

Batch-RegEx



<http://telcontar.net/Software/Batch-RegEx/>

Summary	1
Basic operation	1
Syntax.....	2
Blank lines and comments.....	2
Match and replacement expressions	2
Defaults	2
Expression flags	3
Global flags	3
Match expressions.....	3
Replacement expressions.....	4
Version history.....	4
Licence	5

Summary

Batch-RegEx is a regular expression batch processor that applies a sequence (a “set”) of one or more regular expression replacements to one or more files or folders. Folders are processed recursively.

Version 1.0 was created in 2006 on my Macintosh as a tool for scraping FrontPage-generated HTML pages during a website overhaul. While I made a point of constructing it as a generic application, it has numerous arbitrary restrictions for the sake of simplicity.

Nonetheless, the program is obtuse enough to justify having actual documentation!

Basic operation

The following sequence indicates how the program operates:

1. In the *Regular expression list* field, enter one or more pairs of match and replacement expressions
2. Drop one or more files and folders onto the *Source* field as indicated to specify the files to be processed
3. If desired, specify a space-delimited list of file extensions to filter: only files whose extensions are specified will be processed

4. Click *Go*

If *Create backup subfolders* is selected, copies of the modified files will be created in backup subfolders (named with the current date and time). One such subfolder will be created inside each folder dropped onto the program in step 2 above, and one folder will be created alongside each set of files. (Normally, all files dropped onto the program will share the same parent folder, but this is not guaranteed to be the case.) Processed subfolders will share the same backup folder as their parents, with matching subfolders created within the backup folder. Note that these backup folders only contain the original copies of files that were modified; unmodified files will not be backed up.

Syntax

Batch-RegEx is designed to apply multiple replacements to each file processed. The syntax is orientated towards this goal, and is designed for a reasonable amount of clarity (inasmuch as this is ever possible with regular expressions). Match (search) and replacement expressions are placed onto alternate lines.

Blank lines and comments

Blank lines in the expression list may be placed anywhere, and are ignored. Comments begin with "`\` " (backslash, space) in a form inspired by Acorn assembly syntax.

Match and replacement expressions

In their simplest form, match and replacement expressions resemble standard regular expressions. The regular expression engine is [PCRE](#). Per PCRE, backreferences in replacement expressions use backslashes. The following expression set replaces all `<i>` HTML tags with `` tags, and all `` tags with `` tags:

```
\ Upgrade i tags to em
<i>(.*?)</i>
<em>\1</em>
```

```
\ Upgrade b tags to strong
<b>(.*?)</b>
<strong>\1</strong>
```

Defaults

The default behaviour is as follows:

- Greedy matching is disabled, although use of "`. * ?`" and "`. + ?`" may still be required
- Matches are performed in a case-insensitive manner
- The dot metacharacter (`.`) does not match newlines
- Only the first match within the file is replaced

The above behaviour is not the default for other programs such as perl or grep, and was most likely selected based on the needs of the task for which Batch-RegEx was created. At the time, global flags did not exist.

Expression flags

The following flags can be used to change the behaviour of the program:

Flag	Description
d	Enable '.' (mnemonic: dot) to match newlines
c	Enable case-sensitive matching
m	Replace all matches in the file instead of only the first (mnemonic: 'm' for multiple replace)
g	Enable greedy matching

Global flags

The default behaviour can be set using a line of the form "`?{...}`". One or more expression flags can be specified to change the behaviour of every expression that follows. For example, the following line will enable case-sensitive replacement of all subsequent matches:

```
?{cm}
```

All flags specified within the braces ("`{...}`") are enabled, and all flags not specified are disabled. As such, each use of a global flags specifier is self-evident, indicating exactly what flags are active. Multiple such lines can be used in the same expression set, and each line applies to all expressions that follow, up to the next global flags directive.

