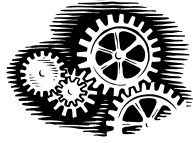


# .NET FRAMEWORK REGULAR EXPRESSIONS



## SINGLE CHARACTERS

| Use    | To match any character    |
|--------|---------------------------|
| [set]  | In that set               |
| [^set] | Not in that set           |
| [a-z]  | In the a-z range          |
| [^a-z] | Not in the a-z range      |
| .      | Any except \n (new line)  |
| \char  | Escaped special character |

## CONTROL CHARACTERS

| Use     | To match                | Unicode |
|---------|-------------------------|---------|
| \t      | Horizontal tab          | \u0009  |
| \v      | Vertical tab            | \u000B  |
| \b      | Backspace               | \u0008  |
| \e      | Escape                  | \u001B  |
| \r      | Carriage return         | \u000D  |
| \f      | Form feed               | \u000C  |
| \n      | New line                | \u000A  |
| \a      | Bell (alarm)            | \u0007  |
| \c char | ASCII control character | –       |

## NON-ASCII CODES

| Use    | To match character with        |
|--------|--------------------------------|
| \octal | 2-3 digit octal character code |
| \x hex | 2-digit hex character code     |
| \u hex | 4-digit hex character code     |

## CHARACTER CLASSES

| Use       | To match character                    |
|-----------|---------------------------------------|
| \p{ctgry} | In that Unicode category or block     |
| \P{ctgry} | Not in that Unicode category or block |
| \w        | Word character                        |
| \W        | Non-word character                    |
| \d        | Decimal digit                         |
| \D        | Not a decimal digit                   |
| \s        | White-space character                 |
| \S        | Non-white-space char                  |

## QUANTIFIERS

| Greedy | Lazy   | Matches           |
|--------|--------|-------------------|
| *      | *?     | 0 or more times   |
| +      | +?     | 1 or more times   |
| ?      | ??     | 0 or 1 time       |
| {n}    | {n}?   | Exactly n times   |
| {n,}   | {n,}?  | At least n times  |
| {n,m}  | {n,m}? | From n to m times |

## ANCHORS

| Use | To specify position                            |
|-----|--|
| ^   | At start of string or line                     |
| \A  | At start of string                             |
| \z  | At end of string                               |
| \Z  | At end (or before \n at end) of string         |
| \$  | At end (or before \n at end) of string or line |
| \G  | Where previous match ended                     |
| \b  | On word boundary                               |
| \B  | Not on word boundary                           |

## GROUPS

| Use                 | To define                      |
|---------------------|--------------------------------|
| (exp)               | Indexed group                  |
| (?<name>exp)        | Named group                    |
| (?<name1-name2>exp) | Balancing group                |
| (?:exp)             | Noncapturing group             |
| (?=exp)             | Zero-width positive lookahead  |
| (?!exp)             | Zero-width negative lookahead  |
| (?<=exp)            | Zero-width positive lookbehind |
| (?<!exp)            | Zero-width negative lookbehind |
| (?>exp)             | Non-backtracking (greedy)      |

## INLINE OPTIONS

| Option | Effect on match    |
|--------|--------------------|
| i      | Case-insensitive   |
| m      | Multiline mode     |
| n      | Explicit (named)   |
| s      | Single-line mode   |
| x      | Ignore white space |

| Use                | To   |
|--------------------|--|
| (?imnsx-imnsx)     | Set or disable the specified options                       |
| (?imnsx-imnsx:exp) | Set or disable the specified options within the expression |

June 2014

© 2014 Microsoft. All rights reserved.

## BACKREFERENCES

| Use                         | To match      |
|-----------------------------|---------------|
| <code>\n</code>             | Indexed group |
| <code>\k&lt;name&gt;</code> | Named group   |

## ALTERNATION

| Use                   | To match                        |
|-----------------------|---------------------------------|
| <code>a   b</code>    | Either <i>a</i> or <i>b</i>     |
| <code>(?exp)</code>   | yes if <i>exp</i> is matched    |
| <code>yes   no</code> | no if <i>exp</i> isn't matched  |
| <code>(?name)</code>  | yes if <i>name</i> is matched   |
| <code>yes   no</code> | no if <i>name</i> isn't matched |

