Bulk Rename Utility

A software utility to rename files and folders

TGRMN Software

Version: 4.1.1

Table of Contents

Overview

Bulk Rename Utility is a software utility for Windows which allows you to rename multiple files and folders, based upon flexible criteria. For example, you can add a prefix or suffix to a file, or you can change three letters with four other letters in a filename. There are lots of ways to manipulate file and folder names.

Bulk Rename Utility supports many flexible renaming criteria, including:

- Rename files and/or folders
- Support for Regular Expressions
- Change the file name or use a fixed filename
- Convert the extension to lower case, upper case, title case - or replace it - or add a secondary extension - or remove it
- Replace text with other text
- Add a fixed prefix or suffix
- Add text to the middle of a name
- Move text to the start/end/middle of a name
- Remove the first *n* characters
- Remove the last *n* characters
- Remove from the *nth* character to the *nth* character
- Remove a list of characters
- Remove all digits, characters or symbols
- Remove double-spaces
- Crop text before or after a fixed character or text (or from anywhere in a name using a wildcard)
- Change to upper case, lower case, title case, sentence case - all with user-defined exceptions
- Auto-number files, with a prefix or suffix, and with a predefined minimum-length if required
- Prefix or suffix the filename with various dates (date-modified, accessed, created, media date, or today's date), in many different formats with variable separators
- Pad numbers
- Reformat dates
- Append the folder name to the file (with multiple levels)
- Store your frequently-used renaming criteria as Favourites
- Copy or move the renamed files to a new location
- Set or clear the Windows attributes on the files (read-only, archived, system, hidden)
- Rename files according to a list (from -> to pairs separated by |)
- Process files in folders and also sub-folders very quickly
- Change file or folder created, modified and accessed timestamps
- Convert Roman numerals to upper or lower case
- Log all activity to a text-file
- Create an UNDO batch file
- Preview all operations before actually renaming.
- Extract EXIF tags from JPEG, TIFF, HEIC and camera RAW files
- Extract certain ID3 tags (V1 and V1.1) from MP3 files
- Extract the Windows Item Date and use it to rename. The Item Date is the primary date of interest for an item, such as videos, HEIC, RAW camera files. In the case of photos, for example, this property maps to Date Taken
- Extract and use the Windows File Properties to rename
- Extract and use PDF and MediaInfo (Video) Properties
- Format all renaming tags with formatters
- Add file hashes to file names (crc32, MD5, SHA)
- Supports renaming using PDF metadata and MediaInfo for video metadata
- Link file extensions for files to be renamed together

- Create and move files into folders based on file names or file dates ('folderize')
- Drag files directly from within Windows Explorer
- Support for renaming scripts using Javascript
- Dark Mode

The application supports Favourites, which allows quick access to frequently-used renaming configurations.
Directory recursion is also fully supported, allowing you to process folders and sub-folders.
If you have suggestions for enhancements or improvements, please get in touch.

Manual Version: 4.1.1

Getting Started

Bulk Rename Utility is straightforward to use.

There are four simple rules to follow:

1. Only complete the boxes you are interested in. For example, don't assume you have to put something in the Regular Expressions box, or the New Location box, etc.
2. IMPORTANT: You have to select files to rename from the file list before the *Rename* button is enabled.
3. The "new name" column will always be shown in the list, so you can experiment until you get the desired result before actually committing the changes.
4. Almost all the boxes on the screen are ONLY for affecting the file name, NOT the extension. The only box which affects the extension is box 11. There is an option in the "Renaming Options' menu to 'Rename File Extensions as Being Part of the File Names' if needed.

So for example, if you just want to change the filename case to upper case, all you need to do is:

1. Change the drop-down in box 4 to "Upper".
2. Select the files you want to rename:
   - Clicking on individual files.
   - Shift + click to select a contiguous range.
   - Ctrl + click to select non-contiguous files.
   - Ctrl + A to select all files (or use the “Select All” button)
   - Clicking on the left or right of the file list and dragging to select multiple files with the mouse.
3. Press the **Rename** button (or **Preview** to review all changes before renaming).

And that's it! No worrying about regular expressions, new folders, auto-numbers, dates etc.
You only need to use the boxes you are interested in. Don't want to move the files to a new folder? Leave the New Location fields blank. Don't want to change the file name case? Leave the *Case* fields as their default values.

It is simple, only use the boxes you are interested in.

➔ As of version 4, you can move the renaming criteria into a separate Window, using Menu -> Display Options -> Detach Renaming Criteria (F12). And you can maximize the file list using Menu -> Display Options -> List -> Maximize File List (F9).

All Renaming Options

When you run the application for the first time you will be presented with the main window, located in the center of the screen. The main screen is laid out in a similar way to Windows Explorer. Below the menu bar are two panes, the left of which is a tree view providing access to your system drives and folders. The right-hand pane is a list view of the files and folders contained within the currently selected branch of the tree shown over on the left.

Below the navigation panes are a series of controls. The controls are grouped together according to function - for example, all the facilities to remove text from a filename. Only complete the boxes you are interested in. For example, don't assume you HAVE to put something in the Regular Expressions box, or the New Location box, etc.

The rules used by the Bulk Rename Utility to rename files are processed in the order that the controls appear. i.e. from left to right. Names are not actually changed until the "Rename" button is clicked, but you can always see a preview of the proposed filename in the *New Name* column. Note that this column is only updated for the files which are selected.

Most rename text fields have two small icons on the right:

- *Chevron (arrow down)*: Opens a menu of Recent Entries for that field and, where supported, lets you insert Renaming Tags.

- *Six-dot button:* Opens the Zoom window. The Zoom window also shows the field’s Recent Entries and allows Renaming Tag insertion (where supported).

Tip: Right-click the six-dot button to go directly to the Insert Renaming Tags dialog.

RegEx (1)

Regular Expression. It allows you to enter a regular expression *Match* and *Replace*. More details can be found here. If you are not using Regular Expressions then both of these fields must be left blank. Do not confuse these fields with the Replacements fields found elsewhere on the screen (Replace (3), see below).

Inc. Ext.: select this option to apply the regular expression to the entire name and extension.
Simple: enables Simple Regular Expressions.
V2: enables Regular Expressions Version 2.

Assistant: Click to open the RegEx Assistant in your browser. The RegEx Assistant uses AI to help you build, test, and refine regular expressions for your renaming tasks. Just describe what you need (for example “remove trailing numbers,” “match dates in YYYY-MM-DD,” or “swap text around”) and it will generate a working regular expression you can try on your file list. The link can be hidden in the Bulk Rename Utility Preferences.

Note: It is also possible to use *<clip>* in the *Replace* field. <clip> will be substituted with the current text content of the Windows Clipboard.

Note:

- Extended version 2 EXIF Tags are supported in *Replace*. They can be added as text, number or as a formatted date to a file name. Read More.
- All Windows File Properties tags are also supported in *Replace*. They can be added as text, number or as a formatted date to a file name. Read More.
  To enable Windows File Properties tags support, activate the option *'Extract Windows File Properties'* from the Renaming menu.
- The following hash value tags are also supported: <(hash:crc32)>, <(hash:md5)>,<(hash:sha1)>,<(hash:sha256)>,<(hash:sha3)>,<(hash:keccak)> in *Replace*. These tags will be replaced by the corresponding hash value of the file and can be used to add a hash value to the name.

- Other renaming tags can be used too.

Click on ☰ to open the multiple RegEx editor, where you can enter multiple Match and Replace pairs and load/save them from/to a file. When loading from a file, each pair must be on its own line, with the *Match* and *Replace* values separated by a comma, tab or |.

Name (2)

File Name. It allows you to manage the file name.

Name drop-down:

- *Keep* - ensure that the original filename is not changed (default).
- *Remove* - completely erase the filename from the selected items.
- *Fixed* - specify a new filename in the box for all selected items. Renaming tags can be used. Renaming tag formatting using the <name> (=current file name) tag can be very powerful.
- *Reverse* - reverse the name, e.g. 12345.txt becomes 54321.txt.
- *Pad Numbers* - Pad the numbers found in the name. The first number that is found in the name is padded. Use *Name Segment* to pad a different number, e.g., not the first one. Specify the amount of padding in the text box, e.g., 3, 4, 5 etc. By default, 0 is used for the padding, but you can specify a different character to use after >. For instance, 5>A pads the first number found to 5 digits using A instead of 0. If just 5 is entered, and a padding character is not specified, then 0 is used to pad. The format to use is *pad_amount> pad_character,* with *pad_character* optional.
- *Reformat Date* - This option tries to automatically find a date in the name and reformat it to YYYY-MM-DD. A different date format can be specified, according to the Custom Date Format rules. For example, if you enter %a-%b-%m-%Y in the text box, the date will be reformatted accordingly to something like Sun-Apr-01-2024 instead of the default 2024-04-01. You can also help Bulk Rename Utility recognize an unusual date format by using the Custom Date Format rules. For example, if you have a date specified as 01+04+1999, Bulk Rename Utility will not recognize it automatically. In that case, you can use %d+%m+%Y>%d %m %Y in the text box, which means reformat date from %d+%m+%Y to %d %m %Y.

Replace (3)

Replacement. It allows you to replace occurrences of one text string with another.

This section consists of two controls. You must specify the text you wish to *Replace* and the text you wish to replace it *With*. It is possible to find a text string and replace it with an empty string, or with a shorter string, or with a longer string. You may also specify whether or not you want the "find" to be case-sensitive using *Match Case*. Note that the text is always replaced with the text you specify, including any specific text case.

Multiple replacements can be specified using the | separator. For example, *Replace:1|2 With:3|4* will replace 1 with 3 and 2 with 4. *Replace:1|2 With:3* will replace 1 with 3 and 2 also with 3. If you need to use | as a character itself for renaming, then escape it with a \. For example, *Replace:1\|2 With:3* will replace 1|2 with 3, the character | will not be used as separator.

NOTE: Click on ☰ to open the multiple Replacements editor, where you can enter multiple Replace and With pairs and load/save them from/to a file. When loading from a file, each replacement pair must be on its own line, with the *Replace* and *With* values separated by a comma, tab or |.

- *Match Case* - Perform case-sensitive replacement.
- *First* - Replace only first match.

To replace only the *First, Last, Start, End, Second, Third, Fourth, Fifth, Sixth, Seventh, Eighth, Ninth* match, use the *position* \modifier\ , for example:

Replace: \first\a - With: b - This will replace only the first a in the name with b.

The position modifier has to be specified between two \ at the start of the replace string. The following positions are supported:

\first\ , \last\ , \start\ , \end\ : replaces only first match, last match, at start of name, at end of name.
\second\ , \third\ , \fourth\ , \fifth\ , \sixth\ , \seventh\ , \eighth\ , \ninth\ : replaces only nth match.
Example: Replace: \last\text - With: - This will replace only the last 'text' with an empty string, i.e. it will be removed.
Example: Replace: \start\text - With: txt - This will replace 'text' with 'txt' only at start of name.
Example: Replace: \second\text - With: txt - This will replace only the second match of 'text' with 'txt'.

Wildcard Support: wildcard characters * and ? are supported. * matches any number of characters, while ? matches one single character. Use double-quotes " for an exact match start to end.

