	**[MemCopy,** From Mem block identifier, From offset, To Mem block identifier, To offset, Number of bytes to move (if omitted, max possible number of bytes will be copied)**]**

Copies data from one memory block into another. Output: Returns the number of bytes moved. (-1 indicates an error)

Example:

```
[MemCopy, FromMemBlock, 1000, ToMemBlock, 2000, 10000]
```

Moves 10, 000 bytes from FromMemBlock offset 1000 to ToMemBlock offset 2000. Returns the mumber of bytes moved.

MemCreate	**[MemCreate,** Mem block identifier, Mem block requested size, Initial fill? (omitted: noFill - faster to create)**]**

Creates a new memory block. Output: Returns the size of the new memory block or error code...0=Error -1=Memory already exists -2=Can't create memory block.

Example:

```
[MemCreate, myTempData, 1000, 0]
```

Creates a new data block named myTempData of 1000 bytes in size, clears it to 0 and returns the Mem size. Returns 0 if data block could not be created.

Example:

```
[MemCreate, ZProjectMem_MyScriptData, 1000, 0]
```

Creates a new data block named **ZProjectMem_MyScriptData** of 1000 bytes in size which will be stored with the ZBrush project file (.ZPR) when saved. Memblocks that begin with **ZProjectMem**_ and are no larger than 100,000 bytes will be stored with the project file. All memblocks beginning **ZProjectMem**_ are relaced or deleted when a new project is loaded.

MemCreateFromFile	**[MemCreateFromFile,** Mem block identifier, File name including the extension (such as brush1.ztl), Optional start file offset for partial file read (default:0), Optional max bytes to read (default:all file)**]**

Creates a new memory block from a disk file. Output: Returns the size of the new memory block or error code...0=Error -1=Memory already exists -2=Can't create memory block -3=File not found.

Example:

```
[MemCreateFromFile, myTempData, mesh.obj]
```

Loads the content of "mesh.obj" file into a new data block named "myTempData" and returns the memory block size. Returns zero if error encountered.

MemDelete	**[MemDelete,** Data block identifier**]**

Deletes a memory block. Output: Returns the size of the deleted memory block. Returns 0 if memory block could not be found.

Example:

```
[MemDelete, myTempData]
```

Deletes "myTempData" memory block. Be sure to delete memory blocks when you have finished with them, so as to free memory.

MemGetSize	**[MemGetSize,** Memory block identifier**]**

Returns the size of a memory block (Also useful for determining if a memory block already exists. Output: Returns the size of the memory block. Returns 0 if data block could not be found.

Example:

```
[MemGetSize, myTempData]
```

Returns the size of the "myTempData" memory block.

MemMove	**[MemMove,** Mem block identifier, From offset, To offset, Number of bytes to move**]**

Move data within an existing memory block. Output: Returns the number of bytes moved.

Example:

```
[MemMove, myTempData, 1000, 2000, 10000]
```

Moves 10, 000 bytes from offset 1000 to start at offset 2000. Returns the number of bytes moved.

MemMultiWrite	**[MemMultiWrite,** Mem block identifier, Value to write, Data format (0:omitted:float 1:signed char 2:unsigned char 3:signed short 4:unsigned short 5:signed long 6:unsigned long 7:fixed16 (16.16)), Offset (in bytes) into memory block, Repeat count, Offset (in bytes) to subsequent writes**]**

Write data to a memory block. Output: Returns the number of actual bytes written

Example:

```
[MemMultiWrite, myTempData, 4, 12, 3, 5, 100]
```

Writes 5 times the value '4' as "signed short" value into "MyTempData" starting at memory offsets "12", "112", "212", "312" and "412".

MemRead	**[MemRead,** Mem block identifier, Read variable, Data format (0:omitted:float 1:signed char 2:unsigned char 3:signed short 4:unsigned short 5:signed long 6:unsigned long 7:fixed16 (16.16)), Offset (in bytes) into memory block**]**

Reads data from a memory block. Output: Returns the number of actual bytes read

Example:

```
[MemRead, myTempData, width, 12, 3]
```

Reads the value from "MyTempData" at memory offset "12" as "signed short" and stores in variable "width".

