

# MBS Real Studio ComputerControl Plugin Documentation

Christian Schmitz

May 15, 2012

## 0.1 Introduction

This is the PDF version of the documentation for the Real Studio Plug-in from Monkeybread Software Germany. Plugin part: MBS Real Studio ComputerControl Plugin

## 0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 7
- 3 List of all classes 51
- 4 List of all modules 53

# Chapter 1

## List of Topics

• 2 Remote Control	7
– 2.5 module RemoteControlMBS	24
* 2.5.1 LinuxConvertCase(keysymbol as integer, byref lowerSymbol as integer, byref upperSymbol as integer) as boolean	25
* 2.5.1 LinuxKeyCodeToKeySymbol(keycode as integer, index as integer) as integer	25
* 2.5.1 LinuxKeyNameToKeySymbol(keyname as string) as integer	25
* 2.5.1 LinuxKeySymbolToKeyCode(keysymbol as integer) as integer	25
* 2.5.1 LinuxKeySymbolToKeyName(keysymbol as integer) as string	26
* 2.5.1 LinuxMouseClicked(ButtonID as integer, ButtonDown as boolean) as boolean	26
* 2.5.1 LinuxMousePositionX as integer	26
* 2.5.1 LinuxMousePositionY as integer	26
* 2.5.1 LinuxMoveMouse(x as integer, y as integer) as boolean	26
* 2.5.1 LinuxPressControlKey(keydown as boolean) as boolean	27
* 2.5.1 LinuxPressKey(Keycode as integer) as boolean	27
* 2.5.1 LinuxPressKey(Keycode as integer, ButtonDown as boolean) as boolean	27
* 2.5.1 LinuxPressOptionKey(keydown as boolean) as boolean	28
* 2.5.1 LinuxPressShiftKey(keydown as boolean) as boolean	28
* 2.5.1 LinuxSupported as boolean	28
* 2.5.1 MacCurrentProcessID as integer	28
* 2.5.1 MacCurrentProcessName as string	29
* 2.5.1 MacDescriptionForKeyCode(keycode as integer) as string	29
* 2.5.1 MacDisplayNameForKeyCode(keycode as integer) as string	29
* 2.5.1 MacForegroundProcessID as integer	29
* 2.5.1 MacFrontProcessName as string	29
* 2.5.1 MacKeyboardLocalizedName as string	30

* 2.5.1 MacKeyboardName as string	30
* 2.5.1 MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean) as boolean	30
* 2.5.1 MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean	31
* 2.5.1 MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean	32
* 2.5.1 MacMousePositionX as integer	32
* 2.5.1 MacMousePositionY as integer	32
* 2.5.1 MacMouseWheel(wheel1 as integer) as boolean	33
* 2.5.1 MacMouseWheel(wheel1 as integer, wheel2 as integer) as boolean	33
* 2.5.1 MacMoveMouse(x as double, y as double) as boolean	34
* 2.5.1 MacPressCommandKey(keydown as boolean) as boolean	34
* 2.5.1 MacPressControlKey(keydown as boolean) as boolean	34
* 2.5.1 MacPressKey(keychar as integer, virtualkey as integer) as boolean	35
* 2.5.1 MacPressKey(keychar as integer, virtualkey as integer, keydown as boolean) as boolean	35
* 2.5.1 MacPressOptionKey(keydown as boolean) as boolean	35
* 2.5.1 MacPressShiftKey(keydown as boolean) as boolean	36
* 2.5.1 MacProcessCount as integer	36
* 2.5.1 MacProcessName(index as integer) as string	36
* 2.5.1 MacProcessVisible(index as integer) as boolean	37
* 2.5.1 MacTextForKeyCode(keycode as integer, KeyAction as integer, ModifierState as integer) as string	37
* 2.5.1 MouseClick(x as integer, y as integer, down as boolean) as boolean	39
* 2.5.1 MoveMouse(x as integer, y as integer) as boolean	39
* 2.5.1 PressControlKey(keydown as boolean) as boolean	39
* 2.5.1 PressOptionKey(keydown as boolean) as boolean	39
* 2.5.1 PressShiftKey(keydown as boolean) as boolean	39
* 2.5.1 WinCurrentProcessID as integer	40
* 2.5.1 WinFindWindow(ClassName as string, WindowName as string) as integer	40
* 2.5.1 WinForegroundProcessID as integer	40
* 2.5.1 WinFrontWindowTitle as string	40
* 2.5.1 WinKeyboardName as string	41
* 2.5.1 WinKeyIsDown(virtualkey as integer) as boolean	41
* 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean) as boolean	41
* 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean	42
* 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean	42
* 2.5.1 WinMoveMouse(x as integer, y as integer) as boolean	43