Example: Replace: Copy (*) - With: C [*] - This will replace the text Copy (*any_number_of_characters*) with C [*any_number_of_characters*]

Example: Replace: Copy (?) - With: Cp [?] - This will replace the text Copy (*one_character*) with Cp [*one_character*]
Example: Replace: Copy (*) - * - This will replace the text Copy (*any_number_of_characters*) with just *any_number_of_characters*
Example with quotes: Replace: "Copy (*)" - * - This will replace the text Copy (*any_number_of_characters*) with just *any_number_of_characters. S*urrounding double-quotes " are used to match the text Copy (*) vs. the entire name, so it will only match if the name starts and ends with Copy (*). If Copy (*) is in the middle of the file name, it will not be matched.

Regex Support: You can also use Regular Expressions (regex) syntax directly in Replace (3) by using the prefix modifier \regex\ in the replace field.
Example: Replace: \regex\(\d{5}) - With: A - This will replace the first 5 digits with an A. The Replace and With boxes are treated as regular expressions.
Example: Replace: \regex\(\d{5}) - With: A$1 - This will replace the first 5 digits with an A plus the matched 5 digits. The Replace and With boxes are treated as regular expressions.
When the \regex\ prefix is used, both the Replace and With fields are treated as regular expressions. In this mode, multiple replacements using the | separator do not apply.
See Regular Expression syntax for more information: Regular Expressions

Note: It is also possible to use the tag *<clip>* in both the *Replace* and *With* fields. <clip> will be substituted with the current text content of the Windows Clipboard.

Note:

- Extended version 2 EXIF Tags are supported in the *With* field. They can be added as text, number or as a formatted date to a file name. Read More.
- All Windows File Properties tags are also supported in the *With* field. They can be added as text, number or as a formatted date to a file name. Read More.
  To enable Windows File Properties tags support, activate the option *'Extract Windows File Properties'* from the Renaming menu.
- The following hash value tags are also supported: <(hash:crc32)>, <(hash:md5)>,<(hash:sha1)>,<(hash:sha256)>,<(hash:sha3)>,<(hash:keccak)> in the *With* field. These tags will be replaced by the corresponding hash value of the file and can be used to add a hash value to the name.
- Other renaming tags can be used too.

Case (4)

Changing of case. Allows the capitalization of file and/or folder names to be changed.

Case drop-down:

- *Same* - leave the original capitalization intact. Then use the *Exceptions* (see below) to upper case or lower case just certain characters or words.
- *Lower* - convert all letters in the name(s) to lower-case.

- *Upper* - converts all letters in the name(s) into capitals.
- *Title* - converts all the words in the selected items to initial caps. A word is generally defined as a string of letters preceded by a space or a bracket or a dash. So "joe public.txt" would become "Joe Public.txt".
- *Sentence* - converts all the words in the sentence to sentence format. This means capitalizing the first letter of the first word after every full stop (.). So "hello EVERYONE. MY name IS FreD.txt" would become "Hello everyone. My name is fred.txt"
- *Title Enhanced* - converts all the words to initial caps. A word is generally defined as a string of letters preceded by a space or a bracket or a dash. So "joe public.txt" would become "Joe Public.txt". For *Title Enhanced,* the words *a, an, and, as, at, but, by, en, for, if, in, nor, of, on, or, per, the, to, v, vs, via* are not capitalized unless they are at the start or at the end of a sentence. More exceptions can be specified using '*Exceptions*', see below for more details.

Exceptions:
You can enter a list of "*exceptions*", separated by colons. So for example if you entered *PHP:ASP:doc* then any occurrence of *php* or *ASP* (or *pHP* or *aSp*) would be converted to upper-case, and every occurrence of *DOC* (or *DoC*) would become *doc*. You can also use spaces to identify words: for example, : book : would always lower-case the text 'book' when preceded and followed by a space.
NOTE: Click on ☰ to open the multiple exceptions editor, where you can enter multiple exceptions easily and load/save them from/to a file. When loading from a file, each exception must be on its own line.

Exceptions for Title Enhanced:
Title Case Enhanced supports additional exception types. You can specify an exception that is a word using the character $. For example using $between will result in the word 'between' not being capitalized unless they are at the start or at the end of a sentence. You can also specify multiple words: $between:$under. You can mix them with non-words exceptions (see above): $between:$under:*PHP:ASP:doc .*

For Title Case Enhanced you can also specify words that you always want to have in upper case, for example, $BOOK:$DISK. They can all be combined: $between:$under:*PHP:ASP:doc:*$BOOK:$DISK .

Other Special Tags in Exceptions:

Exceptions can also contain additional tags that have a special meaning:

- *<clear>* : can only be used with *Title Enhanced.* It clears all the default words that are not to be capitalized unless they are at the start or at the end.
  The default words are: *a, an, and, as, at, but, by, en, for, if, in, nor, of, on, or, per, the, to, v, vs, via.*
- *<ic>* : can only be used with *Title Enhanced.* Ignore all words that are all caps, do not change capitalization for them.
- *<rnlo>* : lower case all Roman Numerals regardless. Applicable to all Case types.
- *<rnup>* : upper case all Roman Numerals regardless. Applicable to all Case types. For example, if you had a file called "Beethoven's niNTH syMPHONY part iii", you might want to use Title Case to format the filename, but this would impact the "iii". Specify <rnup> in the Exceptions to upper case all Roman Numerals.

Exception Examples:
<clear>:<rnup>:$between:$under:*PHP:ASP:doc:*$BOOK:$DISK
PHP:ASP:doc
<ic>
PHP:ASP:doc: book : DISK : iPhone

Remove (5)

Removing parts of a filename.

This section consists of several options for removing parts of a filename. Note that these do not apply to the file extension, just the name.

- *First n* - Remove the first *n* characters from the name. E.g. removing the first 2 characters from "Joe Public.txt" will result in "e Public.txt".
- *Last n* - Remove the last *n* characters from the name. E.g. removing the last 2 characters from "Joe Public.

txt" will result in "Joe Publ.txt".

- *From/to* - Remove a string of text, e.g., from the 6th to the 9th characters. Negative values are also supported, where the position is evaluated from the end of the name. E.g., From -3 To -2 will remove characters from third to last to second to last. -1 identifies the last character in the name.
- *Chars* - Remove occurrences of characters from the name. E.g. typing "QW:#" will result in all occurrences of Q, W, colon and hash being removed.
  A condition can be used for example: *>c* -- remove characters greater than c. *<Z* -- remove characters less than Z. *>c<Z* -- remove characters greater than c and less than Z.
  Use *>ÿ* to remove all characters greater than ÿ, that is all characters past Latin-1 Supplement, such as Emoji.
- *Words* - Remove occurrences of words (separated by spaces).
- *Crop* - Remove any text which occurs before (or after) a specific character or word. See note below.
- *Digits* - Remove all occurrences of the digits 0-9 from the filename.
- *High* - Remove high-ASCII characters (chars from 128 to 255).
- *Trim* - Remove leading and trailing spaces.
- *D/S* - Remove occurrences of double spaces, and replace them with single spaces.
- *Accent* - Remove accented characters and replace them with non-accented versions. File names may contain accented characters, e.g. File names might contain à and á. Bulk Rename Utility provides a facility to replace accented characters with non-accented if needed.
- *Chars* - Remove all characters.
- *Sym* - Remove all symbols !£$%^&*()-_=+[]{};:'@#~,.<>/?\\|`¬¦ from name and it will also remove General Punctuation, Arrows, Miscellaneous Symbols, Dingbats, Miscellaneous Symbols and Pictographs, Emoticons, Transport and Map Symbols Alchemical Symbols, Geometric Shapes Extended, Supplemental Symbols and Pictographs, and Latin-1 Supplement symbols.
- *Lead Dots* - Remove the . and/or .. from the front of filenames (useful if you've copied from a Linux/Unix system).

Note: When you use the "crop" option, you have the ability to specify a "special" value using the wildcard (*). This will remove the specified string, and any characters occupied by the wildcard. So for example, specifying [*] would convert "Hello[ABC] Joe" to just "Hello Joe", as it has removed the two square brackets and everything between. When using Crop with Before or After, the text before or after a specific character or word is removed. To remove everything after a special character inclusive, for example _ , use crop Special and _*. To remove everything before a special character inclusive, for example _ , use crop Special and *_.

Crop Special examples:

"Hello[ABC] Joe" + "[*]" -> "Hello Joe".

"report_2025_final" + "_*" -> "report"

"report_2025_final" + "*_" -> "2025_final"

Crop Before / After examples:

After "_" on "report_2025_final" -> "report_"

Before "_" on "report_2025_final" -> "_2025_final"

Move/Copy (6)

Move a section of text to a different place.

This section allows you to move or copy a fixed number of characters from one part of the string (start or end or range) to another (start, end or position). You could do this with Regular Expressions, but this gives you an alternative. So for example, if all your filenames end with a unique 6-digit sequence code, and you'd rather have this at the start of the filename, then this is the simplest way to achieve it. You can also specify a *separator* in the field *Sep*.
Negative positions are also supported, where the position is evaluated from the end of the name.
The functions available are: Copy first n, Copy last n, Move first n, Move last n, Copy range, Move range.

Add (7)

Add a fixed prefix or suffix to the filename, and certain Exif and ID3 tags.

This section allows you to add a fixed text string to the start of the filename (*Prefix*) and/or a fixed text string to the end of a filename (*Suffix*). Useful for renaming MP3 files, where you can prefix all the filenames with the artist or album name. You can also insert a text string at any point in the filename. Expand the width of this section using the expand button ⊠.
You may also choose to add a "*Word Space*". This option inserts a space before any capital letter (except the first character) that follows a lowercase letter or number. For example, the name "MyHoliday Photographs" would become "My Holiday Photographs".

EXIF Tags: If you are processing JPEG files or other image files that support EXIF, you can also extract and add certain EXIF tags.
These are specified using substitution tags, which you key into the *Prefix*, *Insert* or *Suffix* boxes. The tags supported are:

%a - Aperture
%c - Comments
%e - Exposure
%f - Focal Length
%xb - Exposure Bias
%ma - Camera Make
%mo - Camera Model

Extended EXIF Tags (version 2): Additional extended version 2 EXIF Tags are also supported in *Add (7)*. They can be added as text, number or as a formatted date to a file name. Read More.

ID3 Tags: If you are processing MP3 files, you can also extract certain ID3 tags (n.b. only V1 and V1.1 ID3 tags are supported). These are specified using substitution tags, which you key into the *Prefix*, *Insert* or *Suffix* boxes. The tags supported are:

%r - Artist
%l - Album
%t - Title
%k - Track Number

Other Tags Supported:

- *%z* : You can also specify %z to include the file size (using the format currently specified on the Options menu)
- *<clip>* : It is also possible to use the tag *<clip>*. <clip> will be substituted with the current text content of the *Windows Clipboard*.
- *<removed>* : Use the <removed> tag to add back what was removed in section Remove(5). For example, if in Remove(5) text was removed from position 3 to 6 or First n or Last n or by the crop function, then you can re-add that removed text in a different position, as a suffix, prefix or insertion, using the <removed> tag in Add(7). You can use the <removed> tag multiple times in Add(7).
- See more renaming tags.