MemReadString	**[MemReadString,** Mem block identifier, The string variable, Offset (in bytes) into memory block, Break at line end? (default:no), Skip white space? (default:no), Max read length 1 - 255(default)**]**

Reads a string from a memory block. Output: Returns the number of bytes scanned. (may be larger than the actual bytes read)

Example:

```
[MemReadString, myTempData, tempText, 12, 1]
```

Reads a string from "myTempData" memory block to variable "tempText", starting at memory offset l2 and break at the end of the line.

MemResize	**[MemResize,** Mem block identifier, New size, Optional byte value to fill the newly added memory? (omitted:no)**]**

Resizes an exsiting memory block. Output: Returns the new size of the memory block. Zero indicates an error.

Example:

```
[MemResize, myTempData, 1000]
```

Resizes the memory block "myTempData" to 1000 bytes in size. Returns 0 if data block could not be resized.

MemSaveToFile	**[MemSaveToFile,** Mem block identifier, File name including the extension (such as brush1.ztl), Overwrite if exists? Set to a value (including 0) to save the file even if an identically named file already exists on disk - Default (no argument): do not overwrite**]**

Saves an exisiting memory block to a disk file. Output: Returns the size of the new memory block or error code...0=Error -1=Memory does not exists -2=File already exits -3=File write error.

Example:

```
[MemSaveToFile, myTempData, "mesh.obj"]
```

Saves the content of "myTempData" memory block into a disk file named "mesh.obj" and returns the number of written bytes. Doesn't overwrite as existing file of the same name. Returns zero if error encountered.

```
[MemSaveToFile, myTempData, "mesh.obj",1]
```

Saves the content of "myTempData" memory block into a disk file named "mesh.obj" and returns the number of written bytes. Overwrites an existing file of the same name. Returns zero if error encountered.

MemWrite	**[MemWrite,** Mem block identifier, Value to write, Data format (0:omitted:float 1:signed char 2:unsigned char 3:signed short 4:unsigned short 5:signed long 6:unsigned long 7:fixed16 (16.16)), Offset (in bytes) into memory block**]**

Write data to a memory block. Output: Returns the number of actual bytes written

Example:

```
[MemWrite, myTempData, 4, 12, 3]
```

Writes the value "4" as "signed short" value into "MyTempData" starting at memory offsets "12".

MemWriteString	**[MemWriteString,** Mem block identifier, The string, Offset (in bytes) into memory block, Write terminating zero char (if omitted:yes)**]**

Writes a string into a memory block. Output: Returns the number of bytes written. (including the terminating zero)

Example:

```
[MemWriteString, myTempData, "Hello There", 12]
```

Writes "Hello There" string starting at memory offset l2 and break at the end of the line.

MTransformGet	**[MTransformGet,** Mem block identifier, Optional variable index (default:0)**]**

Gets current transformation values into an existing memory block (**Sub-Level** only).

Example:

```
[MTransformGet, MyDataBlock, 1]
```

store the current 9 transformation values into "MyDataBlock" memory block starting at variable index 1.

MTransformSet	**[MTransformSet,** Mem block identifier, Optional variable index (default:0)**]**

Sets new transformation values from an existing memory block (**Sub-Level** only).

Example:

```
[MTransformSet, MyDataBlock, 1]
```

Sets all 9 transformation values from "MyDataBlock" memory block starting at variable index 1

MVarDef	**[MVarDef,** Mem block identifier, Mem block variables count, Initial fill? (omitted:noFill - faster to create)**]**

Defines a new variables memory block. Output: Returns the variables count of the new memory block or error code...0=Error -1=Memory already exists -2=Can't create memory block.