* 2.5.1 WinPressControlKey(keydown as boolean) as boolean	43
* 2.5.1 WinPressKey(ScanCode as integer) as boolean	43
* 2.5.1 WinPressKey(ScanCode as integer, keydown as boolean) as boolean	44
* 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer) as boolean	44
* 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer, keydown as boolean) as boolean	44
* 2.5.1 WinPressOptionKey(keydown as boolean) as boolean	45
* 2.5.1 WinPressShiftKey(keydown as boolean) as boolean	45
* 2.5.1 WinScanCodeToVirtualKeyCode(ScanCode as integer) as integer	45
* 2.5.1 WinSendMessage(Win as window, Msg as integer, lParam as integer, wParam as integer) as integer	46
* 2.5.1 WinSendMessage(WindowHandle as integer, Msg as integer, lParam as integer, wParam as integer) as integer	46
* 2.5.1 WinVirtualKeyCodeToCharCode(VirtualKeyCode as integer) as integer	47
* 2.5.1 WinVirtualKeyCodeToScanCode(VirtualKeyCode as integer) as integer	47
* 2.5.1 WinVirtualKeyForASCII(Character as integer, byref VirtualKeyCode as integer, Byref ShiftKey as boolean, byref ControlKey as Boolean, byref AltKey as boolean) as boolean	47
– 2.3 class GammaMBS	13
* 2.3.1 Blue(Index as integer) as double	14
* 2.3.1 Constructor	14
* 2.3.1 Constructor(mainwindow as window)	14
* 2.3.1 Green(Index as integer) as double	15
* 2.3.1 Red(Index as integer) as double	15
* 2.3.1 SetGamma(gammaScale as double=1.0) as boolean	15
* 2.3.2 Available as Boolean	16
– 2.1 class KeyCodesMBS	7
* 2.1.1 AsciiToKeyCode(ascii as integer) as integer	7
* 2.1.1 KeyCodeToAscii(keycode as integer) as integer	8
* 2.1.1 KeyCodeToAsciiWithSecondKeyCode(keycode as integer) as integer	9
* 2.1.1 TestForAsciiKeyDown(ascii as integer) as boolean	10
* 2.1.1 TestForKeyDown(keycode as integer) as boolean	10
* 2.1.1 Update	10
* 2.1.2 LastError as Integer	10
* 2.1.2 Name as String	10
* 2.1.2 State as Integer	11
– 2.2 class GammaFadeMBS	11
* 2.2.1 Constructor	11
* 2.2.1 Constructor(mainwindow as window)	12
* 2.2.1 GammaFadeIn(seconds as double) as boolean	12
* 2.2.1 GammaFadeOut(seconds as double) as boolean	12
* 2.2.2 Available as Boolean	13

* 2.2.2 FadeColor as Color	13
– 2.4 class PresskeyMBS	16
* 2.4.1 clear	16
* 2.4.1 mouseclick(down as boolean)	17
* 2.4.1 MouseClick(down as boolean, rightdown as boolean)	17
* 2.4.1 MouseMove(globalx as integer,globaly as integer)	17
* 2.4.1 MouseMoveClick(globalx as integer,globaly as integer,down as boolean)	18
* 2.4.1 MouseMoveClick(globalx as integer,globaly as integer,down as boolean, rightdown as boolean)	18
* 2.4.1 press	18
* 2.4.1 pressraw(down as boolean)	20
* 2.4.2 Charcode as integer	21
* 2.4.2 Command as boolean	21
* 2.4.2 Control as boolean	22
* 2.4.2 Keycode as integer	22
* 2.4.2 Lasterror as integer	23
* 2.4.2 Option as boolean	23
* 2.4.2 Shift as boolean	23
* 2.4.2 VirtualCode as integer	24

## Chapter 2

# Remote Control

### 2.1 class KeyCodesMBS

class KeyCodesMBS

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class for translating between key codes and ASCII characters.

**Notes:**

This class is outdated, please use RemoteControlMBS module.

The list of key codes on a Mac is the same as the one used in the RB documentation for the sprite surface. Some special keys:

Key codes for Windows:

#### 2.1.1 Methods

**AsciiToKeyCode(ascii as integer) as integer**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns the keycode which is needed to be pressed to create this ASCII character.

Modifier keys	
kVirtualCapsLockKey	& h039
kVirtualShiftKey	& h038
kVirtualControlKey	& h03B
kVirtualOptionKey	& h03A
kVirtualCommandKey	& h037
Editing/utility keys	
kVirtualHelpKey	& h072
kVirtualDeleteKey	& h033
kVirtualTabKey	& h030
kVirtualEnterKey	& h04C
kVirtualReturnKey	& h024
kVirtualEscapeKey	& h035
kVirtualForwardDeleteKey	& h075
Navigation keys	
kVirtualHomeKey	& h073
kVirtualEndKey	& h002
kVirtualPageUpKey	& h074
kVirtualPageDownKey	& h079
kVirtualLeftArrowKey	& h07B
kVirtualRightArrowKey	& h07C
kVirtualUpArrowKey	& h07E
kVirtualDownArrowKey	& h07D

**Notes:** You get the main key, not any modifier which may also be pressed.

### KeyCodeToAscii(keycode as integer) as integer

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns the ASCII code which matches the key code.

#### Example:

```
dim k as KeyCodesMBS

k=new KeyCodesMBS

// makes a big A on a german keyboard:
MsgBox chr(k.KeyCodeToAsciiWithSecondKeyCode(0+512))
```

**Notes:**

Note on a Mac:

Bit 0 to 6 are the keycode, bit 7 is 1 if key goes up and 0 if key goes down.

Bit 8 to 15 are modifier keys:

Constants for the modifier keys:

**KeyCodeToAsciiWithSecondKeyCode(keycode as integer) as integer**

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns the ASCII code which matches the key code with the current state.

**Example:**

```
dim k as KeyCodesMBS
```

```
k=new KeyCodesMBS
```

```
MsgBox str(k.KeyCodeToAscii(& h0A))
```

```
MsgBox chr(k.KeyCodeToAsciiWithSecondKeyCode(0))
```

```
// gives on a German Keyboard
```

**Notes:**

On the first call of KeyCodeToAscii you get the ASCII char, but if you call KeyCodeToAsciiWithSecondKeyCode and the state property is not changed, you can get what the user gets if he presses several keys.

KeyCodeToAscii resets state to 0 before calling.

On Windows, KeyCodeToAscii and KeyCodeToAsciiWithSecondKeyCode are equal.

Note on a Mac:

Bit 0 to 6 are the keycode, bit 7 is 1 if key goes up and 0 if key goes down.

Bit 8 to 15 are modifier keys:

Constants for the modifier keys:

**TestForAsciiKeyDown(ascii as integer) as boolean**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns true if the key with the given ascii code is pressed.

**TestForKeyDown(keycode as integer) as boolean**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Returns true if the key with the given key code is pressed.