HASH Value Tags

The following hash value tags are also supported: <(hash:crc32)>, <(hash:md5)>,<(hash:sha1)>,<(hash:sha256)>,<(hash:sha3)>,<(hash:keccak)>. These tags will be replaced by the corresponding hash value of the file and can be used to add a hash value to the file name.

Windows File Properties:

All Windows File Properties tags are also supported in *Add (7)*. They can be added as text, number or as a formatted date to a file name. Read More
To enable Windows File Properties tags support, activate the option *'Extract Windows File Properties'* from the Renaming menu.

Auto Date (8)

Add various dates to the filename, in various formats.

This section allows you to prefix or suffix the filename with a variety of dates - the date that the file was created, modified, or accessed, and also with the current date. The date can be added in 9 different formats, some of which also include the modification time. You can also specify the characters you would like placed between the date/time and the existing filename (the *Sep.* field) and the character to place between the different date/time segments (the *Seg.* field). e.g. a format of DMY, with a *Seg* field of "-" would give a date of (for example) "25-01-03". The "*Cent.*" flag is used to indicate whether or not you want the year to be represented with two digits or 4 digits. Use the "*Off.*" field to specify an offset, in hours, to apply to the date/ time.

You also have the option to specify your own custom date formats, and these are detailed here.

If you are also updating the file's Creation-date or Modified-date, then you can also choose the "Create (New)" and "Modified (New)" values, and these will equate to your new timestamps.

Several of the dates you can use relate to "date taken". These are EXIF flags which are often embedded in digital camera images (images created using a digital camera). Bulk Rename Utility will only get the flags from JPEG images (.JPG or .JPEG extension), TIFFs (.TIF, .TIFF), Nikon (.NEF) and Canon (.CR2) files. If no date is available then no date will be used. So for example, choosing to append a Date Taken field for a .EXE file will simply append nothing. The dates are:

- *Taken (Original)* - The original timestamp, which should never change.
- *Taken (Digitized)* - The original timestamp, but if the file has been edited (Canon Digital Photo Professional etc.) then this date is updated.
- *Taken (Modified)* - The standard "DateTime" EXIF field, which is usually updated if you edit the picture.
- *Taken (Recent)* - The most "recent" of all three. By recent, this means the last timestamp in the file, not necessarily the most recent in time. This is purely to retain the same behaviour as previous versions of Bulk Rename Utility.
- *Current* - The current date (today).
- *Item Date* - This date comes from Windows Explorer itself (System.ItemDate) and it is the primary date of interest for an item. In the case of photos, for example, this date maps to the date the photo was taken. The Item Date will contain the created date for media, pictures, videos, HEIC files, RAW camera files, etc., taken from the file meta data. Seconds are not supported, i.e. accuracy is down to the minute. This date also supports dates earlier than 1970 unlike other dates in Bulk Rename Utility. This date can be shown as a column in the main file list. To see this date in the file list as a column, enable the option "*Extract Windows File Properties*" from the *Renaming Options* menu.

Append Folder Name (9)

Add the name of the containing folder(s).

This section allows you to append (or prepend) the name of the containing folder to the filename. So for example, if you had a file called "Dancers.jpg" in a folder called "Highland Show August 2003" then you could automatically rename the file to "Highland Show August 2003 - Dancers.jpg". Useful if you have lots of folders, each containing the same group of files, and you want to merge all the files into a single folder.

You can specify a separator character(s) to use between the folder name and the file name.

You can also specify how many folders to append. So for example, if you had a picture called "Cat.jpg" in "C: \Documents and Settings\Administrator\Pictures", and you chose 4 levels, the result will be "C-Documents and Settings-Administrator-Pictures-Cat.jpg" The software will automatically remove the ":\" from the root (e. g C:\) as these two characters are illegal in file names.

If you choose a negative level, then only the folder at that level is appended. If you had a picture called "Cat. jpg" in "C:\Documents and Settings\Administrator\Pictures", and you chose -2 level, the result will be 'Administrator-Cat.jpg' and 'Pictures' is not appended. Level -1 has the same effect as level 1.

Numbering (10)

Add sequential numbers.

This section allows you to sequentially number a group of files. Several criteria exist:

- *Mode* - Specify whether you want the number to be appended to the start of the filename, or at the end of the filename, both or insert at a certain position.
- *Start* - Specify a starting number for the numbering.
- *Incr.* - The number by which you want each file name number to be incremented.
- *Pad* - The minimum number of digits occupied by the numeric element. Bases 1-36 will be padded with leading zeros; the a-z and A-Z options will be padded with "a" or "A" as appropriate.
- *Sep.* - A character or characters that you wish to be inserted between the filename and the number. If you enter the special character ":" (colon) in the Sep. box then this will be replaced with the auto-number. So a separator value of ABC:DEF: would result in ABC1DEF1, ABC2ABC2 etc.

  A common scenario would be:
  - Mode = Suffix
  - Start = 1
  - Increment = 1
  - Pad = 4
  - Separator = -

  This would result in filenames with suffixes of -0001, -0002, -0003, -0004 etc.

- *Break* - Reset the auto-number when the *nnn* character changes. e.g. enter 4 to cause the number to reset when the 4th character of the NEW name changes.
- *Folder* - Reset the auto-number upon a change of subfolder.
- *Type* - You can choose to append the auto-number in any numeric base, from base 2 to base 36. e.g. a value of 26 in base 16 would be appended as 1A. Or even use letters, e.g. A-Z or a-z. Roman Numerals and Hebrew Numerals are also supported.
- *Case* - Specify upper case or lower case numbering. Leave empty to use the default casing.

Extension (11)

Change case of the file name extension.

- *Same* - Leave the original capitalization intact.
- *Lower* - Convert all letters in the extension to lower-case.
- *Upper* - Convert all letters in the extension into capitals.
- *Title* - Convert all the words in the selected items to initial caps, e.g. Jpeg.
- *Fixed* - Replace the extension with a fixed extension. For example, use this option to set all your files to ".doc" types.
- *Extra* - Add a secondary extension. For example, change my.holiday to my.holiday.gif
- *Remove* - Remove any file extension. e.g. *My.Holiday* becomes *My*

Filters (12) (a.k.a Selections)

Select which files and/or folders are shown in the file list. By default this will be all the files and folders in the current folder. Note that this section is only used to govern which files/folders are *displayed*. You still need to *select* a file or folder in order to rename it.

- *Filter* - Specify a file mask to use. Normally this would be *.*, or *.mp3, or *.doc etc. Note that this section is only used to govern which files/folders are *displayed*. You still need to *select* a file or folder in order to rename it. So for example, entering a filter of "*.mp3" means that only your MP3 music files will be shown. You can enter multiple file masks by separating them with a space, e.g. "*.mp3 *.doc *.xls". Note: changes to this field only take effect when you move the cursor to a different field or click a different control. This is to prevent the list from rebuilding with each keystroke as you type. You can also prefix a criteria with the

exclamation mark (!) to perform a "not" expression. So "* !*.doc !*.mp3" will select everything except Word documents and MP3 files. Double quotation marks (") can be used around text containing white spaces, e.g. "file with spaces.*". To use the quotation mark character itself, use \".
NOTE: Click on ☰ to open the multiple masks editor, where you can enter multiple masks easily and load/ save them from/to a file. When loading from a file, each mask entry must be on its own line.

- *Match Case* - Match case when applying the filter.
- *RegEx* - The *Filter* is a regular expression. In this case, the exclamation mark (!) to perform a "not" expression does not apply.
- *Folders* - Include folders in the listing.
- *Files* - Include files in the listing.
- *Subfolders* - If the subfolders option is checked, Bulk Rename Utility will process the contents of all sub-folders in the selected branch of the file hierarchy. Care should be taken when using this feature as the utility can potentially list every file on your system.
- *Lvl* - Recursively scan up to this subfolder level. You can also limit subfolder inclusion using comparison operators: <=, < (e.g., <=3, <3), >, >= (e.g., >0, >=5), and = (e.g., =3). For example, to include all subfolders with files but exclude files in the root folder, use >0. To include only files and subfolders at level 2, use =2. To include all files and subfolders up to level 3, use either 3 or <=3.
- *Hidden* - Checking this option will mean that hidden files will be displayed and selectable. Important: you should be careful using this option as many Windows system files are hidden and renaming them could have a detrimental effect on the functionality of your system.
- *Name / Path Min* - Only files/folders and/or paths whose names are greater than (or equal to) this length will be displayed and selectable. Useful if you want to rename files greater than 64 characters for putting onto a CD, for example. N.B. This will exclude files and folders, if the file/folder name is shorter than the nominated length. However, in a recursive scan, sub-folders will still be scanned regardless of the length of the subfolder's name.
- *Name / Path Max* - Only files/folders whose names are lesser than (or equal to) this length will be processed. However, in a recursive scan, sub-folders will still be scanned regardless of the length of the subfolder's name
- *Condition* - Here you can enter a *Javascript condition* to filter files/folders. If the condition evaluates to 'true' for an object, then that object is included in the file list, otherwise it is not. Using a condition you can include/filter files based on name, date, exif, size, attributes, length, etc.

New Location (13)

This section allows you to leave the original files intact, but create copies of the files (with the new names) in a separate folder. This option will honor the "*Overwrite Target Files*" menu option.

- *Copy Not Move* - Copy the files to the new location instead of moving them. Enabled by default.
- *Keep Str*. - Keep Folder Structure. This option allows you to perform copying or moving operations while preserving the original folder hierarchy of your files. Instead of flattening all files into a single (root) folder during the process, Bulk Rename Utility will recreate the subdirectory structure in the destination. For example, if you have files in nested directories like FolderA\Subfolder1\ and FolderB\Subfolder2\, enabling this option will ensure that after processing, the moved or copied files remain in their respective FolderA\Subfolder1 and FolderB\Subfolder2 paths within the target directory. Benefits include maintaining your original organizational structure and reducing the risk of filename conflicts.

Absolute or relative paths can be used in Copy/Move to Location (13). For example, use .\ for the current folder, ..\ for the parent folder, or simply SubfolderName to copy or move files to a subfolder of the current folder. This allows creating copies of files and folders “in place” or relative to the current file path. Alternatively, you can use an absolute path such as C:\FolderA, in which case all selected files and folders are renamed and copied or moved there.

Special (14)

On top of the above renaming options, there are 5 additional special actions that can be performed on files. These actions are also accessible via the "***Special***" menu and they are:

- Order
- Change File Attributes
- Change File Timestamps
- Character Translations
- Javascript Renaming (Javascript renaming gives you total flexibility and control on file renaming by using Javascript code)

Name Segment

This section allows you to enter a specific segment of the filename that you want all the renaming criteria to apply to.
For example, specifying the Name Segment from 1 to 5 will restrict all the renaming criteria to just the first 5 characters of the filename.
In this way, you can apply the renaming criteria to only a part of the file name, like capitalize only the first 5 characters or replace only the first 5 characters, etc.
Negative positions are also supported, where the position is evaluated from the end of the name, e.g., from -5 to -2.