Example:

```
[MVarDef, myTempData, 1000, 0]
```

Creates a new data block named myTempData of 1000 variables in size, clear it to 0 and return the variables count. Returns 0 if data block could not be created.

MVarGet	**[MVarGet,** Mem block identifier, Variable index (0 based)**]**

Reads a float value from a memory block. Output: Returns the float value.

Example:

```
[MVarGet, myTempData, 1]
```

Returns the 2nd float value from the "MyTempData" memory block.

MVarSet	**[MVarSet,** Mem block identifier, Variable index (0 based), The value to write**]**

Writes a float value to a memory block. Output: Returns the old value of the variable.

Example:

```
[MVarSet, myTempData, 1, 4]
```

Sets the 2nd float value of the "MyTempData" memory block to 4.

SoundPlay	**[SoundPlay,** Mem block identifier, Optional play mode (0:Play once, don't wait for completion (default); 1:Play once, wait for completion; 2:Play loop, don't wait for completion)**]**

Plays the sounds loaded into a specified memory block. Output: Returns the zero if command executed successfully (**Sub-Level** only).

Example:

```
[SoundPlay, SayHello]
```

Plays the "SayHello" memory block.

SoundStop	**[SoundStop,** Mem block identifier**]**

Stops the currently specified sound. Output: Returns the zero if command executed successfully (**Sub-Level** only).

Example:

```
[Soundstop, SayHello]
```

Stops playback of the "SayHello" memory block.

Tools and SubTools

GetActiveToolPath	**[GetActiveToolPath]**

Returns the full path of the active tool (**Sub-Level** only). Output: The path of the active tool.

SubToolSelect	**[SubToolSelect,** Subtool Index (zero based).**]**

Selects the specified subtool index (**Sub-Level** only). Output: Returns zero if OK, -1 if error.

Example:

```
[SubToolSelect, 4]
```

Selects the 5th subtool.

SubToolLocate	**[SubToolLocate,** Unique Subtool ID**]**

Locates a subtool by the specified unique ID (**Sub-Level** only). Output: Returns the index of the located subtool or -1 if error.

SubToolGetID	**[SubToolGetID,** Subtool Index (zero based). If omited then use the currently selected subtool.**]**

Returns the unique subtool ID (**Sub-Level** only). Output: Returns the unique subtool ID or zero if error. *Note that **duplicates** of meshes have the same ID.*

Example:

```
[SubToolGetID, 4]
```

Returns the unique ID for the 5th subtool.

SubToolGetActiveIndex	**[SubToolGetActiveIndex]**

Returns the index of the active subtool (**Sub-Level** only). Output: Returns the index of the active subtool (zero based).

SubToolGetCount	**[SubToolGetCount]**

Returns the number of subtools in the active tool (**Sub-Level** only). Output: Returns the number of subtools. Return 0 if error.

ToolGetActiveIndex	**[ToolGetActiveIndex]**

Returns the index of the active tool (**Sub-Level** only). Output: Returns the index of the active tool (zero based).

ToolGetCount	**[ToolGetCount]**

Returns the number of available tools (**Sub-Level** only). Output: Returns the number of available tools.

ToolGetPath	**[ToolGetPath,** Tool Index (zero based). If omited then use the currently selected tool.**]**

Returns the file path or name of the specified tool (**Sub-Level** only). Output: Result path (without the .ztl). Empty if error.

Example:

```
[ToolGetPath, 4]
```

Returns the path of the 5th tool

ToolGetSubToolID	**[ToolGetSubToolID,** Tool Index (zero based). If omited then use the currently selected tool.,Subtool Index (zero based). If omited then uses the selected subtool.**]**

Returns the unique subtool ID (**Sub-Level** only). Output: Returns the unique subtool ID or zero if error. *Note that **duplicates** of meshes have the same ID.*

Example:

```
[ToolGetSubToolID, 1, 4]
```

Returns the unique subtool ID of the 5th subtool in the 2nd tool.