**Update**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Updates the layout so it is the current one.

## 2.1.2 Properties

**LastError as Integer**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The last error code.  
**Notes:**

The last function was successful if lasterror is 0.  
If the last function was not available on this machine, the value is set to -1.  
Other values are Mac OS error codes.  
(Read and Write property)

**Name as String**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The name for the current keyboard layout.  
**Notes:**

On Mac OS 9 e.g. "Deutsch".  
On Mac OS X e.g. "German".  
On Windows 98 e.g. "00020407".  
(Read only property)

### State as Integer

Plugin Version: 3.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The last state.  
**Notes:**

The function `KeyCodeToAscii` stores a code here which the function `KeyCodeToAsciiWithSecondKeyCode` will use on further calls.  
(Read and Write property)

## 2.2 class GammaFadeMBS

### class GammaFadeMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class to perform gamma fades on Mac OS X and Windows.  
**Notes:** On Mac OS X a system API is used. On Windows the GammaMBS class is used internally.

### 2.2.1 Methods

#### Constructor

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Initializes the gamma fade class and queries the current gamma values.  
**Notes:**

On Windows a window is needed to get the exclusive access to the gamma controls.

The `Available` property is set to true if the constructor was successful.  
See also:

- 2.2.1 Constructor(mainwindow as window) 12

### Constructor(mainwindow as window)

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Initializes the gamma fade class and queries the current gamma values.

**Notes:**

On Windows a window is needed to get the exclusive access to the gamma controls.

The Available property is set to true if the constructor was successful.  
See also:

- 2.2.1 Constructor 11

### GammaFadeIn(seconds as double) as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Fades back to the normal screen.

**Notes:**

Seconds is the duration of the fade in seconds. Typical 0.3 seconds.  
Returns true on success and false on failure.

### GammaFadeOut(seconds as double) as boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Fades to a solid color.

**Notes:**

Seconds is the duration of the fade in seconds. Typical 0.5 seconds.  
Returns true on success and false on failure.

## 2.2.2 Properties

### Available as Boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Whether the constructor was successful.

#### Notes:

True if the initialization code worked.

On Mac OS X the constructor fails to get access if there is another gamma fade running.

On Windows the access to the gamma controls can fail if you don't provide a window, the hardware can't do gamma fades (e.g. running in VMWare) or DirectX is not available.

(Read and Write property)

### FadeColor as Color

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The color to use for the fade.

#### Example:

```
dim g as new GammaFadeMBS
g.FadeColor=&c0000FF // blue
```

#### Notes:

Default is black.

(Read and Write property)

## 2.3 class GammaMBS

### class GammaMBS

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** A class to change the gamma settings of the display.

**Notes:**

You can use that class to change the gamma table. For example to invert the screen or increase/decrease one of the colors.

Works on Mac OS X and Windows.

**2.3.1 Methods****Blue(Index as integer) as double**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The blue value with the given index.

**Notes:**

Index is from 0 to 255 and value from 0.0 to 1.0.  
(Read and Write computed property)

**Constructor**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Initializes the gamma class and queries the current gamma values.

**Notes:**

On Windows a window is needed to get the exclusive access to the gamma controls.

The Available property is set to true if the constructor was successful.  
See also:

- 2.3.1 Constructor(mainwindow as window)

14

**Constructor(mainwindow as window)**

Plugin Version: 8.4 Console & Web: No Mac: Yes, Win: Yes, Linux: No, . **Function:** Initializes the gamma class and queries the current gamma values.

**Notes:**

On Windows a window is needed to get the exclusive access to the gamma controls.

The Available property is set to true if the constructor was successful.

See also:

- 2.3.1 Constructor

14

**Green(Index as integer) as double**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The green value with the given index.

**Notes:**

Index is from 0 to 255 and value from 0.0 to 1.0.

(Read and Write computed property)

**Red(Index as integer) as double**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The red value with the given index.

**Notes:**

Index is from 0 to 255 and value from 0.0 to 1.0.

(Read and Write computed property)

**SetGamma(gammaScale as double=1.0) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Sets the gamma values.

**Notes:**

The current table assigned to the main screen.

All values are multiplied with gammaScale

## 2.3.2 Properties

### Available as Boolean

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Whether the constructor was successful.

#### Notes:

True if the initialization code worked.

On Mac OS X the constructor fails to get access if it does not find a display with gamma controls.

On Windows the access to the gamma controls can fail if you don't provide a window, the hardware can't do gamma fades (e.g. running in VMWare) or DirectX is not available.  
(Read and Write property)

## 2.4 class PresskeyMBS

### class PresskeyMBS

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** This class allows you to simulate keyboard activity.

**Notes:** This class is deprecated, please use RemoteControlMBS module.

### 2.4.1 Methods

#### clear

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Resets this object.

**mouseclick(down as boolean)**

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Clicks the mouse button.

**Notes:**

Clicks the mouse at current position.

Left button is used.

If you forget the mouseup you may loose control about a Mac with Mac OS X, but if you have enabled external terminal login you can login via SSH and quit Realbasic from outside.

The Mac version sets the lasterror property.

See also:

- 2.4.1 MouseClick(down as boolean, rightdown as boolean)

17

**MouseClick(down as boolean, rightdown as boolean)**

Plugin Version: 8.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Clicks the mouse with the given buttons.

**Notes:** Same as MouseClick, but with an additional rightdown parameter.

See also:

- 2.4.1 mouseclick(down as boolean)

17

**MouseMove(globalx as integer,globaly as integer)**

Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Moves the mouse.

**Example:**

```
dim x,y as integer
dim p as PresskeyMBS
p=new PresskeyMBS
x=system.mousex
y=system.mousey+10 // move ten pixels down
p.mousemove x,y
```

**Notes:**

Moves the mouse on screen.  
 The destination must be specified in global screen coordinates.  
 Requires Mac OS 8.5 or newer for Mac OS Classic.

**MouseMoveClick(globalx as integer,globaly as integer,down as boolean)**

Plugin Version: 2.8 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Moves the mouse.  
**Notes:**

Moves the mouse on screen and clicks.  
 The destination must be specified in global screen coordinates.  
 Requires Mac OS 8.5 or newer for Mac OS Classic.

The Mac version sets the lasterror property.