You can change the order of the name segmentation in the renaming criteria using Special (14) Order

You can also use text and position in the Name Segment.
For example, you can set Name Segment -> From to '-': this will look for a '-' character in the file name and apply the name segment from there.
If you specify Name Segment->From: - and Name Segment->To: -, this will segment the file name from the first '-' to the second '-'.
You can also specify a position for the text, for example Name Segment->From: -,2 (comma separated) means from the second - in the file name.
Using Name Segment->From: (and Name Segment->To:) will segment the file name from the first open parenthesis to the first closed.
Using Name Segment->From: _ and Name Segment->To: _,3 will segment the file name from the first underscore to the third.
Using just Name Segment->To: _,2 will segment the file name from the beginning of the file name to the second underscore.

Renaming From a Text File

Accessible from the '***Actions***' menu, '*Import Rename-Pairs*' allows renaming files from a text file.

Using the Application

Launching the Application

When you run the application for the very first time you will be presented with the main window, located in the center of the screen. The main screen is laid out in a similar way to Windows Explorer. Below the menu bar are two panes, the left of which is a tree view providing easy access to your system drives and folders. The right-hand pane is a list view of the files and folders contained within the currently selected branch of the tree shown over on the left.

Below the navigation panes are a series of controls. The controls are grouped together according to function - for example, all the facilities to remove text from a filename.
➔ As of version 4, you can move the renaming criteria into a separate Window, using Menu -> Display Options -> Detach Renaming Criteria (F12). And you can maximize the file list using Menu -> Display Options -> List -> Maximize File List (F9).

Whenever you re-launch the application most of the settings from your previous session will be retained, such as the directory and file-filter you were using. For this to work you must save your renaming criteria to a Favourites file, using "Save as..."

Only one instance of the application may be launched. If you try to run the application, and there is already a version running, then the original instance will be displayed. It is possible to allow multiple instances in the Program Preferences.

If you chose the *Complete* installation option then the easiest way to access the Bulk Rename Utility dialog is from Windows Explorer itself. Simply highlight the folder or files which you wish to rename, and then either right-click or press shift Function Key 10 and select "Bulk Rename Here" from the context menu. Also, if you've previously saved the renaming criteria to a Favourite file then you can double-click that Favourite file to launch the application.

Renaming Files and Folders

It is easy to rename files and folders:

1. Select the folder which contains the objects you wish to rename. If required, you may also specify a file filter to restrict your list.
2. Enter the renaming criteria. The column beside the file name ('New Name') will show you what the new file name will become.
3. Select the files you wish to process (use CTRL or SHIFT to select multiple files).
4. Hit the *Rename* button. (IMPORTANT: This button is not enabled unless you have selected one or more files/folders to rename)

Note: You can sort the file list using the column headers, as you would do in Windows Explorer.
The files will always be processed in the displayed sequence - useful when you're using the Numbering facility.

The renaming criteria are processed from left to right, according to the number listed against each section

So for example, a text replace will be performed before a change of case. The specific operations are as follows:

1) Apply any fixed name changes from an imported text file

2) Apply any Regular Expression reformatting [RegEx (1)]

3) Remove any file name, or use a fixed name [File (2)]

4) Perform any text substitutions [Repl. (3)]

5) Perform Character Translations [Special (14)]

6) Perform any changes of case [Case (4)]

7) Remove *n* digits from the start, middle or end of the filename, and optionally remove certain characters, and/or all characters, and/or all digits and/or all symbols and/or all high-ascii characters [Remove (5)]

8) Move any text from the start, middle or end of the filename to the start, middle or end of the filename [Move/Copy (6)]

9) Add any prefixes or suffixes, or apply text to the middle of the filename. Suffixes are added at the end of the filename, i.e. before any file extension [Add (7)]

10) Apply any "auto-date" text as a prefix or suffix. "Sep" is the text to insert between the filename and the date; "Seg" is the separator between the day, month, year, hour, minute and second segments. Or you can use a custom date format. [Auto Date (8)]

11) Add the containing folder name as a prefix or a suffix, with a user-defined separator [Append Folder Name (9)]

12) Perform any auto-numbering [Numbering (10)]

13) Make any changes to the case of the file extension, or change the extension [Extension (11)]

14) Process Javascript code [Special (14)]. This is the last step.

The *selected* files will be renamed according to your selection criteria. You can select multiple files by holding down the SHIFT or CTRL keys. If there are any problems with the rename operation then you will be given the option to roll back (undo) the operation.

Remember - you can always preview the new name in the file list. This allows you to refine your criteria before actually renaming the files.

To preserve your renaming criteria for the next time you use the utility, save the renaming criteria to a

Favourite file.

You can use the Reset button to reset all your renaming criteria back to the default values (or the smaller *R* buttons to just reset one group).

Most rename text fields have two small icons on the right:

Chevron (arrow down): Opens a menu of Recent Entries for that field and, where supported, lets you insert Renaming Tags.

Six-dot button: Opens the Zoom window. The Zoom window also shows the field's Recent Entries and allows Renaming Tag insertion (where supported).

Tip: Right-click the six-dot button to go directly to the Insert Renaming Tags dialog.

If needed, you can change the order of the renaming criteria in Special (14).

Changes to File Timestamps and/or to File Attributes are also applied if specified in Special (14).

Actions Menu

The actions menu provides you with a variety of shortcuts to speed up common tasks, and these are described below.

Select All (Ctrl+A)
This will select all the files and folders in the file-list window.

Deselect All (Ctrl+D)
This will de-select all the files and folders in the file-list window.

Invert Selection (Ctrl+I)
Any entry which is currently selected will be de-selected, and vice-versa. Handy for two-pass renames, where you want to rename a whole batch of files in one go, then rename "everything else" in another go.

Select from Clipboard
If you have a list of file names on the clipboard (such as the output from *DIR /B*) then the software will try to select files which match the names from the clipboard.
It will also try to select files that you copied in Windows File Explorer with Ctrl+C.

Jump to Path (Ctrl+J)
This option will display a window, allowing you to type or select a full pathname to navigate to. The path must exist (it will be validated). Network / UNC paths are supported (e.g. \\computer\share\).

Rename Object Manually (F2)
Enable manual name editing of the selected object in the file list.

Refresh Files (F5)
This will refresh the contents of the right-hand file-list window.

Refresh Tree (Ctrl+F5)
This will refresh the contents of the left-hand tree window. Note that you can refresh the contents of the selected branch by collapsing and re-expanding the branch.

Show/Hide Tree (F11)
This will hide or show the folder "tree". This is useful if you have located your folder and you are now doing lots of work in the one folder. The "tree" can be removed to give you more space. Note: The tree will always be visible at program startup, even if it was hidden when you closed the program.

Zoom (F8)
This allows you to launch a new window to edit the current text field. Space on the screen is limited, and some fields are small, so this option allows you to enter long text with ease. You can type text into the new window, in a larger font, and the text will then appear in the field on the main screen. The zoom window will also show the *recent entries* for the current text field. It can be opened also by clicking on the dots ⠿ in a control or with F8.

List

List > Reposition > Move Up, Down, Top, Bottom, Swap, Remove from List

Allows you to reposition/reorganize items in the file list, useful for renumbering.

List -> Apply Random Sort to Current List

This allows you to sort the file list in a random sequence. Useful if you want to create a slideshow in a random display-sequence. Once you've sorted in random order, apply a numeric auto-number prefix to keep the list in that order.

List -> Show Only Items Affected by Renaming Criteria

This feature in Bulk Rename Utility helps users focus specifically on files or folders that will undergo changes based on the current renaming criteria.
When you apply renaming criteria, this feature filters the file list to display only those items whose names will be modified by the current criteria if they are applied. Unaffected items will be hidden, simplifying the view.
Click the "Show Only Items Affected by Renaming Criteria" button on the toolbar or use the corresponding menu command.
If only one item or no items are selected in the file list, the feature will automatically select all items. It will then update the list to display only the items affected by the renaming criteria, hiding unaffected files and folders from view.

If multiple items are already selected, the feature will filter the selection to show only the items affected by the renaming criteria among the currently selected items.

To restore the full file list, use the Refresh button or press F5.

List -> Clear All Items from Current List

This allows you to clear all items from list (items are not deleted from disk, only cleared from Bulk Rename Utility file list)

List -> Clear All Non-Selected Items from Current List

This allows you to clear all items that are not currently selected from the list (items are not deleted from disk, only cleared from Bulk Rename Utility file list)

List -> Auto-Select All Items After Listing a Folder

Let all files and folders be selected automatically when the list of files/folders is built.

List -> Auto-Select All Items After Drag and Drop

Let all files and folders be selected automatically after they are dragged and dropped into the program.

Import Rename-Pairs (CSV)

Import Rename-Pairs from File (CSV)

Import rename pairs from a file to be used for renaming (renaming from a text file, comma-separated values CSV).

Import Rename-Pairs from Clipboard

Import new name-old name rename pairs from the clipboard to be used for renaming.

View Imported Rename-Pairs

If you have imported rename pairs then this option will show the list of pairs imported.

Select Imported Rename-Pairs

If you have imported rename pairs then this option will select the imported rename pairs in the file list (if they are matching).
If the imported rename-pairs include the *full file paths* (e.g. c:\folder\file.ext), then all the files imported are selected in the file list, if they exist.

Clear Imported Rename-Pairs

If you have imported rename pairs then this option will clear the list of pairs.
Normally the list would only be removed by loading a different list or by exiting the application.

Debug New Name

If you have a one (and only one) file or folder selected in the right-hand pane then this menu item becomes available. What this will do is display a simple dialog box which shows how the file's "new name" has been derived. This can help when you are using lots of the controls and boxes, and you aren't getting the end result that you are expecting.

Reset All Renaming Criteria (Ctrl+T)

This allows you to quickly reset all the renaming-criteria which are currently selected. Handy if you want to start from scratch.

Default Renaming Criteria: default.bru Support

Bulk Rename Utility supports a default.bru file for loading default renaming criteria.

When using the "Reset" button or creating a new favourite file with the "New" command, the renaming criteria will now be initialized using the settings from a special file named default.bru, if it exists in the installation folder.

This allows users to define their own default renaming setup that is automatically applied whenever starting fresh, ideal for users who frequently begin with a preferred baseline configuration.

Notes:

The default.bru file must be saved manually by the user using "File > Save" with the desired default settings, and placed in the Bulk Rename Utility installation directory.

If no default.bru file is found, the standard built-in defaults will be used instead (all empty).

This feature enhances flexibility and speeds up workflow by ensuring a consistent starting point for renaming operations.

Reset All Criteria to Last Saved (Ctrl+E)

Reverts all renaming criteria to the last saved status.

Clear All Renaming Criteria (Ctrl+Shift+T)

This command provides a hard reset of the renaming panel, clearing all input fields, selections, and settings across every renaming criteria section.

This function is designed for users who want to start from a completely blank state, with no active renaming rules or filters applied.
This is a hard reset and does not load the default.bru file, even if one is present in the installation folder.
It is the same as the **Reset All Renaming Criteria** command if no *default.bru* is present.

Preview (Ctrl+P)

Preview all the renaming actions, before actually renaming (name change, timestamp change, attribute change, etc). This is an alternative to pressing the PREVIEW button on the screen.

Rename (Ctrl+R)

Actually perform the rename. This is an alternative to pressing the RENAME button on the screen.

Undo Rename (Ctrl+Z)

If you have performed a rename operation and then realized you've made a mistake, this is the option you need. The Undo Rename function allows you to reverse your rename actions, including any timestamp and attribute changes. Files are processed in reverse order, ensuring that even complex renaming operations involving folders and their contents can be reverted successfully.