ToolGetSubToolsCount	**[ToolGetSubToolsCount,** Tool Index (zero based). If omited then use the currently selected tool.**]**

Returns the number of subtools in the specified tool index (**Sub-Level** only). Output: Returns the number of subtools. Return 0 if error.

Example:

```
[ToolGetSubToolsCount, 4]
```

Returns the subtool count for the 5th tool ...

ToolLocateSubTool	**[ToolLocateSubTool,** Unique Subtool ID, Optional subtool index result**]**

Locates a subtool by the specified unique ID (**Sub-Level** only). Output: Returns the index of the located tool and subtool or -1 if error.

ToolSelect	**[ToolSelect,** Tool Index (zero based).**]**

Selects the specified tool index (**Sub-Level** only). Output: Returns zero if OK, -1 if error.

Example:

```
[ToolSelect, 4]
```

Selects the 5th tool.

ToolSetPath	**[ToolSetPath,** Tool Index (zero based). If omited then use the currently selected tool., New Path. Path extension (such as .ztl) will be omited.**]**

Sets the file path or name of the specified tool (**Sub-Level** only). Output: Returns zero if OK, -1 if error.

Example:

```
[ToolSetPath,, [FileNameAdvance,[ToolGetPath]]]
```

Advances the file name of the active tool by 1.

Transpose Action Line

TransposeGet	**[TransposeGet,** Start xPos, Start yPos, Start zPos, End xPos, End yPos, End zPos, Action Line Length, x of red axis, y of red axis, z of red axis, x of green axis, y of green axis, z of green axis, x of blue axis, y of blue axis, z of blue axis**]**

Gets current Transpose Action Line values (**Sub-Level** only).

Example:

```
[TransposeGet, xPos, yPos, zPos]
```

Sets the variables xPos, yPos and zPos to the 3D position values of the start of the transpose action line.

TransposeIsShown	**[TransposeIsShown]**

Returns status of transpose line. Output: Returns 1 if shown, zero if not (**Sub-Level** only).

TransposeSet	**[TransposeSet,** Start xPos, Start yPos, Start zPos, End xPos, End yPos, End zPos, Action Line Length, x of red axis, y of red axis, z of red axis, x of green axis, y of green axis, z of green axis, x of blue axis, y of blue axis, z of blue axis**]**

Sets current Transpose Action Line values (**Sub-Level** only).

Example:

```
[TransposeSet, xPos, yPos, zPos]
```

Sets the start of the transpose action line 3D position to xPos, yPos, zPos.

Curves

CurveAddPoint	**[CurveAddPoint,** Curve Index (zero based), x position, y position, z position**]**

Add a new point to the specified curve (**Sub-Level** only). Output: Returns the point index (zero based) or -1 if failed.

Example:

```
[CurveAddPoint,1,1,2,3]
```

Appends a new point (x=1,y=2,z=3) to the second curve in the list.

CurvesCreateMesh	**[CurvesCreateMesh,** Name, Action (0(default): Append mesh to the active mesh, 1: Add as a new subtool, 2: Export OBJ file if file does not exist, 3: Export Obj file and overwrite if exists), Thickness (zero: single side mesh)**]**

Creates a mesh from the current curves (**Sub-Level** only). Output: Returns the number of points in the new mesh. zero=error, -1=file exists.

Example:

```
[CurvesCreateMesh, myCurveMesh, 1, 10]
```

Creates a mesh of 10 units thickness from the current curves and appends it as a new subtool named 'myCurveMesh'.

CurvesDelete	**[CurvesDelete,** name**]**

Deletes named curves list (**Sub-Level** only).

Example:

```
[CurvesDelete, myCurves]
```

Deletes curves list named 'myCurves'.

CurvesNewCurve	**[CurvesNewCurve]**

Creates a new curve in the current curves list (**Sub-Level** only). Output: Returns the curve index (zero based) or -1

if failed.

Example:

```
[CurvesNewCurve]
```

Creates a new curve in the current curves list.