Mac OS X only:

Based on the values entered, the appropriate mouse-down, mouse-up, mouse-move, or mouse-drag events are generated, by comparing the new state with the current state.  
 See also:

- 2.4.1 MouseMoveClick(globalx as integer,globaly as integer,down as boolean, righdown as boolean) 18

**MouseMoveClick(globalx as integer,globaly as integer,down as boolean, righdown as boolean)**

Plugin Version: 8.1 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Moves the mouse cursor and clicks the mouse with the given buttons.

**Notes:** Same as MouseMoveClick, but with an additional righdown parameter.

See also:

- 2.4.1 MouseMoveClick(globalx as integer,globaly as integer,down as boolean) 18

**press**

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Presses a key.

**Example:**

' example for Mac OS X:

```
dim p as PresskeyMBS
```

```
p=new PresskeyMBS
p.shift=true
p.charcode=asc("H")
p.keycode=4
p.press 'press H
p.shift=false
p.charcode=asc("e")
p.keycode=14
p.press 'press e
p.charcode=asc("l")
p.keycode=37
p.press 'press l
p.press 'press l
p.charcode=asc("o")
p.keycode=31
p.press 'press o
```

' example for Windows:

```
editfield1.setfocus 'so we get something in it!
```

```
p=new PresskeyMBS
p.shift=true
p.charcode=asc("H")
p.virtualCode=-1 'automatically try to get this code
p.press 'press H
p.shift=false
p.virtualCode=-1 'after a call you find the virtualCode there
p.charcode=asc("e")
p.press 'press e
p.virtualCode=-1
p.charcode=asc("l")
p.press 'press l
p.press 'press l
p.virtualCode=-1
p.charcode=asc("o")
p.press 'press o
```

### Notes:

Simulates a keypress and handles the Settings for Shift, Control, Command and Option key. Command shows no reaction in my tests for Mac OS Classic

On Windows, if you set `virtualcode=-1` this method will set it to the code matching the `charcode` property.

The Mac version sets the `lasterror` property.

### `pressraw(down as boolean)`

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Presses a key.

#### **Example:**

' Works perfectly on Mac OS X but Shift can't be done Raw on Mac OS 9:

```
dim p as PresskeyMBS
```

```
p=new PresskeyMBS
```

```
p.charcode=0
```

```
p.keycode=& h38
```

```
p.pressraw true 'shift down
```

```
p.charcode=asc("H")
```

```
p.keycode=4
```

```
p.pressraw true 'press H
```

```
p.pressraw false
```

```
p.charcode=0
```

```
p.keycode=& h38
```

```
p.pressraw false 'shift up
```

```
p.charcode=asc("e")
```

```
p.keycode=14
```

```
p.pressraw true 'press e
```

```
p.pressraw false
```

```
p.charcode=asc("l")
```

```
p.keycode=37
```

```
p.pressraw true 'press l
```

```
p.pressraw false
```

```
p.pressraw true 'press l
```

```
p.pressraw false
```

```
p.charcode=asc("o")
```

```
p.keycode=31
```

```
p.pressraw true 'press o
```

```
p.pressraw false
```

**Notes:**

Simulates a keypress and doesn't handle the Settings for Shift, Control, Command and Option key.  
On Windows, if you set virtualcode=-1 this method will set it to the code matching the charcode property.

The Mac version sets the lasterror property.

## 2.4.2 Properties

### Charcode as integer

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The character you want to press.

**Example:**

```
dim p as PresskeyMBS

p=new PresskeyMBS
p.charcode=8
p.keycode=& h33
p.press // press backspace on Mac OS
```

**Notes:**

On Mac OS Classic the final printed charcode inside an editfield is the one you pass.  
(Read and Write property)

### Command as boolean

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Tells the Press method to press down command key before and release it after it simulates the key press.

**Example:**

```
dim p as new PresskeyMBS
p.command=True
```

**Notes:**

It looks like the command key is not working for Mac OS Classic.

On Windows the control key is pressed if control or command properties are true.  
(Read and Write property)

**Control as boolean**

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Tells the Press method to press down control key before and release it after it simulates the key press.

**Example:**

```
dim p as new PresskeyMBS
p.Control=True
```

**Notes:**

On Windows the control key is pressed if control or command properties are true.  
(Read and Write property)

**Keycode as integer**

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** For Mac OS the number of the key on the keyboard which is pressed.

**Example:**

```
dim p as PresskeyMBS
p=new PresskeyMBS
p.keycode=2
p.charcode=asc("d")
p.press 'shows a "d" on Mac OS
```

**Notes:**

You find this codes in that nice tables for the codes of the Spritesurface in the Realbasic books.  
(Read and Write property)

**Lasterror as integer**

Plugin Version: 3.3 Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The last error code.  
**Notes:**

Not all functions of this class fill this property!  
(Read and Write property)

**Option as boolean**

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Tells the Press method to press down option key before and release it after it simulates the key press.

**Example:**

```
dim p as PresskeyMBS
p.option=True
```

**Notes:** (Read and Write property)

**Shift as boolean**

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** Tells the Press method to press down shift before and release it after it simulates the key press.

**Example:**

```
dim p as PresskeyMBS

p=new PresskeyMBS
p.shift=true
p.charcode=asc("1")
p.keycode=18
p.press 'press !
```

**Notes:** (Read and Write property)

### VirtualCode as integer

Console & Web: Yes Mac: Yes, Win: Yes, Linux: No, . **Function:** The virtual keycode to press.

**Example:**

```
dim p as PresskeyMBS
p=new PresskeyMBS
p.charcode=asc("m")
p.virtualcode=-1
p.press ' makes a "m" on Windows.
```

**Notes:**

If you don't know this code, just pass -1 and this value will be filled automatically using the charcode property.

After you call press or pressraw you'll find the matching virtualcode in this property.

(Read and Write property)

## 2.5 module RemoteControlMBS

**module RemoteControlMBS**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** This module collects methods for simulating user input with mouse and keyboard.

**Notes:**

You can use this class for GUI scripting.

Requires libXtst.so.6 on Linux to be installed.

### 2.5.1 Methods

**LinuxConvertCase(keySymbol as integer, byref lowerSymbol as integer, byref upperSymbol as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Converts the case of the given keySymbol.