Bulk Rename Utility now includes multiple-level undo functionality, allowing you to revert several previous rename activities. Additionally, a visible and interactive list of undo actions is provided, letting you select and undo specific operations individually. This enhances your control and flexibility, making it easier to manage and correct your file-renaming tasks.

Create Undo Batch File (Ctrl+B)

Similar to *Undo Rename* above, this will allow you to create a batch file (.bat) which can be run from the command line, and which will reverse out your last renaming operation.

Display Options Menu

This menu contains a variety of options to customize the application appearance.

Always On Top

This will keep the Bulk Rename Utility window on top of all other windows.

List > ...

Highlight Name Changes: Highlight the detailed changes between new name and old name, e.g. additions, removals, etc.

Several options to customize the appearance of the file list: *Show Gridlines*, *Show Icons*, *Show File Sizes as*, *Show Picture Viewer*.

Select Columns: Select which columns to display in the file list.

Autofit All Columns: Automatically set the best width for all columns based on content.

Size All Columns to Width: Automatically set the best width for all columns based on their content and the available width of the file list.

Set Content of Custom Column: Set the content of the custom column, using an expression which can contain Windows File Properties tags, version 2 EXIF tags and Hash tags. For example, you can set the custom column to <(Title)> which would show the Windows File Property 'Title' for each file. For more info on the tags that can be used for the custom column, see Using the Custom Column.

Expand File List (Ctrl+F9): Make the file list bigger by collapsing some renaming controls.

Maximize File List (F9): Make the file list as big as possible by collapsing most renaming controls.

Colours > ...

Several options to customize application highlight colours.

Font > ...

The font size used by Bulk Rename Utility can be selected to be bigger (zoom in) or smaller (zoom out). Select '*Reset*' to reset the font size to its original size. The default system font is used unless the option is de-selected.
Enable the option 'Use Monospaced Font for List and Tree' to make sure that each character occupies the same amount of horizontal space in the file list and tree. This uniformity makes it easier to align text vertically.

Sorting > Logical Sorting

Logical Sorting (also known as natural sorting) in Windows means sorting names in a way that reflects how humans naturally expect numbers to be ordered, even when they’re embedded in text.
For example, with logical sorting, the order is File1.txt, File2.txt, File10.txt, while without logical sorting (i.e. standard lexicographical sorting), the order would be: File1.txt, File10.txt, File2.txt.
Logical sorting treats numeric portions as actual numbers, so it sorts File2 before File10, which feels more intuitive.
Bulk Rename Utility enables logical sorting by default under Display Options -> Sorting, so filenames with numbers are listed in the expected, human-friendly order.
If you want to sort files with standard lexicographical sorting instead, disable this option.

Sorting > Group Affected Files

This option allows you to sort your file listing in the normal way, but with all "affected" files together (files with new name). Therefore, all files with "new names" will be sorted together, and all files which are not affected by your renaming criteria will be sorted together. When you click a column-heading to sort the files, all the files which are affected (normally shown in green) will be grouped together. Note that this does not apply to the Size, Date or Attributes columns. If this option is activated, after having selected some files/ folders click on a column-heading to group the files with "new name" together.

Sorting > Sort Files and Folders Together

With this option selected, files and folders will be sorted together, intermingled according to their name. This differs from the normal Windows Explorer view, where folders are sorted together and then files are sorted together.

Detach Renaming Criteria

This option will show the renaming criteria in a separate detached window using multiple tabs that can be rearranged.

Renaming Options Menu

This menu contains a variety of options to customize the renaming behavior.

Retain Autonumber

If you are using an Autonumber in your renaming then this option will allow the last-used number to be retained. So for example if you start at 1 and rename 5 files, after the rename is complete the "start at" value will show 6. This is useful for renaming in lots of batches but keeping the numbers contiguous.

Rename in Reverse Order (Advanced Option)

Normally files are renamed from top to bottom, e.g. the first item in the list followed the second item etc. But this can cause you problems if files already exist. Let's say you want to rename 1.txt to 2.txt, 2.txt to 3.txt, and 3.txt to 4.txt. Normally the first rename (1.txt to 2.txt) would fail because 2.txt already exists. This option will overcome this, as the files will be processed in reverse order. This option is only required on rare occasions.

Prevent Duplicates

This option allows you to overcome the situation whereby a rename would fail because a file with the same name already exists. If you try to rename a file, and there's already a file with the same name, the software will make a subsequent attempt to rename the file but with a "_1" suffix. If this fails, it will try with "_2" as the suffix, and will continue up to "_99". The limit of 99, and the separator character (underscore, _) are used by default.
You can change the default format that is used to prevent duplicates. To change the format, enable the option *Prevent Duplicates*. A message will be shown that the option is now enabled, and clicking on "*Change Format*" will open a further dialog where you can specify the format string. You can also set the duplicate counter to *Start From 1*, and you can prevent duplicates across folders.

Link Files By Extension

This option allows you to link files by extension, where you specify a Master file extension and one or more Linked file extensions. Once linked, the file names with the master file extensions will dictate how files with the linked file extensions are renamed. See Link Files by Extension.

Advanced Options > Allow Using '\' in Renaming Criteria for Creation of New Folders (Advanced Option)

Use this option with caution. This option can create new folders during renaming and can not be undone. With this option selected, if the new resulting file name contains a \, then a new folder is created, instead of just considering the file name invalid. So for example, if you had a file called a.txt and your renaming-criteria caused the file name a.txt to become 20190801\a.txt, then a new folder 20190801\ is created and the file a.txt is moved in there. This is called '*folderize*'. Using this option, you can reorganize files into folders based on their timestamps, for instance.
Before using this option it is recommended to backup your files

Advanced Options > Overwrite / Delete Existing Files (Advanced Option)

Use this option with caution. This option can delete files and can not be undone. With this option selected, if the destination file name for a rename operation already exists, that file will be overwritten / deleted. So for example, if you had files called a.txt, b.txt and c.txt, and your renaming-criteria caused file a.txt to become c.txt, file c.txt will be overwritten. As such, you need to use this option with extreme caution - if you enter incorrect renaming criteria then you could overwrite valuable existing files. This option is also applicable if you choose to copy files to a New Location (box 13 on the screen).
Before using this option it is recommended to backup your files or copy the files to a new empty location (box 13 on the screen).

Advanced Options > Skip Renaming of File If File Name Already Exists

When this option is enabled, Bulk Rename Utility will automatically skip the renaming of a file if the file name already exists in the same folder.

Advanced Options > Bypass Name and Path Validation Performed by Windows

This option allows you to bypass the standard name and path validation enforced by Windows. It is particularly useful when working with files or folders that contain characters or naming patterns normally considered invalid by the operating system. For example, Windows does not allow filenames to end with a dot (.) or a space ( ), and attempting to rename such files through standard methods will typically fail or silently ignore the change. By enabling this option, the application uses the extended \\?\ path prefix to directly access the file system, allowing operations such as removing or correcting invalid trailing characters, renaming paths with reserved patterns, or working with long paths. Use this feature with care, as bypassing validation may result in files that are difficult to open, delete, or manage through standard Windows tools.

ID3 / Exif Data / File Properties > Extract Exif Data

Use this option to control whether EXIF tags are extracted from photos and shown in the file list’s "Taken (Original)" column. The following file types are processed: .jpg, .jpeg, .jpe, .jfif, .tif, .tiff, .dng, .heic, .heif, .avif. The following camera RAW formats are also supported: .crw, .cr2, .cr3, .nef, .nrw, .arw, .sr2, .srf, .srw, .orf, .rw2, .raf, .pef, .rwl, .3fr, .erf, .kdc, .dcr, .mef, .mos, .mrw, .x3f, .bay. To use EXIF property tags in Bulk Rename Utility, enable this option. This is required only in versions prior to 4.1.0.0. In 4.1.0.0 and later, EXIF property tags work automatically when used without enabling it.

ID3 / Exif Data / File Properties > Extract ID3 Data

Use this option to control whether or not ID3 (Version 1) tags will be extracted from MP3 files. At present only *Artist, Album* and *Title* are extracted, but even this can slow down the processing. If you don't need these fields then leave this option unchecked to speed up the processing.

ID3 / Exif Data / File Properties > Extract Windows File Properties

Use this option to enable showing Windows File Properties in the file list, column "Item Date".

File/Folder Extensions > Rename File Extensions as Being Part of the File Name

With this option selected, the file name extensions will be ignored and your renaming operations will always affect the entire file name. Use this option with great care, as Windows relies on the file name extension to decide what to do with a file.

File/Folder Extensions > Rename Folder Extensions as Being Part of the Folder Name

With this option selected, any "extensions" within folder names will be ignored - this is because folder extensions have no real meaning, unlike file name extensions which identify the file type. So for example, if you have a folder called "Holidays.2004" and you choose to remove the last 2 characters, the resulting folder name will be "Holidays.20" or "Holida.2004", depending upon the status of this option.

Log Renaming Activity to a File (Ctrl+L)

Use this option to record a log of all your renaming activity (including undo requests). By default, the log file will be created in the *Documents* folder with a *.log* file extension. Please note that only renaming activity is logged, not changes to attributes or timestamps.

After enabling this option, using the menu "*Enable Logging to File*", the log file location will be shown, which you can change. You can also select "*View Log File"* from the menu to open and view the log file contents. To change log file location after it has been set, disable the option "*Enable Logging to File*" and then re-enable.

Show Warning Message Before Renaming

This option allows you to control whether or not you are warned prior to renaming files.

Show Confirmation Message After Renaming

This option allows you to control whether or not a confirmation message is shown after renaming is complete.

Confirm Each Renaming Operation Individually

This option allows you to control whether or not each renaming operation is to be confirmed individually with "Yes", "Yes to All", "No", "Cancel".

Special Menu

Change Renaming Criteria Order
Open the Change Renaming Order settings.

Change File Attributes
Open the Change File Attributes settings.

Change File Attributes
Open the Change File Attributes settings.

Change File Timestamps
Open the Change File Timestamps settings.

Change Character Translations
Open the Character Translations settings.

Change Javascript Renaming
Open the Javascript Renaming settings.

Javascript Libraries ...

Include sugar.js : make the library sugar.js available to Bulk Rename Utility Javascript code.
Include date.js : make the library date.js available to Bulk Rename Utility Javascript code.

Context Menu

If you ***right-click*** an item in the right-hand window (e.g. a file or a folder) you will be presented with a Context menu (or "Shell" menu). You will see the standard Windows context menu (with options to Edit, Delete, Copy etc.), with many additional menu options in a new "*Copy to Clipboard*" menu. The "Copy to Clipboard" menu items allow you to copy / export various pieces of information, such as the file name or folder path, to the Windows clipboard. You can also use Ctrl+C to copy ALL the columns for all the selected files to the clipboard, including the 'New Name' column.

