

# MBS REALbasic RegEx Plugin Documentation

Christian Schmitz

January 17, 2011

## 0.1 Introduction

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

## 0.2 Content

- 1 List of all topics 3
- 2 All items in this plugin 5
- 3 List of all classes 27

# Chapter 1

## List of Topics

• 2 Regular Expressions	5
– 2.1 class RegExMBS	5
* 2.1.1 Compile(pattern as string) as boolean	6
* 2.1.1 CompileMemory(pattern as memoryblock, ByteOffset as integer) as boolean	7
* 2.1.1 Escape(text as string) as string	7
* 2.1.1 Execute(text as string, start as integer) as integer	7
* 2.1.1 ExecuteMemory(text as memoryblock, ByteOffset as integer, ByteLength as integer) as integer	8
* 2.1.1 Offset(index as integer) as integer	9
* 2.1.1 OffsetCharacters(index as integer) as integer	10
* 2.1.1 Replace(NewText as string) as string	11
* 2.1.1 ReplaceAll(Target as string, NewText as string) as string	11
* 2.1.1 ReplaceSelection(NewText as string) as string	12
* 2.1.1 StringNumber(name as string) as integer	12
* 2.1.1 Study as boolean	12
* 2.1.1 Substring(index as integer) as string	13
* 2.1.1 Substring(name as string) as string	13
* 2.1.1 UTF8supported as boolean	13
* 2.1.1 Version as string	14
* 2.1.2 CompileOptionAnchored as Boolean	14
* 2.1.2 CompileOptionAutoCallOut as Boolean	14
* 2.1.2 CompileOptionCaseLess as Boolean	14
* 2.1.2 CompileOptionDollarEndOnly as Boolean	15
* 2.1.2 CompileOptionDotAll as Boolean	15
* 2.1.2 CompileOptionExtended as Boolean	16

* 2.1.2 CompileOptionFirstLine as Boolean	16
* 2.1.2 CompileOptionMultiline as Boolean	17
* 2.1.2 CompileOptionNoAutoCapture as Boolean	17
* 2.1.2 CompileOptionNoUTF8Check as Boolean	18
* 2.1.2 CompileOptions as Integer	18
* 2.1.2 CompileOptionUngreedy as Boolean	18
* 2.1.2 CompileOptionUTF8 as Boolean	19
* 2.1.2 Count as Integer	19
* 2.1.2 ErrorMessage as String	20
* 2.1.2 ErrorOffset as integer	20
* 2.1.2 ExecuteOptionAnchored as Boolean	20
* 2.1.2 ExecuteOptionNotBOL as Boolean	21
* 2.1.2 ExecuteOptionNotEmpty as Boolean	21
* 2.1.2 ExecuteOptionNotEOL as Boolean	21
* 2.1.2 ExecuteOptionNoUTF8Check as Boolean	21
* 2.1.2 ExecuteOptionPartial as Boolean	22
* 2.1.2 ExecuteOptions as Integer	22
* 2.1.2 ExtraHandle as Integer	22
* 2.1.2 Handle as Integer	22
* 2.1.2 Lasterror as Integer	23
* 2.1.2 Text as String	23
* 2.1.2 TextMemory as String	23

## Chapter 2

# Regular Expressions

### 2.1 class RegExMBS

#### class RegExMBS

class, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** A class for fast Regular Expression Search in a perl compatible way.

**Example:**

```
dim r as new RegExMBS
dim searchString as string = ".o"

if r.Compile(searchString) then

dim s as string="Hello World"

dim start as integer = 0
while r.Execute(s,start)>0

dim p as integer = r.OffsetCharacters(0)
dim l as integer = r.OffsetCharacters(1)-r.OffsetCharacters(0)

MsgBox "Found "+searchString+" on position "+str(p)+" with length "+str(l)+" in """+s+""""

start = r.OffsetCharacters(1)
wend
```

```

else
MsgBox "failed to compile"
end if

```

**Notes:** uses the PCRE library. You may check the PCRE documentation.

### 2.1.1 Methods

#### Compile(pattern as string) as boolean

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Compiles a pattern.