**Notes:**

In lowerSymbol you will receive the key symbol for a lower case key and in upperSymbol the uppercase keys.

So passing in the key symbol of "a" or "A" will both return "a" and "A".

**LinuxKeyCodeToKeySymbol(keycode as integer, index as integer) as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Returns from the keyboard mapping tables the key symbol for the given key code.

**Notes:** Index is the offset in the tables. 0 for the first key symbol and 1 for the second. If the user presses shift the keyboard driver uses index=1 to access the capital letters.

**LinuxKeyNameToKeySymbol(keyname as string) as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Searches the key mapping list for the key symbol for a given name.

**Notes:** Returns 0 on failure.

**LinuxKeySymbolToKeyCode(keySymbol as integer) as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Searches the hardware dependent key code for the given key symbol.

**Notes:** Returns 0 on any error.

**LinuxKeySymbolToKeyName(keysymbol as integer) as string**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Returns the name of the key symbol.

**Notes:** Can return "" on an invalide keysymbol value.

**LinuxMouseClicked(ButtonID as integer, ButtonDown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Simulates a mouse click with the given button.

**Notes:**

ButtonID=1 is the left mouse button.

Returns true on success.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

**LinuxMousePositionX as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Queries the current mouse position.

**LinuxMousePositionY as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Queries the current mouse position.

**LinuxMoveMouse(x as integer, y as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Moves the mouse to the given position.

**Notes:** Returns true on success.

**LinuxPressControlKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Presses the ctrl key.  
**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**LinuxPressKey(Keycode as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Presses the key with the given keycode.  
**Notes:**

Performs a key down and a key up event.  
Returns true on success.  
See also:

- 2.5.1 LinuxPressKey(Keycode as integer, ButtonDown as boolean) as boolean 27

**LinuxPressKey(Keycode as integer, ButtonDown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Performs a key event.  
**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.  
See also:

- 2.5.1 LinuxPressKey(Keycode as integer) as boolean 27

**LinuxPressOptionKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Presses the alt key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**LinuxPressShiftKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Presses the shift key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**LinuxSupported as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: No, Linux: Yes, . **Function:** Whether the functions are available for simulation of keyboard and mouse events.

**Notes:** Returns true if the Xtest extension is installed.

**MacCurrentProcessID as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the process ID of your application.

**Notes:** Returns 0 on any error.

**MacCurrentProcessName as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the name of the current process.

**Example:**

```
msgbox RemoteControlMBS.MacCurrentProcessName
```

**Notes:** This is the name of your application.

**MacDescriptionForKeyCode(keycode as integer) as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Translates a keycode to a string for displaying.

**Notes:** Keycodes which match a special key like "Return", "F1" or "Backspace" are returned as descriptive strings.

**MacDisplayNameForKeyCode(keycode as integer) as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Translates a keycode to a string for displaying.

**Notes:** Same as `MacTextForKeyCode(keycode, 4)`

**MacForegroundProcessID as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Returns the process ID of the application in the foreground.

**Notes:** Returns 0 on any error.

**MacFrontProcessName as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The name of the front process.

**Example:**

```
RemoteControlMBS.MacFrontProcessName="iChat" // move iChat to front
```

**Notes:** (Read and Write computed property)

**MacKeyboardLocalizedName as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The localized name of the current keyboard layout.

**Notes:** Returns "" on any error.

**MacKeyboardName as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The name of the current keyboard layout.

**Notes:** Returns "" on any error.

**MacMouseClicked(x as double, y as double, updateMouseCursorPosition as boolean, MouseButton1 as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Clicks the mouse buttons on the given position.

**Example:**

```
// doing a context menu click with control key:
```

```
dim x,y as Integer
x=System.MouseX
y=System.MouseY
```

```
call RemoteControlMBS.PressControlKey true
call RemoteControlMBS.MacMouseClicked x, y, true, true
call RemoteControlMBS.MacMouseClicked x, y, true, false
call RemoteControlMBS.PressControlKey false
```

**Notes:**

Returns true on success.

If updateMouseCursorPosition is true the mouse position is updated on screen.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

See also:

- 2.5.1 MacMouseClicked(x as double, y as double, updateMouseCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean 31
- 2.5.1 MacMouseClicked(x as double, y as double, updateMouseCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean 32

**MacMouseClicked(x as double, y as double, updateMouseCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Clicks the mouse buttons on the given position.

**Example:**

```
// do a right click:
```

```
dim x,y as Integer
x=System.MouseX
y=System.MouseY
```

```
call RemoteControlMBS.MacMouseClicked x, y, true, false, true
call RemoteControlMBS.MacMouseClicked x, y, true, false, false
```

**Notes:**

Returns true on success.

If updateMouseCursorPosition is true the mouse position is updated on screen.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

See also:

- 2.5.1 `MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean)` as boolean 30
- 2.5.1 `MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean)` as boolean 32

**MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Clicks the mouse buttons on the given position.

**Notes:**

Returns true on success.

If `updateCursorPosition` is true the mouse position is updated on screen.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

See also:

- 2.5.1 `MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean)` as boolean 30
- 2.5.1 `MacMouseClicked(x as double, y as double, updateCursorPosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean)` as boolean 31

**MacMousePositionX as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Queries the current mouse position.

**MacMousePositionY as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Queries the current mouse position.

**MacMouseWheel(wheel1 as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Performs a wheel event.

**Example:**

[call](#) RemoteControlMBS.MacMouseWheel 1

**Notes:**

Returns true on success.

A typical value for the wheel changes are in range from -10 to 10.

See also:

- 2.5.1 MacMouseWheel(wheel1 as integer, wheel2 as integer) as boolean

33

**MacMouseWheel(wheel1 as integer, wheel2 as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Performs a wheel event.

**Example:**

[call](#) RemoteControlMBS.MacMouseWheel 10,5

**Notes:**

Returns true on success.

A typical value for the wheel changes are in range from -10 to 10.