Additional menu items:

- *Open Containing Folder* (for the selected file) (Alt+Q).
- *Show List of File Properties* (for the selected file) (Alt+P).
- *Show EXIF info (Photos)* (for the selected file) (Alt+X).
- *Paste from Clipboard*. See Details
- *Reposition* (move item up, down, top, bottom, swap, delete from list):

Action	**Shortcut**	**What it does**
Move Up Selected Item	`Ctrl+Alt+Up`	Moves the selected item up by one position.
Move Down Selected Item	`Ctrl+Alt+Down`	Moves the selected item down by one position.
Move Top Selected Item	`Ctrl+Alt+PgUp`	Sends the selected item to the top of the list.
Move Bottom Selected Item	`Ctrl+Alt+PgDn`	Sends the selected item to the bottom of the list.
Swap Selected Two Items	`Ctrl+Alt+S`	Swaps the positions of exactly two selected items. Select two items, then press the shortcut.
Remove Selected Item(s) from List	`Delete`	Removes the selected item(s) from the working list. Does not delete files from disk.

Note:

1. If you perform an operation on a file/folder using the Shell menu which changes the name of that file, or deletes the file, then this is not automatically reflected in Bulk Rename Utility. This avoids unnecessary rebuilding of the file listing.
2. If you are using the "sub-folders" option, and you select multiple files which span two or more folders, then the right-click context menu might give unpredictable results.

Saving Your Settings

When you quit the application, your current settings (menu choices etc.) will be stored in a .INI file in the same folder as the executable program. As Bulk Rename Utility doesn't require an Installer this makes it quite useful as a utility on a "memory stick" or a "Tools CD".

If you need two sets of preferences with different values then create two copies of the executable with different names, and you'll get two INI files.

Using Favourites

A Favourite is a collection of configuration values which can be used to provide quick access to commonly-used criteria. For example, if you frequently want to rename a bunch of pictures in one folder, and you frequently want to rename a bunch of sound files in another folder, then you could create two Favourites called Pictures and Sound Files. Each Favourite holds with it the selection criteria, rename criteria, and current folder.

When you launch the application for the first time you will not have any Favourites defined, but you can create favourites in the same way as you would create text files or pictures. Store them anywhere on your hard drive and use them as you wish.

Each individual Favourites file contains all the criteria you used to rename files. So for example, you could create a Favourite for renaming your pictures; a Favourite for stripping dodgy characters out of filenames etc. You can then recall these Favourites whenever you need them.

To use favourites, just use the ***Open*** and ***Save/Save As*** menu options in the same way as you would do in Word or Excel and other programs. When you choose to save a favourite, all the settings currently in place will be saved into that favourite. You can then recall these quickly, either from the *Open* command or by double-clicking a Favourite file in Windows Explorer. Favourites files have the file extension .bru.

If you have loaded a Favourite and made changes to it, but wish to revert back to the original version, then choose *Revert all criteria to last saved* in the *Actions* menu.

In the *File Menu -> Favourites,* there are two options related to saving favourites (.bru file):

- **Auto Save On Exit**: This option instructs Bulk Rename Utility to automatically save all latest settings to the currently open favourite file when closing the application.
- **Store Pathname and Refresh File List on Open**: This option instructs Bulk Rename Utility to save and then restore the current path in the favourite file when opening it.

_autoload.bru file: This is a special file that is auto-loaded on 'Drag and Drop' or Bulk Rename Here if this file is present in the same folder as the files. This allows you to have different .bru files automatically when selecting files from File Explorer in different folders. Bulk Rename Utility also navigates automatically to the folder containing the *_autoload.bru* file when an *_autoload.bru* file is loaded.

Favourite Files: Automatic Path Storage with ".path" Suffix

A new feature in Bulk Rename Utility enhances the handling of favourite files (.bru) by allowing automatic storage and restoration of the current path when using favourite files whose names end with ".path".

When a favourite file is saved or loaded and its filename ends with ".path", the current folder path is automatically saved into the favourite file and restored upon loading, regardless of the "Store Pathname and Refresh File List on Open" option setting.

This allows users to associate specific folder paths with their favourite rename configurations simply by naming the file with the ".path" suffix. It provides greater control and convenience when working with multiple folders and saved rename setups.

Example:

Saving a favourite as MyImages.path.bru will store the current directory path. Later, loading MyImages.path.bru will automatically switch Bulk Rename Utility to that same directory, restoring the file list accordingly.

This feature is especially useful for users who frequently work with rename presets across different folders and want to ensure consistency and automation when opening favourites.

Sorting

The columns in the file list have three possible states: **Sort Ascending**, **Sort Descending**, and **No Sort**.

Clicking on a column header cycles through each state.

When **No Sort** is active (i.e. no sort icon is shown in the column header), objects are listed in the order returned by Windows, or in the order they were added to the list.

When dragging and dropping objects into the list, adding them from Windows File Explorer, or pasting them from the clipboard, the objects will be sorted - or not - according to the current column sorting state.

The last sorting state is remembered when the application is restarted.

Random Sorting

Accessible from: Actions > List > Apply Random Sort to Current List

This option shuffles the current file list into a random order. It's particularly useful when you want to create a slideshow or playlist with a randomized display sequence.

After applying a random sort, you can preserve the new order by adding a numeric auto-number prefix using the Numbering (10) section. This ensures that the files stay in the same randomized order when copied or renamed.

Note: The random order is not retained once the list is refreshed, unless you save it using numbering or other renaming criteria.

Other Options

Logical Sorting, Group Affected Files, Sort Files and Folders Together. See Display Options > Sorting

Picture Viewer

Bulk Rename Utility allows you to display a preview window, which will show the picture for the currently selected file. Only certain file-types are supported, these being jpeg (JPG), bitmap (BMP), Icons (ICO), WIndows Metafiles (WMF), Portable Network Graphics (PNG), Scalable Vector Graphics (SVG), Tag Image File Format (TIF) and Compuserve Graphical Interchange Format (GIF). On Windows 10/11, the following image types are also supported: .dng, .heic, .heif, .avif -- plus common camera RAW formats: .crw, .cr2, .cr3, .nef, .nrw, .arw, .sr2, .srf, .srw, .orf, .rw2, .raf, .pef, .rwl, .3fr, .erf, .kdc, .dcr, .mef, .mos, mrw, .x3f, .bay.

In addition, you can only display the picture for a single file; if you select multiple files, then no preview will be displayed.

Click on the picture in the Picture Viewer to resize it.

Bulk Rename Here

When you install Bulk Rename Utility, you have the option to enable the "Context" menu integration.. If you choose this option (recommended), a new **Bulk Rename Here** option will appear when you right-click on a file or folder in Windows File Explorer. Selecting this option will launch Bulk Rename Utility and automatically select the folder or files you right-clicked on (or the parent folder if you right-clicked on a file). This provides quick and convenient access to the utility.

Bulk Rename Utility adds the following commands to the Windows File Explorer context menu::

Bulk Rename Here:
his command can be used on multiple files or a folder. It will open Bulk Rename Utility with the selected files and the parent folder fully scanned. Files might be resorted according to the sorting options within Bulk Rename Utility. If a favourite file (.bru) was recently used, it will be automatically loaded.

Bulk Rename Selected Items:
This command can also be used on multiple files or a folder. It opens Bulk Rename Utility with only the selected files, preserving their order as shown in Windows File Explorer, provided the column sorting status in Bulk Rename Utility is set to No Sort. If a favourite file (.bru) was recently used, it will be loaded automatically.

Note: You can select up to a maximum of fifty thousand items at one time.

If you are using the Favourites feature and have enabled the option to automatically save Favourites upon exit (Auto Save on Exit), please note that the folder name stored in the favourite will be updated to match the folder you right-clicked on.

For performance reasons, when using the *Bulk Rename Here* right-click function, the *Sub-folders* option will be disabled, regardless of its saved setting. This prevents unintentional recursive directory searches, such as when selecting a high-level folder (e.g., C:\). The note regarding Favourites above also applies here.

➲ **Maintaining Item Order**: When selecting multiple items, if you want to preserve their order in Bulk Rename Utility, make sure you right-click on the first item after selecting all desired files, then choose the relevant menu option.

➲ **_autoload.bru file**: This is a special file that is automatically loaded when using 'Drag and Drop' or Bulk Rename Here, provided the file exists in the same folder as the selected files. This allows you to have different .bru files automatically loaded in different folders.

Drag and Drop from Explorer

You can drag files and folders directly from within Windows Explorer. This means you can select files from anywhere on your computer and aggregate them all together, allowing you to rename them in a single operation.

To perform this task, launch Bulk Rename Utility in the normal way. Then launch Windows Explorer (or My Computer), and find the files or folders that you wish to process. Select all the files/folders and drag them over the Bulk Rename Utility "file list" window (you should see the icon change). Then let go of the mouse, and the files will be listed.

As this function is available for ANY Explorer window, you can even use the inbuilt Windows "Find" facility to find files on your computer (for example, all your Word documents) and then drag the files directly from the Search Results window.

It's important to note that when you drag the files/folders across, you are NOT performing any operations on those files. You have not moved those files! You are simply displaying them in the Bulk Rename Utility window for subsequent processing.

➔ Note that any files/folders you drag on the window will be added at the bottom of the list of files/folders already there.
If you instead wish to show ONLY the items you are dragging, then simply hold down the **CTRL key** while dragging.

➔ If you want all dragged items to be automatically selected for renaming, enable the option **List -> Auto-Select All Items After Drag and Drop** in the Actions Menu.

➔ **Maintaining Item Order**: When selecting multiple items, if you want to preserve their order in Bulk Rename Utility, make sure you start dragging from the first item after selecting all the desired files.

➔ **_autoload.bru file**: This is a special file that is automatically loaded when using 'Drag and Drop' or Bulk Rename Here, provided the file exists in the same folder as the selected files. This allows you to have different .bru files automatically loaded in different folders.

Single File Quick Rename

If you want to quickly rename a file or folder manually, without applying any criteria, you can press **F2** or double-click while the filename is selected to edit it.

Simply press the ENTER key without making any changes, to cancel the operation, or press ESC. If you rename a folder and are in *recursive* mode, then the file list will be rebuilt.

During a manual rename, **Tab** and **Shift+Tab** function as they do in Windows File Explorer for consistency:

- **Tab** completes the manual rename and initiates a manual rename operation on the next item in the list.
- **Shift+Tab** completes the manual rename and initiates a manual rename operation on the previous item in the list.

Press **ENTER** to complete and apply the manual rename.

General Program Preferences

The general program preferences are accessible from the Main Menu -> Tools or by pressing Ctrl+F12.

The following preferences are available:

General

Allow multiple instances of Bulk Rename Utility to run at the same time: Yes / No.
By default, only one instance of Bulk Rename Utility can be running.

Auto load the '_autoload.bru' file if present in a directory on drag and drop or on "Bulk Rename Here": Yes / No.
If enabled (Yes), the _autoload.bru file will automatically load when performing a 'Drag and Drop' or using the Bulk Rename Here option, provided the file exists in the same folder as the selected files.
This feature allows you to have different .bru files automatically loaded for different folders, streamlining the renaming process.

Clear all items already in the list on "Bulk Rename Selected Items" from Windows File Explorer: Yes / No.
If enabled (Yes), all existing items in the Bulk Rename Utility list will be cleared when Bulk Rename Selected Items is selected from Windows File Explorer.
By default, this option is set to Yes, ensuring a clean list whenever you use the Bulk Rename Selected Items command. Set to No to add files to the existing list.