**Example:**

```

dim r as new RegExMBS
dim searchString as string = ".o"

if r.Compile(searchString) then
msgbox "OK"
else
MsgBox "failed to compile"
end if

```

**Notes:**

Some predefined patterns like `\b` do not support Unicode well, so you may work around that by using your own pattern.

Returns true on success and false on failure.  
 ErrorMessage, Lasterror, ErrorOffset and Handle are set.

The following table lists the error codes than may be returned by `Compile()`, along with the error messages that may be returned by both compiling functions.

**CompileMemory(pattern as memoryblock, ByteOffset as integer) as boolean**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.3, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Compiles a pattern.

**Notes:**

Same as Compile, but the text is stored in a memoryblock and must be a 0 terminated C string. Be careful to use valid UTF8 input and provide offset in byte units and not in characters.

**Escape(text as string) as string**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 7.8, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Escapes the string.

**Example:**

```
dim r as RegExMBS
msgbox r.Escape("Hello [ ] ") // shows Hello \[ \]
```

**Notes:**

The string is converted to UTF8 and all the RegEx special characters are escaped.  
Returns "" on low memory.

**Execute(text as string, start as integer) as integer**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Performs a search with the compiled pattern.

**Example:**

```

dim r as RegExMbs
dim s as string
dim c as integer

s="123 ABC 456"

r=new RegExMBS
if r.Compile(" \D+ ") then
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 3 8
// 1 for ubound of the offset array
// 3 for 3 bytes before the matched pattern
// 8 for the 8 bytes before the end of the matched pattern
end if

```

**Notes:**

Returns the number of found offsets.

text must be in UTF-8 text encoding.

Start must be 0 for the first character and the byte offset for other chacters. Do not pass values from OffsetCharacters here!

Return values from Execute:

If Execute() fails, it returns a negative number. The following are defined in the header file:

**ExecuteMemory(text as memoryblock, ByteOffset as integer, ByteLength as integer) as integer**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.3, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Performs a search with the compiled pattern.

**Notes:**

Same as Execute, but the text is stored in a memoryblock.

Be careful to use valid UTF8 input and provide offset and length in byte units and not in characters.

**Offset(index as integer) as integer**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Get the offset in the offset list with given index in bytes.

**Example:**

```

dim r as RegExMbs
dim s as string
dim c as integer

s="123 ABC 456"

r=new RegExMBS
if r.Compile("..") then
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if

r=new RegExMBS
if r.Compile(".\xF6.") then // finds using Unicode codepoint
c=r.Execute(s,0)
MsgBox str(c)+" "+str(r.Offset(0))+" "+str(r.Offset(1))
// shows: 1 4 10
// 1 for ubound of the offset array
// 4 for 4 bytes before the matched pattern
// 10 for the 10 bytes before the end of the matched pattern
end if

```

**Notes:**

If you found a pattern in a string you get here:

Invalid indexes return 0.

Count is the number of entries here.

**OffsetCharacters(index as integer) as integer**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.3, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Get the offset in the offset list with given index in characters.

**Example:**

```

dim r as new RegExMBS
dim searchString as string = ".o"

if r.Compile(searchString) then

dim s as string=" Hello World"

if r.Execute(s,0)>0 then
dim lines(-1) as string

lines.Append str(R.Count)+" offset found."
lines.Append "In Bytes:"
lines.Append " Start of matched patern: "+str(R.Offset(0))
lines.Append " End of matched patern: "+str(R.Offset(1))
lines.Append " Length of matched patern: "+str(R.Offset(1)-r.Offset(0))

lines.Append "In Characters:"
lines.Append " Start of matched patern: "+str(R.OffsetCharacters(0))
lines.Append " End of matched patern: "+str(R.OffsetCharacters(1))
lines.Append " Length of matched patern: "+str(R.OffsetCharacters(1)-r.OffsetCharacters(0))

MsgBox Join(lines,EndOfLine)
end if

else
MsgBox "failed to compile"
end if

```

**Notes:**

This function is identical to Offset(), but returns characters instead of bytes.

Works only with valid UTF-8 strings as input.

Value is calculated on each function call based on Offset(index) and current text.

If you found a pattern in a string you get here:

Invalid indexes return 0.  
Count is the number of entries here.

### **Replace(NewText as string) as string**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Replaces the text on the current found position and returns the complete new text.