See also:

- 2.5.1 MacMouseWheel(wheel1 as integer) as boolean

33

**MacMoveMouse(x as double, y as double) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Moves the mouse to the given position.

**Example:**

```
call RemoteControlMBS.MacMoveMouse(100,100)
```

**Notes:** Returns true on success.

**MacPressCommandKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Presses the command key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**MacPressControlKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Presses the ctrl key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**MacPressKey(keychar as integer, virtualkey as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Presses the given key.

**Notes:**

keychar is the ASCII code of the key you press and virtualkey is the key code.

Returns true on success.

Performs a key down and a key up event.

See also:

- 2.5.1 MacPressKey(keychar as integer, virtualkey as integer, keydown as boolean) as boolean 35

**MacPressKey(keychar as integer, virtualkey as integer, keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Presses the given key.

**Notes:**

keychar is the ASCII code of the key you press and virtualkey is the key code.

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

See also:

- 2.5.1 MacPressKey(keychar as integer, virtualkey as integer) as boolean 35

**MacPressOptionKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Presses the option key. (labeled with alt)

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**MacPressShiftKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Presses the shift key.  
**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**MacProcessCount as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Counts the high level processes.

**Example:**

```
msgBox str(remoteControlMBS.macprocessCount)
```

**MacProcessName(index as integer) as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** The name of the process with the given index.

**Example:**

```
dim i,c as integer

// search iChat and make it visible
c=remoteControlMBS.macprocesscount
for i=0 to c-1
if remoteControlMBS.macprocessname(i)="iChat" then
remoteControlMBS.macprocessVisible(i)=true
end if
next
```

**Notes:** Index is from 0 to MacProcessCount-1.

**MacProcessVisible(index as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Whether a process is visible or not.

**Example:**

```
dim i,c as integer

// search Safari and make it invisible
c=remoteControlMBS.macprocesscount
for i=0 to c-1
if remoteControlMBS.macprocessname(i)="Safari" then
remoteControlMBS.macprocessVisible(i)=false
end if
next
```

**Notes:**

Index is from 0 to MacProcessCount-1.  
(Read and Write computed property)

**MacTextForKeyCode(keycode as integer, KeyAction as integer, ModifierState as integer) as string**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: No, Linux: No, . **Function:** Translates a keycode to a string.

**Example:**

```
// try some keycodes like e, E, or
dim key as string = "@"

const kCommandKey = 1
const kShift = 2
const kCapsLock = 4
const kOption = 8
const kControl = 16

// with no modifiers
for i as integer = 0 to 127

dim s as string = RemoteControlMBS.MacTextForKeyCode(i, 3, 0)
```

```
if StrComp(key, s, 0) = 0 then
MsgBox "Found with keycode "+str(i)
Return
end if

next

// with shift
for i as integer = 0 to 127

dim s as string = RemoteControlMBS.MacTextForKeyCode(i, 3, kShift)

if StrComp(key, s, 0) = 0 then
MsgBox "Found with keycode "+str(i)+" with Shift"
Return
end if

next

// with option
for i as integer = 0 to 127

dim s as string = RemoteControlMBS.MacTextForKeyCode(i, 3, kOption)

if StrComp(key, s, 0) = 0 then
MsgBox "Found with keycode "+str(i)+" with Option"
Return
end if

next
```

**Notes:**

Works only with Unicode keyboard layouts which need Mac OS X 10.5.

Modifier values:

Possible constants for the Key Action:

**MouseClicked(x as integer, y as integer, down as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Performs a mouse click.

**Notes:**

The mouse cursor is moved to the given position and the first mouse button is pressed.  
Returns true on success.

**MoveMouse(x as integer, y as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Moves the mouse to the given location.

**Notes:** Returns true on success.

**PressControlKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Presses the ctrl key.

**Notes:** Returns true on success.

**PressOptionKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Presses the option key.

**Notes:** Returns true on success.

**PressShiftKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: Yes, Win: Yes, Linux: Yes, . **Function:** Presses the shift key.

**Notes:** Returns true on success.

**WinCurrentProcessID as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns the process ID of your application.

**Notes:** Returns 0 on any error.

**WinFindWindow(ClassName as string, WindowName as string) as integer**

Plugin Version: 10.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Finds a window based on the classname and/or the window name.

**Example:**

```
dim h as integer = RemoteControlMBS.WinFindWindow("SciCalc","")
msgbox hex(h)
```

**Notes:**

One of the names can be an empty string so only the other name is used. Normally you should prefer the class name as it is not localized. The `WindowsListMBS` class can help you finding the class name of a window.

Returns 0 on any error.

**WinForegroundProcessID as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns the process ID of the application which owns the foreground window.

**Notes:** Returns 0 on any error.

**WinFrontWindowTitle as string**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** The title of the window which has the focus.

**Example:**

title=remoteControlMBS.winfrontWindowTitle

**Notes:** May return "" on any error.

### **WinKeyboardName as string**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns the name of the current Windows keyboard layout.

### **WinKeyIsDown(virtualkey as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Tests whether a given key is currently pressed.

### **WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a mouse click on the given position.

**Notes:**

If AbsolutePosition is false the given coordinates are relative to the current position.  
Returns true on success.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

See also:

- 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean 42
- 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean 42

**WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a mouse click on the given position.

**Notes:**

If AbsolutePosition is false the given coordinates are relative to the current position.  
Returns true on success.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

See also:

- 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean) as boolean 41
- 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean 42

**WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean, MouseButton3 as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a mouse click on the given position.

**Notes:**

If AbsolutePosition is false the given coordinates are relative to the current position.  
Returns true on success.

Be aware that the user may get into trouble if you forget to release a mouse button you pressed before. So always call this method another time to release the mouse buttons.

See also:

- 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean) as boolean 41
- 2.5.1 WinMouseClicked(x as integer, y as integer, AbsolutePosition as boolean, MouseButton1 as boolean, MouseButton2 as boolean) as boolean 42

**WinMoveMouse(x as integer, y as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Moves the mouse cursor to the given position.