Hide the folder tree when starting with "Bulk Rename Selected Items" from Windows File Explorer: Yes / No.
When you select files or folders directly in Windows File Explorer and launch "Bulk Rename Utility" using the context menu option "Bulk Rename Selected Items", the Bulk Rename Utility typically opens showing both the Folder tree (navigation pane showing folders) and the File list (showing the items you've selected).
By enabling this option ("Hide the folder tree when starting with 'Bulk Rename Selected Items' from Windows File Explorer"), the folder tree (navigation pane) will not appear, leaving only the selected items visible in the file list.
This simplifies and declutters the interface when you're renaming specific files you've selected directly, making it easier to focus solely on those selected items without the distraction of the folder navigation pane.

Ask to refresh the folder tree if a new drive is detected: Yes / No.
If enabled (Yes), Bulk Rename Utility will prompt you to refresh and update the folder tree whenever a new drive is detected.
This feature ensures that the folder tree stays up-to-date with any newly added drives.

When the Enter key is pressed while on items in the file list: Enter the rename edit mode / Open the file (s) with default app.
Up to 9 files can be opened in their default app at the same time.

At startup, recall the last used criteria if no favorite file (.bru) is active, instead of starting with blank criteria: Yes / No.
This feature in Bulk Rename Utility provides a convenient way to streamline workflows by determining how renaming criteria are handled when the application starts.
When enabled (Yes), the utility will automatically recall and load the last renaming criteria you used during the previous session. This happens only if no favorite file (a .bru file, which stores predefined criteria) is active.
It allows users to continue where they left off without reconfiguring the criteria from scratch.
When disabled (No), the utility will start with blank renaming criteria, requiring users to manually define new criteria or load a favorite file to proceed.

At startup, navigate to the last path selected if no other path is specified: Yes / No.
When enabled (Yes), Bulk Rename Utility will automatically open the folder you last had selected whenever the program starts, unless you've specified a different path (for example via a command-line argument, drag-and-drop or launching from Windows File Explorer). This lets you pick up right where you left off without having to browse for the same directory each time.

Automatically save the used renaming criteria to the recent entries when renaming is performed: Yes / No.
This option in Bulk Rename Utility determines whether the renaming criteria you use during a renaming operation are automatically added to the Recent Entries list for easy access later. Here's a detailed explanation:
When set to Yes, after performing a renaming operation, the criteria you used (e.g., Replace, Add, Numbering options, etc.) are automatically saved to the Recent Entries list.

These saved entries can be quickly accessed and reused for future renaming tasks, reducing the need to recreate the same criteria manually.
When set to No, the renaming criteria are not automatically saved to the Recent Entries list.

Display

Dark Mode: Same as Windows / Yes / No.
Dark mode support on versions of Windows before Windows 10 is limited.

Apply Windows Themes: Yes / No.
Apply default Windows themes. Yes by default. Can be set to No to disable all Windows themes, like in Bulk Rename Utility version 3.4 or earlier.

Bulk Rename Utility logo on the main window: Show / Hide.

Bulk Rename Utility logo type: Default or Custom (bru.bmp, bru.png or bru.svg)
When a custom image is selected, Bulk Rename Utility will load the logo shown on the main window from the file bru.bmp. If that fails, it will then try bru.png or bru.svg. If that also fails, it will revert to the default image. The best size for the image is *300x100* pixels. The image file has to be in the same folder as the file 'Bulk Rename Utility.exe'

Show links to online AI assistants in the main window: Yes / No / RegEx Only.

When set to Yes, Bulk Rename Utility displays links to online AI assistants (Rename Assistant, RegEx Assistant) in the main window, allowing you to click through and open any tool in your default browser for AI-driven renaming support. When set to No, all AI assistant links are hidden (but still available to be selected from the main menu). Choosing 'RegEx Only' will display only the RegEx Assistant link, keeping the other AI tool links hidden.

Updates

Ask to check for program updates every 30 days: Yes / No.

Message Preferences

Click on "View Message Preferences" to manage which informational or warning messages Bulk Rename Utility displays. You can individually enable or disable messages according to your preference, controlling whether specific prompts or alerts appear during your file-renaming operations. This includes choosing whether particular warnings or notifications will be shown again in the future, or remain hidden to streamline your workflow.

- Warning: "Allow using '\' in Renaming Criteria for Creation of New Folders" option is active; prompt to continue (show/don't show).
- Warning: "Overwrite/Delete Existing Files" option is active; prompt to continue (show/don't show).
- Warning: "Skip renaming if filename already exists" option is active; prompt to continue (show/don't show).
- This will open the Bulk Rename Utility website in your browser, do you want to continue (show/don't show).

Additional Features

Custom Date Formats

You can append a variety of dates to the file names, either at the beginning or the end. For example, you could prefix the file names with their *Modified* date, or you could suffix them with the *Creation Date*. See section 'Auto Date (8)'.
A new feature, introduced in version 2, is the ability to add date and time information in a very flexible manner, using standard formatting codes. The full list of codes is defined below. So for example, if you wanted to suffix the file name with the day name and the month name you could enter a custom code of "%a %b".

The full list of codes is:

Code	Meaning
%a	Abbreviated weekday name
%A	Full Weekday name
%b	Abbreviated month name
%B	Full month name
%d	Day of Month (01-31)
%H	Hour in 24-hour format (00-23)
%I	Hour in 12-hour format (01-12)
%j	Day of Year (01-366)
%m	Month number (01-12)
%M	Minute (00-59)
%p	AM/PM Indicator
%S	Seconds (00-59)
%U	Week number of year (00-53), with Sunday as the first day of the week.
%w	Weekday (0-6), with Sunday=0.
%W	Week number of year (00-53), with Monday as the first day of the week.
%y	Year, with no century indicator (00-99)
%Y	Year, with century indicator (e.g. 2004)
%z	Time zone name
%%	Percentage sign

Capitalization

Use *|U* at the start of the custom date format to upper case the Auto Date (8). Use *|L* to lower case it.
For example, |U%d-%b-%Y would result in 14-JUN-2024 instead of 14-Jun-2024, while |L%d-%b-%Y would result in 14-jun-2024 instead of 14-Jun-2024.

Other Text

You can enter other text in the file name, and the symbols above will be substituted with the values. So, you could enter a custom value of "Created on %a, %d %B, %Y" to get "Created on Tuesday, 25th March, 2004".

Alternate Format Syntax: { ... }

In addition to using the % codes above, you can write date/time formats inside braces { } using JS-style tokens.
Using the alternate format below requires { }.

Example:

Report_{YYYY-MM-DD}_at_{HH-mm-ss} => Report_1971-09-21_at_09-26-00
{YYYYMMDD_HHmmss} => 20250908_170109

Supported tokens (case-sensitive):

Token	Meaning
YYYY	Four-digit year => %Y
YY	Two-digit year (00–99) => %y
MMMM	Full month name => %B
MMM	Abbreviated month name => %b
MM	Month number (01–12) => %m
DDDD	Day of year (001–366) => %j
DD	Day of month (01–31) => %d
dddd	Full weekday name => %A
ddd	Abbreviated weekday name => %a
HH	Hour 00–23 => %H
hh	Hour 01–12 => %I
mm	Minute 00–59 => %M
ss	Second 00–59 => %S
AA	AM/PM indicator => %p
WW	Week number (Mon first) => %W
ww	Week number (Sun first) => %U
ee	Weekday number (0=Sun … 6=Sat) => %w
ZZ	Numeric time zone offset (e.g. +0930) => %z

Escaping braces:

{{ produces a literal { and }} produces a literal }.

Notes:

- Tokens are replaced only inside braces; outside text is kept as-is.
- Unknown or unsupported tokens inside braces are left unchanged.
- Capitalization prefixes (|U/|L) still apply to the final rendered date text.
- To include a literal percent sign in the final output, use %% (the standard escape) after conversion.

Recursive Scans

It is possible to perform a recursive scan and rename from the current folder. This allows you to rename folders and files contained within any subdirectories from the current folder. Subdirectories of subdirectories are also scanned, right down to the lowest level. To do this, enable the Subfolders option of Selection (12).

This option needs to be treated with great care - if you scan a high-level folder such as C:\ or C:\Program Files then the program could have tens of thousands of files to scan. Whilst the system should cope with in excess of 250,000 files, it will take a long time for the file list to be displayed.

As such, it is recommended that you only use the Subfolders Option of Selection (12) only if you really need it.

Please note that, if you rename a folder whilst using the recursive option, after the rename is complete the file list will be refreshed automatically. This is to prevent problems whereby the file-list becomes out of synchronization with the files on your hard drive. If you don't rename a folder (just files), or if you rename anything whilst not using the recursive option, then no refresh is necessary.

Renaming From A Text File (CSV)

With Bulk Rename Utility it is possible to rename files according to a text file. The text file should contain a list of "old name" and "new name" pairs, separated by a pipe symbol (|) or a comma (,). Bulk Rename Utility will read ANSI and Unicode (UTF-16) text files. Starting from version 3.5, Bulk Rename Utility can also read UTF-8 encoded files, e.g. CSV files from Excel.

To use this facility, create a text file in for the correct format, for example:

```
Track001.mp3|Headlong.mp3
Track002.mp3|Rushes.mp3
TRACK003.mp3|AnywhereIs.mp3
```

or use comma separated (CSV)

```
Track001.mp3,Headlong.mp3
Track002.mp3,Rushes.mp3
TRACK003.mp3,AnywhereIs.mp3
```

One entry per line.

Next, click the *Import Rename-Pairs* on the *Actions* menu and select your file. If the file is valid, then select all the files to rename and you will see the "new name" column reflecting the contents of the text file.

When comparing the current filename with the text file, the application ignores the *case* of the filename, e.g. it will match Track001.mp3 and TRACK001.mp3. This makes it easy to rename the files.

Click on "Rename Pairs Imported" to show the list of imported renaming pairs.

Specifying Full Paths

You can also specify *full paths*, such as c:\folder\track1.mp3, in the first column for the old file names.
Then Bulk Rename Utility will match these names against the full path and not just the file name.
Partial / relative paths are not supported.

Example:

```
c:\folder1\Track001.mp3|Headlong.mp3
c:\folder2\Track002.mp3|Rushes.mp3
c:\folder3\Subfolder\TRACK003.mp3|AnywhereIs.mp3
```

or use comma separated (CSV)

```
c:\folder1\Track001.mp3,Headlong.mp3
c:\folder2\Track002.mp3,Rushes.mp3
c:\folder3\Subfolder\TRACK003.mp3,AnywhereIs.mp3
```

Text File Requirements

Bulk Rename Utility will read ANSI and Unicode (UTF-16) text files. Starting from version 3.5, Bulk Rename Utility can also read UTF-8 encoded files, e.g. CSV files from Excel.
Only files with CR LF (Windows) or LF (Unix) line endings are supported.

Names with Comma

If a name contains a comma, it must be enclosed in double quotes. This quoting prevents the comma from being interpreted as a field separator, for example:
"Track,001.mp3","Bob,1.mp3"

Extensions

The full file name + extension is matched, not just the file name.

Note: You can **view** all the imported rename-pairs by selecting '*View Imported Rename-Pairs*' from the Bulk Rename Utility menu or by clicking on "*Rename Pairs Imported*" on the main window.