**Notes:**

You need to call Execute before.  
Lasterror is set.  
NewText must have UTF-8 text encoding.

\0 references the whole found pattern, \1 to \15 the subexpressions.  
\t is replaced with chr(9), \r and \n with chr(13) and \\ with \.

### **ReplaceAll(Target as string, NewText as string) as string**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Searches the target string for current pattern and replaces all occurrences with the new text.

**Notes:**

You need to call Compile before to initialize the pattern and you should call Study before to optimize the pattern.  
Lasterror is set.  
Target and NewText must have UTF-8 text encoding.

\0 references the whole found pattern, \1 to \15 the subexpressions.  
\t is replaced with chr(9), \r and \n with chr(13) and \\ with \.

**ReplaceSelection(NewText as string) as string**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Replaces the text on the current found position and returns new text for that selection.

**Notes:**

This method is for text editors where you will store result in editfield.seltext to replace the current selection. Lasterror is set.

You need to call Execute before.

NewText must have UTF-8 text encoding.

\0 references the whole found pattern, \1 to \15 the subexpressions.

\t is replaced with chr(9), \r and \n with chr(13) and \\ with \.

**StringNumber(name as string) as integer**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** This convenience function finds the number of a named substring capturing parenthesis in a compiled pattern.

**Notes:**

name: Name whose number is required

The yield of the function is the number of the parenthesis if the name is found, or PCRE\_ERROR\_NO-SUBSTRING otherwise.

**Study as boolean**

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** After you compiled a pattern study can optimize it.

**Notes:**

Only useful if you use Execute several times.

In that case you call one time Compile, one time Study and several times Execute.

Errormessage is set.

### Substring(index as integer) as string

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Returns the subexpression found with the given index.

**Notes:**

Returns "" on any error.

Lasterror is set.

See also:

- 2.1.1 Substring(name as string) as string

13

### Substring(name as string) as string

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Returns the subexpression found with the given name.

**Notes:**

Returns "" on any error.

Lasterror is set.

See also:

- 2.1.1 Substring(index as integer) as string

13

### UTF8supported as boolean

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Whether UTF8 is supported.

**Notes:** If this ever is false, please complain. This plugin is designed to work only on UTF8 strings for best performance.

### Version as string

method from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The version of the PCRE library as an ASCII string.

## 2.1.2 Properties

### CompileOptionAnchored as Boolean

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Force pattern anchoring

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

### CompileOptionAutoCallOut as Boolean

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Compile automatic callouts

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

### CompileOptionCaseLess as Boolean

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Do caseless matching

**Example:**

```
dim r as new RegExMBS
dim searchString as string = "hello"
```

```

r.CompileOptionCaseLess=True

if r.Compile(searchString) then

dim s as string=" Hello World"

if r.Execute(s,0)>0 then
MsgBox "Found: "+mid(s, r.OffsetCharacters(0)+1, r.OffsetCharacters(1)-r.OffsetCharacters(0))
else
MsgBox "nothing found"
end if

else
MsgBox "failed to compile"
end if

```

**Notes:**

The current library version is compiled to match only ASCII characters caseless.  
(Read and Write property)

**CompileOptionDollarEndOnly as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin  
version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: \$ not to match newline at end

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

**CompileOptionDotAll as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin  
version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: . matches anything including endofline

**Example:**

```

dim r as new RegExMBS
dim searchString as string = "<.+>"

```

```

r.CompileOptionDotAll = false // finds only if true

if r.Compile(searchString) then

dim s as string="<Step enable=""True"" +EndOfLine.unix+"id=""93"" name=""Beep"" />"

if r.Execute(s,0)>0 then
MsgBox "Found: " +mid(s, r.OffsetCharacters(0)+1, r.OffsetCharacters(1)-r.OffsetCharacters(0))
else
MsgBox "nothing found"
end if

else
MsgBox "failed to compile"
end if

```

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

### CompileOptionExtended as Boolean

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Ignore whitespace and # comments

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

### CompileOptionFirstLine as Boolean

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Force matching to be before newline

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

**CompileOptionMultiline as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: `^` and `$` match newlines within data

**Example:**

```

dim r as RegExMBS
dim n,i,c as integer
dim s as string

s=ReplaceLineEndings(EditField1.text,EndOfLine.UNIX)

r=new RegExMBS

'r.CompileOptionFirstLine=True
r.CompileOptionMultiline=True

if r.Compile("^...$ ") then
n=0
do
c=r.Execute(s,n)

if c>0 then
MsgBox r.Substring(0)
n=r.Offset(1)
end if

loop until c=0
end if

```

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

**CompileOptionNoAutoCapture as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Disable numbered capturing parentheses (named ones available)

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

### **CompileOptionNoUTF8Check as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Do not check the pattern for UTF-8 validity.

