

ifRegex

Please see the PCRE documentation (<http://www.pcre.org/>) for documentation on the PCRE library used for regular expression matching. See [the perlre man page](#) for complete documentation of the possible regular expressions this library can parse and match. In general, most Perl compatible regular expressions are supported.

Implemented By

- `roRegex`

Supported Methods

- `IsMatch(str as String) as Boolean`
- `Match(str as String) as Object`
- `Replace(str as String, replacement as String) as String`
- `ReplaceAll(str as String, replacement as String) as String`
- `Split(str as String) as Object`
- `MatchAll(str as String) as Object`

Description of Methods

`IsMatch(str as String) as Boolean`

Returns true if `str` matches the matching pattern.

Example from Brightscript Debugger Interactive Shell:

```
> r = CreateObject("roRegex", "cad", "i")
> ? r.IsMatch("AbraCadabra")
true
>
```

`Match(str as String) as Object`

Returns an `roArray` of matched substrings from `str`.

If no match was made, an empty array is returned.

If a match was made, the entire match is returned in `array[0]`. If there are no parenthetical substrings this is the only entry in the array.

If the matching pattern contains `N` parenthetical substrings, the relevant substrings are returned as an array of length `N+1`, where `array[0]` is again the entire match and each additional entry in the array is the match for the corresponding parenthetical expression.

Example from Brightscript Debugger Interactive Shell:

```
> r = CreateObject("roRegex", "(a|(z))(bc)", "")
> ? r.Match("abcd")
abc
a

bc
>
```

Note that entry 2 of the array is an empty string, corresponding to the parenthesized "z" in the regular expression.

Replace(str as String, replacement as String) as String

Replaces the first occurrence of a matching pattern in str with replacement and returns the result.

The replacement may contain numbered back-references to parenthetical substrings.

Example from Brightscript Debugger Interactive Shell:

```
> r = CreateObject("roRegex", "(\\d+)\\s+(\\w+)", "")
> ? r.Replace("123 abc", "word:\\2 number:\\1")
> word:abc number:123
>
```

ReplaceAll(str as String, replacement as String) as String

Similar to Replace() but replaces all occurrences of the matching pattern, not just the first one.

Examples from Brightscript Debugger Interactive Shell:

```
> r = CreateObject("roRegex", "a", "i")
> ? r.ReplaceAll("Abracadabra", "x")
xbrxcxdxbrx

> r = CreateObject("roRegex", "a", "")
> ? r.ReplaceAll("Abracadabra", "x")
Abrxcxdxbrx
>
```

Split(str as String) as Object

Uses the matching pattern as a separator and splits the string on the separator boundaries.

Returns an roList of substrings of str that were separated by strings which match the pattern in the CreateObject call. The separator strings are not returned.

If no matches were found, the returned list contains a single item with the string unchanged.

Examples from Brightscript Debugger Interactive Shell:

```
> r = CreateObject("roRegex", ",", "") ' split on comma
> ? r.Split("first, second, third and fourth")
first
second
third and fourth
```

```
> r = CreateObject("roRegex", "/+", "") ' split on one or more slashes
> ? r.Split("example.com/images///2012/cat.jpg")
> example.com
> images
> 2012
> cat.jpg
>
```

Note that in the first example, the last two strings begin with a space, since each comma in the string was followed by a space.

MatchAll(str as String) as Object

Available since firmware version 8.1

`MatchAll()` adds the ability to return all matches of the specific regular expression pattern in the target string as an array where the first element is the full matched string and if there are any capture groups those are returned in subsequent array elements.

Example:

```
> r = CreateObject("roRegex", "\d+", "")
> arr = r.MatchAll("123 456 789")
> print FormatJSON(arr)
[["123"],["456"],["789"]]
```