## SUBSTITUTION

| Use                   | To substitute                              |
|-----------------------|--|
| <code>\$n</code>      | Substring matched by group number <i>n</i> |
| <code>\${name}</code> | Substring matched by group <i>name</i>     |
| <code>\$\$</code>     | Literal \$ character                       |
| <code>\$&amp;</code>  | Copy of whole match                        |
| <code>\$`</code>      | Text before the match                      |
| <code>\$'</code>      | Text after the match                       |
| <code>\$+</code>      | Last captured group                        |
| <code>\$_</code>      | Entire input string                        |

## COMMENTS

| Use                       | To                 |
|---------------------------|--------------------|
| <code>(?# comment)</code> | Add inline comment |
| <code>#</code>            | Add x-mode comment |

For detailed information and examples, see <http://aka.ms/regex>

To test your regular expressions, see <http://regexlib.com/RETester.aspx>

## SUPPORTED UNICODE CATEGORIES

| Category  | Description                     |
|-----------|---------------------------------|
| <b>Lu</b> | Letter, uppercase               |
| <b>Ll</b> | Letter, lowercase               |
| <b>Lt</b> | Letter, title case              |
| <b>Lm</b> | Letter, modifier                |
| <b>Lo</b> | Letter, other                   |
| <b>L</b>  | Letter, all                     |
| <b>Mn</b> | Mark, nonspacing combining      |
| <b>Mc</b> | Mark, spacing combining         |
| <b>Me</b> | Mark, enclosing combining       |
| <b>M</b>  | Mark, all diacritic             |
| <b>Nd</b> | Number, decimal digit           |
| <b>Nl</b> | Number, letterlike              |
| <b>No</b> | Number, other                   |
| <b>N</b>  | Number, all                     |
| <b>Pc</b> | Punctuation, connector          |
| <b>Pd</b> | Punctuation, dash               |
| <b>Ps</b> | Punctuation, opening mark       |
| <b>Pe</b> | Punctuation, closing mark       |
| <b>Pi</b> | Punctuation, initial quote mark |
| <b>Pf</b> | Punctuation, final quote mark   |
| <b>Po</b> | Punctuation, other              |
| <b>P</b>  | Punctuation, all                |
| <b>Sm</b> | Symbol, math                    |
| <b>Sc</b> | Symbol, currency                |
| <b>Sk</b> | Symbol, modifier                |
| <b>So</b> | Symbol, other                   |
| <b>S</b>  | Symbol, all                     |
| <b>Zs</b> | Separator, space                |
| <b>Zl</b> | Separator, line                 |
| <b>Zp</b> | Separator, paragraph            |
| <b>Z</b>  | Separator, all                  |
| <b>Cc</b> | Control code                    |
| <b>Cf</b> | Format control character        |
| <b>Cs</b> | Surrogate code point            |
| <b>Co</b> | Private-use character           |
| <b>Cn</b> | Unassigned                      |
| <b>C</b>  | Control characters, all         |

For named character set blocks (e.g., Cyrillic), search for "supported named blocks" in the MSDN Library.

## REGULAR EXPRESSION OPERATIONS

Class: `System.Text.RegularExpressions.Regex`

### Pattern matching with Regex objects

| To initialize with | Use constructor                                    |
|--------------------|--|
| Regular exp        | <code>Regex(String)</code>                         |
| + options          | <code>Regex(String, RegexOptions)</code>           |
| + time-out         | <code>Regex(String, RegexOptions, TimeSpan)</code> |

### Pattern matching with static methods

Use an overload of a method below to supply the regular expression and the text you want to search.

### Finding and replacing matched patterns

| To                    | Use method  |
|-----------------------|---|
| Validate match        | <code>Regex.IsMatch</code>  |
| Retrieve single match | <code>Regex.Match (first)</code><br><code>Match.NextMatch (next)</code> |
| Retrieve all matches  | <code>Regex.Matches</code>  |
| Replace match         | <code>Regex.Replace</code>  |
| Divide text           | <code>Regex.Split</code>  |
| Handle char escapes   | <code>Regex.Escape</code><br><code>Regex.Unescape</code>                |

### Getting info about regular expression patterns

| To get        | Use Regex API   |
|---------------|---|
| Group names   | <code>GetGroupNames</code><br><code>GetGroupNameFromNumber</code>   |
| Group numbers | <code>GetGroupNumbers</code><br><code>GetGroupNumberFromName</code> |
| Expression    | <code>ToString</code>   |
| Options       | <code>Options</code>  |
| Time-out      | <code>MatchTimeOut</code>   |
| Cache size    | <code>CacheSize</code>  |
| Direction     | <code>RightToLeft</code>  |