**Notes:**

Returns true on success.

Internally the coordinates are converted to normalized absolute coordinates which can lead to have the mouse one pixel off the requested position due to rounding.

**WinPressControlKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Presses the ctrl key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

**WinPressKey(ScanCode as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a key event with the given data.

**Notes:**

This methods creates a keydown and a keyup event.

Returns true on success.

See also:

- 2.5.1 WinPressKey(ScanCode as integer, keydown as boolean) as boolean 44
- 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer) as boolean 44
- 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer, keydown as boolean) as boolean 44

**WinPressKey(ScanCode as integer, keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a key event with the given data.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

See also:

- 2.5.1 WinPressKey(ScanCode as integer) as boolean 43
- 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer) as boolean 44
- 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer, keydown as boolean) as boolean 44

**WinPressKey(virtualkey as integer, ScanCode as integer) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a key event with the given data.

**Notes:**

This methods creates a keydown and a keyup event.

Returns true on success.

See also:

- 2.5.1 WinPressKey(ScanCode as integer) as boolean 43
- 2.5.1 WinPressKey(ScanCode as integer, keydown as boolean) as boolean 44
- 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer, keydown as boolean) as boolean 44

**WinPressKey(virtualkey as integer, ScanCode as integer, keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Performs a key event with the given data.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

See also:

- 2.5.1 WinPressKey(ScanCode as integer) as boolean 43
- 2.5.1 WinPressKey(ScanCode as integer, keydown as boolean) as boolean 44
- 2.5.1 WinPressKey(virtualkey as integer, ScanCode as integer) as boolean 44

### **WinPressOptionKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Presses the alt key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

### **WinPressShiftKey(keydown as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Presses the shift key.

**Notes:**

Returns true on success.

Be aware that the user may get into trouble if you forget to release a key you pressed before. So always call this method another time to release the key.

### **WinScanCodeToVirtualKeyCode(ScanCode as integer) as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Searches the virtual key code for the given scancode.

**Notes:**

Each key has a hardware dependend scan code. For each scancode you can get the virtual key code which is hardware independend.

Returns 0 on failure.

**WinSendMessage**(Win as window, Msg as integer, lParam as integer, wParam as integer) as integer

Plugin Version: 10.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Sends a Windows message to a given window.

**Example:**

```
const WM_CLOSE = & h0010
```

```
call RemoteControlMBS.WinSendMessage(window1, WM_CLOSE, 0, 0)
```

**Notes:** The result depends on the message command.

See also:

- 2.5.1 WinSendMessage(WindowHandle as integer, Msg as integer, lParam as integer, wParam as integer) as integer 46

**WinSendMessage**(WindowHandle as integer, Msg as integer, lParam as integer, wParam as integer) as integer

Plugin Version: 10.1 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Sends a Windows message to a given window.

**Example:**

```
const WM_CLOSE = & h0010
```

```
dim h as integer = RemoteControlMBS.WinFindWindow("SciCalc", "")
```

```
call RemoteControlMBS.WinSendMessage(h, WM_CLOSE, 0, 0) // Closes Calc
```

**Notes:** The result depends on the message command.

See also:

- 2.5.1 WinSendMessage(Win as window, Msg as integer, lParam as integer, WParam as integer) as integer 46

**WinVirtualKeyCodeToCharCode(VirtualKeyCode as integer) as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Queries the char code for a virtual key code.

**Notes:** Returns 0 on failure.

**WinVirtualKeyCodeToScanCode(VirtualKeyCode as integer) as integer**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Finds the scan code to match the given virtual key code.

**Notes:** Returns 0 on failure.

**WinVirtualKeyForASCII(Character as integer, byref VirtualKeyCode as integer, Byref ShiftKey as boolean, byref ControlKey as Boolean, byref AltKey as boolean) as boolean**

Plugin Version: 8.4 Console & Web: Yes Mac: No, Win: Yes, Linux: No, . **Function:** Returns for a given character which combination of key code and modifiers created the character.

**Notes:**

For Character, you should use asc(string), but you may need to convert the string to Windows ANSI text encoding.

Returns 0 on failure.