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

### **CompileOptions as Integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.3, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The internal value of all the compile options.

**Notes:**

You can get and set the bits using the CompileOption\* Boolean properties.

(Read only property)

### **CompileOptionUngreedy as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Invert greediness of quantifiers.

**Notes:**

Basically this is about whether to find the next matching item or the last matching item in the whole string. Matching the next item is always much faster.

(Read and Write property)

**CompileOptionUTF8 as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Compile: Run in UTF-8 mode.

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

**Count as Integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Number of offsets found.

**Example:**

```

dim r as new RegExMBS
dim searchString as string = ".o"

if r.Compile(searchString) then

dim s as string=" Hello World"

if r.Execute(s,0)>0 then
dim lines(-1) as string

lines.Append str(R.Count)+" offset found."
lines.Append "In Bytes:"
lines.Append " Start of matched patern: "+str(R.Offset(0))
lines.Append " End of matched patern: "+str(R.Offset(1))
lines.Append " Length of matched patern: "+str(R.Offset(1)-r.Offset(0))

lines.Append "In Characters:"
lines.Append " Start of matched patern: "+str(R.OffsetCharacters(0))
lines.Append " End of matched patern: "+str(R.OffsetCharacters(1))
lines.Append " Length of matched patern: "+str(R.OffsetCharacters(1)-r.OffsetCharacters(0))

MsgBox Join(lines,EndOfLine)
end if

else
MsgBox "failed to compile"
end if

```

**Notes:** (Read only property)

### **ErrorMessage as String**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The last message reported.

**Notes:**

Set by Study and Compile.  
(Read only property)

### **ErrorOffset as integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The last error offset from the compile function.

**Notes:** (Read only property)

### **ExecuteOptionAnchored as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Execute: Match only at the first position

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

**ExecuteOptionNotBOL as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Execute: Subject is not the beginning of a line

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

**ExecuteOptionNotEmpty as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Execute: An empty string is not a valid match

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

**ExecuteOptionNotEOL as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Execute: Subject is not the end of a line

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

**ExecuteOptionNoUTF8Check as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Execute: Do not check the subject for UTF-8 validity

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

**ExecuteOptionPartial as Boolean**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** Option for Execute: Return PCRE\_ERROR\_PARTIAL for a partial match

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

**ExecuteOptions as Integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.3, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The internal value of all the execute options.

**Notes:**

You can get and set the bits using the CompileOption\* Boolean properties.

(Read only property)

**ExtraHandle as Integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The handle to the extra data structure.

**Notes:** (Read only property)

**Handle as Integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The handle to the pattern data structure.

**Notes:** (Read only property)

**Lasterror as Integer**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The last error code reported.

**Notes:**

0 is no error and -1 is some parameter error.

(Read only property)

**Text as String**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.2, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The text used for the last successful compile call.

**Notes:** (Read only property)

**TextMemory as String**

property from class RegExMBS, Regular Expressions, MBS REALbasic RegEx Plugin (RegEx), Plugin version: 6.3, console safe, Mac OS X: Works, Windows: Works, Linux x86: Works.