CurvesNew	[CurvesNew, name]

Creates a new curves list (**Sub-Level** only).

Example:

```
[CurvesNew, myCurves]
```

Creates a new curve list named 'myCurves'.

CurvesToUI	[CurvesToUI]

Copy the ZScript curves to UI (**Sub-Level** only). Output: Returns zero of OK or -1 if failed.

Example:

```
[CurvesToUI]
```

Copies the current zscript curves to the UI so they are visible to the user.

Special 3D Tool Commands

DispMapCreate	[DispMapCreate, Image Width, Image Height, Smooth (default:yes), SubPoly (default:0), Border (default:8), UVTile index (default:ignores UV tiles)]

Creates DisplacementMap Output: Returns zero if executed successfully. Any other value indicates an error

Example:

```
[DispMapCreate, 1024, 1024, 1, 7, 2]
```

Creates a DispMap, image size 1024x124, smooth=yes, border=7, UVTi1e index=2

Mesh3DGet	**[Mesh3DGet,** Property: (0:PointsCount 1:FacesCount 2:XYZ bounds 3:UVBounds 4:1stUVTile 5:NxtUVTile 6:PolysInUVTile 7:3DAreaOfUVTile 8:Full3DMeshArea), Optional input 1 Vertix/Face/Group/UVTile H index (0 based), Optional input 2, Optional output variable1, Optional output variable2, Optional output variable3, Optional output variable4, Optional output variable5, Optional output variable6, Optional output variable7, Optional output variable8**]**

Gets information about the currently active Mesh3D tool. Output: Returns zero if command executed successfully, any other value indicates an error (**Sub-Level** only).

Example:

```
[Mesh3DGet, 0]
```

returns the number of vertices.

NormalMapCreate	**[NormalMapCreate,** Image Width, Image Height, Smooth (default:yes), SubPoly (default:0), Border (default:8), UVTile index (default:ignores UV tiles), Local(tangent) coordinates? (default:world coordinates)**]**

Creates NormalMap Output: Returns zero if executed successfully. Any other value indicates an error

Example:

```
[NormalMapCreate, 1024, 1024, 1, 7, 2]
```

Creates a Normal Map, image size 1024x124, smooth=yes, border=7, UVTi1e index=2.

ZSphereAdd	**[ZSphereAdd ,** xPos, yPos, zPos, Radius, Parent index (0 based), Optional Color 0x000000<->0xffffff (RED*65536)+(GREEN*256)+BLUE, Optional Mask (0:unmasked to 255:fully masked), Optional TimeStamp, Optional Flags (0:default, 1:invisible link to parent)**]**

Adds new ZSphere to the currently active ZSpheres tool Output: Returns the the index of the new ZSphere or -1 if command failed (**Sub-Level** only).

Example:

```
[ZSphereAdd, 0, .5, 1, .1, 0]
```

Adds a ZSphere located at (0, 0.5, 1) with 0.1 radius and ZSphere #0 as the parent.

ZSphereDel	**[ZSphereDel ,** ZSphere index (Sphere 0 can't be deleted)**]**

Deletes a ZSphere from the currently active ZSpheres tool Output: Returns zero if command executed successfully (**Sub-Level** only).

Example:

```
[ZSphereDel, 2]
```

Deletes the 3rd ZSphere.

ZSphereEdit	**[ZSphereEdit ,** ZSpheres editing commands, Store undo? (0:Skip Undo, 1:Store undo)**]**

Prepares the currently active ZSpheres tool for ZScript editing session. Output: Returns the zero if command executed successfully.

Example:

```
[ZSphereEdit, ...commands...]
```

begins ZSphere edit session and executes ...commands...