VK_ LBUTTON	& h01
VK_ RBUTTON	& h02
VK_ CANCEL	& h03
VK_ MBUTTON	& h04
VK_ XBUTTON1	& h05 (Windows 2000 and newer)
VK_ XBUTTON2	& h06 (Windows 2000 and newer)
VK_ BACK	& h08
VK_ TAB	& h09
VK_ CLEAR	& h0C
VK_ RETURN	& h0D
VK_ SHIFT	& h10
VK_ CONTROL	& h11
VK_ MENU	& h12
VK_ PAUSE	& h13
VK_ CAPITAL	& h14
VK_ KANA	& h15
VK_ HANGUL	& h15
VK_ JUNJA	& h17
VK_ FINAL	& h18
VK_ HANJA	& h19
VK_ KANJI	& h19
VK_ ESCAPE	& h1B
VK_ CONVERT	& h1C
VK_ NONCONVERT	& h1D
VK_ ACCEPT	& h1E
VK_ MODECHANGE	& h1F
VK_ SPACE	& h20
VK_ PRIOR	& h21
VK_ NEXT	& h22
VK_ END	& h23
VK_ HOME	& h24
VK_ LEFT	& h25
VK_ UP	& h26
VK_ RIGHT	& h27
VK_ DOWN	& h28
VK_ SELECT	& h29
VK_ PRINT	& h2A
VK_ EXECUTE	& h2B
VK_ SNAPSHOT	& h2C
VK_ INSERT	& h2D
VK_ DELETE	& h2E
VK_ HELP	& h2F
VK_ 0 - VK_ 9	are the same as ASCII "0" - "9" (& h30 - & h39)
VK_ A - VK_ Z	are the same as ASCII "A" - "Z" (& h41 - & h5A)
VK_ LWIN	& h5B
VK_ RWIN	& h5C
VK_ APPS	& h5D
VK_ SLEEP	& h5F
VK_ NUMPAD0	& h60
VK_ NUMPAD1	& h61
VK_ NUMPAD2	& h62
VK_ NUMPAD3	& h63
VK_ NUMPAD4	& h64
VK_ NUMPAD5	& h65
VK_ NUMPAD6	& h66
VK_ NUMPAD7	& h67
VK_ NUMPAD8	& h68
VK_ NUMPAD9	& h69
VK_ MULTIPLY	& h6A
VK_ DIVIDE	& h6B
VK_ F1	& h6C
VK_ F2	& h6D
VK_ F3	& h6E
VK_ F4	& h6F
VK_ F5	& h70
VK_ F6	& h71
VK_ F7	& h72
VK_ F8	& h73
VK_ F9	& h74
VK_ F10	& h75
VK_ F11	& h76
VK_ F12	& h77
VK_ F13	& h78
VK_ F14	& h79
VK_ F15	& h7A
VK_ F16	& h7B
VK_ F17	& h7C
VK_ F18	& h7D
VK_ F19	& h7E
VK_ F20	& h7F
VK_ F21	& h80
VK_ F22	& h81
VK_ F23	& h82
VK_ F24	& h83
VK_ F25	& h84
VK_ F26	& h85
VK_ F27	& h86
VK_ F28	& h87
VK_ F29	& h88
VK_ F30	& h89
VK_ F31	& h8A
VK_ F32	& h8B
VK_ F33	& h8C
VK_ F34	& h8D
VK_ F35	& h8E
VK_ F36	& h8F
VK_ F37	& h90
VK_ F38	& h91
VK_ F39	& h92
VK_ F40	& h93
VK_ F41	& h94
VK_ F42	& h95
VK_ F43	& h96
VK_ F44	& h97
VK_ F45	& h98
VK_ F46	& h99
VK_ F47	& h9A
VK_ F48	& h9B
VK_ F49	& h9C
VK_ F50	& h9D
VK_ F51	& h9E
VK_ F52	& h9F
VK_ F53	& hA0
VK_ F54	& hA1
VK_ F55	& hA2
VK_ F56	& hA3
VK_ F57	& hA4
VK_ F58	& hA5
VK_ F59	& hA6
VK_ F60	& hA7
VK_ F61	& hA8
VK_ F62	& hA9
VK_ F63	& hAA
VK_ F64	& hAB
VK_ F65	& hAC
VK_ F66	& hAD
VK_ F67	& hAE
VK_ F68	& hAF
VK_ F69	& hB0
VK_ F70	& hB1
VK_ F71	& hB2
VK_ F72	& hB3
VK_ F73	& hB4
VK_ F74	& hB5
VK_ F75	& hB6
VK_ F76	& hB7
VK_ F77	& hB8
VK_ F78	& hB9
VK_ F79	& hBA
VK_ F80	& hBB
VK_ F81	& hBC
VK_ F82	& hBD
VK_ F83	& hBE
VK_ F84	& hBF
VK_ F85	& hC0
VK_ F86	& hC1
VK_ F87	& hC2
VK_ F88	& hC3
VK_ F89	& hC4
VK_ F90	& hC5
VK_ F91	& hC6
VK_ F92	& hC7
VK_ F93	& hC8
VK_ F94	& hC9
VK_ F95	& hCA
VK_ F96	& hCB
VK_ F97	& hCC
VK_ F98	& hCD
VK_ F99	& hCE
VK_ F100	& hCF
VK_ F101	& hD0
VK_ F102	& hD1
VK_ F103	& hD2
VK_ F104	& hD3
VK_ F105	& hD4
VK_ F106	& hD5
VK_ F107	& hD6
VK_ F108	& hD7
VK_ F109	& hD8
VK_ F110	& hD9
VK_ F111	& hDA
VK_ F112	& hDB
VK_ F113	& hDC
VK_ F114	& hDD
VK_ F115	& hDE
VK_ F116	& hDF
VK_ F117	& hE0
VK_ F118	& hE1
VK_ F119	& hE2
VK_ F120	& hE3
VK_ F121	& hE4
VK_ F122	& hE5
VK_ F123	& hE6
VK_ F124	& hE7
VK_ F125	& hE8
VK_ F126	& hE9
VK_ F127	& hEA
VK_ F128	& hEB
VK_ F129	& hEC
VK_ F130	& hED
VK_ F131	& hEE
VK_ F132	& hEF
VK_ F133	& hF0
VK_ F134	& hF1
VK_ F135	& hF2
VK_ F136	& hF3
VK_ F137	& hF4
VK_ F138	& hF5
VK_ F139	& hF6
VK_ F140	& hF7
VK_ F141	& hF8
VK_ F142	& hF9
VK_ F143	& hFA
VK_ F144	& hFB
VK_ F145	& hFC
VK_ F146	& hFD
VK_ F147	& hFE
VK_ F148	& hFF

shiftKey	512	shift key down?
alphaLock	1024	alpha lock down?
optionKey	2048	option key down?
controlKey	4096	control key down?
rightShiftKey	8192	right shift key down?
rightOptionKey	16384	right Option key down?
rightControlKey	32768	right Control key down?

shiftKey	512	shift key down?
alphaLock	1024	alpha lock down?
optionKey	2048	option key down?
controlKey	4096	control key down?
rightShiftKey	8192	right shift key down?
rightOptionKey	16384	right Option key down?
rightControlKey	32768	right Control key down?

Command Key	1
Shift	2
CapsLock	4
Option	8
Control	16

kUCKeyActionDown	0	The user is pressing the key.
kUCKeyActionUp	1	The user is releasing the key.
kUCKeyActionAutoKey	2	The user has the key in an auto-key pressed state that is, the user is holding down the key for an extended period of time and is thereby generating multiple key strokes from the single key.
kUCKeyActionDisplay	3	The user is requesting information for key display, as in the Key Caps application.



# Chapter 3

## List of all classes

• GammaFadeMBS	11
• GammaMBS	13
• KeyCodesMBS	7
• PresskeyMBS	16



## Chapter 4

### List of all modules

- RemoteControlMBS

24