**Function:** The text used for the last successful compiletext call.

**Notes:** (Read only property)

0 no error  
1 \at end of pattern  
2 \c at end of pattern  
3 unrecognized character follows \  
4 numbers out of order in { } quantifier  
5 number too big in { } quantifier  
6 missing terminating ] for character class  
7 invalid escape sequence in character class  
8 range out of order in character class  
9 nothing to repeat  
10 operand of unlimited repeat could match the empty string  
11 internal error: unexpected repeat  
12 unrecognized character after (?  
13 POSIX named classes are supported only within a class  
14 missing )  
15 reference to non-existent subpattern  
16 erroffset passed as NULL  
17 unknown option bit(s) set  
18 missing ) after comment  
19 parentheses nested too deeply  
20 regular expression too large  
21 failed to get memory  
22 unmatched parentheses  
23 internal error: code overflow  
24 unrecognized character after (?<  
25 lookbehind assertion is not fixed length  
26 malformed number after ?(  
27 conditional group contains more than two branches  
28 assertion expected after ?(  
29 (?R or (?digits must be followed by )  
30 unknown POSIX class name  
31 POSIX collating elements are not supported  
32 this version of PCRE is not compiled with PCRE\_ UTF8 support  
33 spare error  
34 character value in \x { ... } sequence is too large  
35 invalid condition ?(0)  
36 \C not allowed in lookbehind assertion  
37 PCRE does not support \L, \l, \N, \U, or \u  
38 number after ?C is >255  
39 closing ) for ?C expected  
40 recursive call could loop indefinitely  
41 unrecognized character after ?P  
42 syntax error after ?P  
43 two named groups have the same name  
44 invalid UTF-8 string  
45 support for \P, \p, and \X has not been compiled

PCRE_ERROR_NOMATCH	-1	The subject string did not match the pattern.
PCRE_ERROR_NULL	-2	Either code or subject was passed as "".
PCRE_ERROR_BADOPTION	-3	An unrecognized bit was set in the options argument.
PCRE_ERROR_BADMAGIC	-4	PCRE stores a 4-byte "magic number" at the start of the compiled code, to catch the case when it is passed a junk pointer and to detect when a pattern that was compiled in an environment of one endianness is run in an environment with the other endianness. This is the error that PCRE gives when the magic number is not present.
PCRE_ERROR_UNKNOWN_NODE	-5	While running the pattern match, an unknown item was encountered in the compiled pattern. This error could be caused by a bug in PCRE or by overwriting of the compiled pattern.
PCRE_ERROR_NOMEMORY	-6	If a pattern contains back references, but the ovector that is passed to <code>Execute()</code> is not big enough to remember the referenced substrings, PCRE gets a block of memory at the start of matching to use for this purpose. If the call via <code>pcre_malloc()</code> fails, this error is given. The memory is automatically freed at the end of matching.
PCRE_ERROR_MATCHLIMIT	-8	The backtracking limit, as specified by the <code>match_limit</code> field in a <code>pcre_extra</code> structure (or defaulted) was reached.
PCRE_ERROR_RECURSIONLIMIT	-21	The internal recursion limit, as specified by the <code>match_limit_recursion</code> field in a <code>pcre_extra</code> structure (or defaulted) was reached.
PCRE_ERROR_CALLOUT	-9	This error is never generated by <code>Execute()</code> itself. It is provided for use by callout functions that want to yield a distinctive error code. See the <code>precallout</code> documentation for details.
PCRE_ERROR_BADUTF8	-10	A string that contains an invalid UTF-8 byte sequence was passed as a subject.
PCRE_ERROR_BADUTF8_OFFSET	-11	The UTF-8 byte sequence that was passed as a subject was valid, but the value of <code>startoffset</code> did not point to the beginning of a UTF-8 character.
PCRE_ERROR_PARTIAL	-12	The subject string did not match, but it did match partially. See the <code>prepartial</code> documentation for details of partial matching.
PCRE_ERROR_BADPARTIAL	-13	The PCRE <code>PARTIAL</code> option was used with a compiled pattern containing items that are not supported for partial matching. See the <code>prepartial</code> documentation for details of partial matching.
PCRE_ERROR_INTERNAL	-14	An unexpected internal error has occurred. This error could be caused by a bug in PCRE or by overwriting of the compiled pattern.
PCRE_ERROR_BADCOUNT	-15	This error is given if the value of the <code>ovecsize</code> argument is negative.

index	offset
0	start of matched pattern
1	end of matched pattern
2	start of subexpression 1
3	end of subexpression 1
2*n	start of subexpression n
2*n+1	end of subexpression n

index	offset
0	start of matched pattern
1	end of matched pattern
2	start of subexpression 1
3	end of subexpression 1
2*n	start of subexpression n
2*n+1	end of subexpression n



## Chapter 3

### List of all classes

- RegExMBS

5