ZSphereGet	**[ZSphereGet,** Property: 0:ZSpheres count, 1:xPos, 2:yPos, 3:zPos, 4:radius, 5:color, 6:mask, 7:ParentIndex(-1:none), 8:LastClickedIndex(-1:none), 9:TimeStamp, 10:ChildsCount, 11:ChildIndex (2nd index), 12:TimeStampCount, 13:TimeStampIndex, 14:flags, 15:Twist Angle, 16:Membrane, 17:X Res, 18:Y Res, 19:Z Res, Optional ZSphere index (0 based), Optional 2nd index (0 based)**]**

Gets information about the currently active ZSpheres tool. (Must be placed within ZSphereEdit command) Output: Returns the value of the specified property (**Sub-Level** only).

Example:

```
[ZSphereGet, 0]
```

returns the number of ZSpheres.

```
[ZSphereGet, 2, 1]
```

returns the Y position of the 2nd ZSphere.

ZSphereSet	**[ZSphereSet ,** Property: 0:unused, 1:xPos, 2:yPos, 3:zPos, 4:radius, 5:color, 6:mask, 7:ParentIndex, 8:unused, 9:TimeStamp, 10:unused, 11:unused, 12:unused, 13:unused, 14:flags, 15:Twist Angle, 16:Membrane, 17:X Res, 18:Y Res, 19:Z Res, 20:XYZ Res, 21:UserValue, ZSphere index (0 based), New property value**]**

Modifies a property of the currently active ZSpheres tool. (Must be placed within ZSphereEdit command) Output: Returns zero if command executed successfully (**Sub-Level** only).

Example:

```
[ZSphereSet, 4, 6, .5]
```

sets the radius of ZSphere index 6 to 0.5.

Index

CurvesDelete
CurvesNewCurve
CurvesNew
CurvesToUI
Delay
DispMapCreate
Exit
FileDelete
FileExecute
FileExists
FileGetInfo
FileNameAdvance
FileNameAsk
FileNameExtract
FileNameGetLastTyped
FileNameGetLastUsed
FileNameMake
FileNameResolvePath
FileNameSetNext
FontSetColor
FontSetOpacity
FontSetSize
FontSetSizeLarge
FontSetSizeMedium
FontSetSizeSmall
FrontColorSet
GetActiveToolPath
HotKeyText
IButton
IClick
IClose
IColorSet
IConfig
IDialog
IDisable
IEnable
IExists
IMaximize
IMinimize
IModGet
IModSet
Interpolate
IPress
IReset
IsDisabled
IsEnabled
ISet
ISetHotkey
ISetMax
ISetMin
ISetStatus
IShowActions
IShow
ISlider
IsLocked
IStroke
ISubPalette
IsUnlocked
ISwitch
IToggle
IUnlock
IUnPress
IUpdate
IVPos
IWidth
Logical Operators
Loop
LoopContinue
LoopExit
Math Functions
Math Operators
MemCopy
MemCreate
MemCreateFromFile
MVarDef
MVarGet
MVarSet
NormalMapCreate
Note
NoteBar
NoteIButton
NoteIGet
NoteISwitch
PageSetWidth
PaintBackground
PaintBackSliver
PaintPageBreak
PaintRect
PaintTextRect
PD
PenMoveCenter
PenMoveDown
PenMoveLeft
PenMoveRight
PenMove
PenSetColor
PixolPick
PropertySet
Randomize
RGB
RoutineCall
RoutineDef
SectionBegin
SectionEnd
ShellExecute
Sleep
SleepAgain
SoundPlay
SoundStop
StrAsk
StrExtract
ToolGetCount
ToolGetPath
ToolGetSubToolID
ToolGetSubToolsCount
ToolLocateSubTool
ToolSelect
ToolSetPath
TransformGet
TransformSet
TransposeGet
TransposeIsShown
TransposeSet
Val
VarAdd
VarDec
VarDef
VarDiv
VarInc
VarListCopy
VarLoad
VarMul
VarSave
VarSet
VarSize
VarSub
Var
ZBrushInfo
ZBrushPriorityGet
ZBrushPrioritySet
<zscriptinsert>
ZSphereAdd
ZSphereDel
ZSphereEdit
ZSphereGet
ZSphereSet

If MemDelete StrFind