The following two lines have the net effect of setting only the **m** flag; the **c** flag set by the first line is cleared due to its omission from the second line:

```
?{c}  
?{m}
```

Global flags can also be set in the Options dialog box; flags set in Options will be applied to each new expression set by way of a global flags directive automatically entered into the regular expression list field. As such, changes to global settings will not affect previously-saved expression sets that specify their own global flags. As a note, it is valid to clear all global flags with an empty flags list, thus: `?{ }`

Match expressions

Expression flags can also be set on a per-expression basis. For example, the following expression is set to replace all matches:

```
?m <i>(.)</i>  
<em>\1</em>
```

The "`?m`" term at the start of the match expression sets the flags. Flags are set by lines of the form "`?flags pattern`".

To override a global flag, precede a flag with a hyphen-minus. For example, the replacement below is performed in a case-insensitive manner, because the **c** flag set by the global flags line is overridden by the "`-c`" term in the match expression:

```
?{cm}
```

```
\ Override the global "c" flag here
?-c foo
bar
```

Replacement expressions

A single pseudo-flag "-" is used with replacement expressions to indicate that the matched text is to be deleted. A blank line cannot be used, as they are used for presentation and are consequently ignored!

For example, to delete all <o> tags from a document:

```
?m </?o>
\ Replace with nothing
?-
```

Version history

1.4b1

- Adapted for modern Windows editions (Vista upwards)
- The backup files are now created in subfolders to preserve their original names and prevent them from being processed again in error
- The creation timestamp is preserved when writing out the modified files
- When processing subfolders, relative pathnames were reported incorrectly (missing leading character and extraneous backslash)
- *Open sets folder* item in the Expressions menu (to allow for expression set files to be renamed, copied and deleted)
- A summary is given at the end of a batch process (the count of files that were processed, skipped, and aborted due to errors)
- Changed expression sets are indicated with an asterisk in the title bar
- File extension filtering: only process files with the extensions specified
- Expression set name now comes first in the window title
- *Expressions* → *Start over* is now Ctrl+N (as in New) as Ctrl+R typically indicates Refresh!
- *Go* button no longer the default button, to avoid accidental operation
- Tips window (now Syntax guide) text was not read-only!
- A new icon that is potentially marginally less dreadful than the previous icon

1.3.3

- Phantom buttons that appeared in 1.3.2 removed (you never saw them)
- Creation of backup files is now optional [Jaco Gunter]

1.3.2

- Use of ?{g} now works, instead of setting ?{d} [Jonathan Livni]

1.3.1

- Passing the program an entire folder no longer causes it to process all the backup files and then stop short [Jonathan Livni]
- Relative paths are shown in the progress window for items within folders for clarity

1.3

- Now supports preferences for setting default flags [Anthony Bowden]

- Global effect flag lines now supported
- Global effect flags are now negated in expressions with `-'

1.2.4

- Window positions are now memorised

1.2.3

- The number of matched rules is reported for each file; any files for which no rules matched are skipped and left untouched.
- The save changes dialog no longer maps the escape key to No instead of Cancel!

1.2.2

- Improved message dialog boxes

1.2.1

- Comments are now supported in the expression list (lines beginning with "`\ ")
- When performing multiple operations on the same files, the correct files are read back in, instead of the back-up copies being read instead [Anthony Bowden]
- Expressions —Start over implemented
- Prompt to save changes when loading a saved set or starting over
- The output file now has the same creator code as the input file
- Supplied mailto: shortcut file no longer contains an e-mail address long since invalid before this app was conceived (!!)

1.2

- Saved expression sets: save and recall expression lists
- Output window no longer displays nonsensical progress

1.1.1

- Instead of writing the output to a new file, each input file is renamed as a back-up, and the output file replaces the input file.
- Correct Web page URL for the about box Web hyperlink!

1.1

- Initial Microsoft Windows support
- An initial blank link in the expression list no longer causes a hang

Licence

Batch-RegEx is copyright 2006–2015 Daniel Beardsmore. The Batch-RegEx application (“program”) and source code are made available under the zlib licence, the terms of which are included in a separate document. The program may be used without restriction free of charge.