If you hover the mouse pointer on "*Rename Pairs Imported*", Bulk Rename Utility will show the current imported file path as a tool tip.

Select Imported Rename-Pairs

Once you have imported the rename pairs, then use Actions > Import Rename-Pairs (CSV) -> Select Imported Rename-Pairs to select the imported rename pairs in the file list (if they are matching).
If the imported rename-pairs include the *full file paths* (e.g. c:\folder\file.ext), then all the files imported are selected in the file list, if they exist.

Renaming Using The Clipboard

With Bulk Rename Utility, it is possible to rename multiple files according to the content of the Windows Clipboard.

There are two separate ways to do this:

Method 1
Right-click on objects in the *file list* and select "**Paste from Clipboard** > **Set New Names from Clipboard**" from the menu (shortcut **Alt+N**).

Bulk Rename Utility expects the clipboard to contain a list of file names, such as:

FileName1
FileName2
FileName3
FileName4

with each name is on a separate line.

Bulk Rename Utility will use this list to apply the new file name to the current selection.
FileName1 will be applied to the first selected element, and so on.
If more elements are selected than there are items in the clipboard, the names are repeated.
The clipboard content can be copied from a spreadsheet, a Word document, generic text, or from Windows File Explorer using Ctrl+C.

Method 2
From the main menu, select the Action menu, then choose *Import Rename-Pairs (CSV)*, and finally **Import Rename-Pairs from Clipboard** (shortcut **Ctrl+Alt+C**).

Bulk Rename Utility expects the clipboard to contain a list of old name and new name pairs:

FileNameOld1,FileNameNew1
FileNameOld2,FileNameNew2
FileNameOld3,FileNameNew3
FileNameOld4,FileNameNew4

where each old name-new name pair is on a separate line.

OR with full file paths:

C:\folder1\FileNameOld1,FileNameNew1
C:\folder2\FileNameOld2,FileNameNew2
G:\folder\FileNameOld3,FileNameNew3
C:\folder1\subfolder\FileNameOld4,FileNameNew4

The format is the same as for Renaming From A Text File (CSV); both commas and pipes (|) are supported as separators between the old name and new name.
Also, a tab is supported as a separator, so you can select two columns in an Excel spreadsheet with old

name and new name, for example, and copy them into the clipboard with Ctrl+C, and then import that information into Bulk Rename Utility using Import Rename-Pairs from Clipboard (Ctrl+Alt+C).

The content of the clipboard will be imported, and you can then select the files in Bulk Rename Utility to rename them according to the imported list, in the same manner as Renaming From A Text File (CSV).

Method 3

Right-click on the file list and select "**Paste from Clipboard** > **Paste Items from Clipboard**" (shortcut Ctrl+V) or "**Paste from Clipboard** > **Clear and Paste Items from Clipboard**" (shortcut Shift+Ctrl+V) from the context menu.
This will paste files intended for renaming from the clipboard, provided the file paths exist. File paths can be copied from Windows File Explorer or from lists in Word, Excel, or other text editors. Make sure each file pasted into Bulk Rename Utility includes the full path, for example:

```
C:\folder1\FileNameOld1.ext
C:\folder2\FileNameOld2.jpg
G:\folder\FileNameOld3.doc
C:\folder1\subfolder\FileNameOld4.xls
```

Once files are pasted into the file list, you can apply your renaming criteria as usual.

Regular Expressions

Bulk Rename Utility supports powerful Regular Expression processing. This allows you to enter a flexible Match expression, and a flexible Replacement expression, and the program will generate the appropriate name using these expressions. For example, you can use match and replace strings to swap two words in a filename, or remove numbers, or apply fixed formatting.

A full description of Regular Expressions is beyond the scope of this help file, but a wide range of resources is available on the internet.
However, the syntax supported by Bulk Rename Utility is the same as that offered by PERL 5. The precise implementation is via the PCRE Library, and full notes on the Perl Regular Expression syntax can be found here.
We also have a section on our forum dedicated to Renaming Files using Regular Expressions.

Note: The default Regular Expressions supported by Bulk Rename Utility are based on PCRE not on the newer PCRE2 library. To support version 2 of the Regular Expressions, enable the option '**v2**'. This will enable support for Regular Expressions Version 2, which is based on the Boost RegEx and fully supports Perl Regular Expression to the latest version, ECMAScript and JavaScript.
There is also the option 'Simple' available to support a simple basic format for file name replacements, please see below for more info.

A quick summary of the syntax is:

Character	**Usage**
*	Matches the previous character zero or more times
+	Matches the previous character one or more times
?	Matches the previous character zero or one times
.	Matches any single character except the newline
^	Matches the start of the input
$	Matches the end of the input
x|y	Matches either first or second character listed
(pattern)	Matches pattern
{number}	Matches exactly number times
{number,}	Matches number, or more, times (note comma)
{num1, num2}	Matches at least num1 and at most num2 times
[abc]	Matches any character listed between the [ ]
[^abc]	Matches all characters except those listed between the [ ]
[a-e]	Matches any characters in the specified range (a,b,c,d,e)
[^K-Q]	Matches all characters except in the specified range
\	Signifies that the next character is special or a literal.
\b	Matches only on a word boundary
\B	Matches only inside a word
\f	Matches only on a form feed character
\n	Matches only on a new line
\r	Matches only on a carriage return

\s	Matches only on a blank space
\S	Matches only on nonblank spaces
\t	Matches only on a tab
\d	Matches any digit

Replacements are usually performed on the basis of "components, and these are defined using \ notation, e.g. \1 matches the first element, \2 matches the second

Example Regular Expression:

Match:	(Louis Armstrong)(.\[0-9].)([A-Za-z]*)
Replace:	\1 \3

EXAMPLE:

Assume you have a file called Program Files, and you wish to swap the names around (e.g. Files Program). A Regular Expression which performs this task is :

^([A-Z][a-z]*) ([A-Z][a-z]*)

Let us break this down into components:

^ This means start at the beginning of the string

([A-Z][a-z]*) This is a single "group", which we will use later. What this says is that we want any capital letter, followed by any number of lower-case letters. The single capital letter is denoted by the **[A-Z]**, i.e. allow a letter in the range A to Z. The lower-case letters are denoted by **[a-z]** in the same way, followed by an asterisk. This means we can allow any number of letters.

We then allow a single space. If I had wanted multiple spaces I would probably have typed "space asterisk", or possibly (*) to group.

We then have exactly the same again, i.e. we are denoting two words.

Notice we had two sets of brackets. Everything within each set of brackets is treated as a "grouping", and we refer to these groupings as **\1**, **\2**, **\3** etc.

So, lets say we wanted to swap around the two words in the filename. We would put:

^([A-Z][a-z]*) ([A-Z][a-z]*)

For the match string, and

\2 \1

As the replacement string. Of course, we're free to manipulate the replacements string as we like. For example, it would be quite valid to have:

The \2 which are used to run the \1

For the replacement string. This would result in:

The *Files* which are used to run the *Program*.

The above example is very precise. If we wanted to swap the first two words of a name, but keep the remaining text the same, we could put

^([A-Z][a-z]*) ([A-Z][a-z]*)(.*)

\2\1\3

This says to create three groups: the first group is the first word, the second group is the second word, and the third group is everything that's left.

REGULAR EXPRESSIONS (VERSION 2)

Bulk Rename Utility 3.4 introduces Regular Expressions version 2, which can be enabled by switching on the option **v2** in group RegEx (1).

Version 2 Regular Expressions support additional syntax and are fully compliant with the latest Perl Regular Expressions.

v2 Regular Expressions are based on the Boost library syntax and they support Modifiers, Branch Reset, Lookahead, Lookbehind, Conditional Expressions and more.
v2 replace syntax is documented here Replace Format String Syntax.

Two Regular Expression modifiers are also supported under version 2:

- **/i** makes the regular expression case-insensitive
- **/g** globally matches the pattern repeatedly in the string and does not stop at the first match.

For instance:

Match : (\w)/g

Replace: \L$1

will replace ALL upper-case to lower-case with v2 Regular Expressions.

Note that in v2 the unmatched text is copied to the output, unlike in the default regular expressions.

SIMPLE REGULAR EXPRESSIONS

Bulk Rename Utility 3.4 introduces Simple Regular Expressions, which can be enabled by switching on the option **Simple** in group RegEx (1).

Simple regular expressions work by matching text and then removing or rearranging the matched text.

The syntax is very simple, up to 9 matching tags can be used %1, %2, %3 until %9, which match a string of text. Examples:

Match: %1-%2

Replace: %2-%1

It will match first a string of text with %1 then the character '-' and then another string of text. It will then replace it with the second string of matched text, the character '-' and the first string of text matched. Effectively it will switch text around the character '-'.

Match: %1(%2)

Replace: %2

It will match first a string of text with %1 then a '(' and then another string of text followed by ')'. It will then replace it with the second string of matched text and the rest is dropped.

Match: %1-%2-%3 %4

Replace: %3-%2-%1 %4

It will match and reorder text such as '2020-03-12 Boston' into '12-03-2020 Boston'.

Match: %1%2%3%4

Replace: ABC%4

Replaces the first 3 characters with 'ABC'.

Match: %1_%2_%3

Replace: %1_%2

This will remove everything (%3) after the second underscore (_).

Match: %1_%2_%3

Replace: %2_%3

This will remove everything (%1) before the first underscore (_).

Up to 9 matching tags can be used : %1, %2, %3 until %9.

There are also 5 additional modifiers that can be used in the Replace string when **Simple** is used. These 5 modifiers are the same modifiers as when **Simple** is not used:

\U : Causes all subsequent characters to be output in upper case, until a \E is found.
\L: Causes all subsequent characters to be output in lower case, until a \E is found.
\E : Terminates a \L or \U sequence.
\l : Causes the next character to be outputted, to be output in lower case.
\u : Causes the next character to be outputted, to be output in upper case.

Examples:

Match: %1

Replace: \U%1

Capitalizes the name because %1 matches the whole name and \U%1 capitalizes it.

Match: %1_%2

Replace: \U%1\E_%2

Capitalizes the part of the name before _.

Match: %1_%2

Replace: \U%1\E_\L%2

Capitalizes the part of the name before _ and lowercases the part of the name after _.

Match: %1%2%3%4

Replace: \L%1%2%3\E%4

Lowercases the first 3 characters of a name.

SPECIFYING MULTIPLE REGULAR EXPRESSIONS USING THE (?X) SEPARATOR

Bulk Rename Utility 3.4 introduces the ability to specify multiple Regular Expressions in group RegEx (1).
Only v2 Regular Expressions and Simple Regular Expressions do support multiple regular expressions.
Multiple Regular Expressions can be specified by separating them with the separator (?X).

NOTE: Click on ☰ to open the multiple RegEx editor, where you can enter multiple Match and Replace pairs.

For example:
Match: (S)(?X)(P)
Replace: A(?X)R

will run two Regular Expression match and replace, first:
Match: (S)
Replace: A

and then:
Match: (P)
Replace: R

If only one expression is used in the Replace field, then it is used for both matches:
Match: (S)(?X)(P)
Replace: A

will run two matches, first:
Match: (S)
Replace: A

and then:
Match: (P)
Replace: A

If you need to use (?X) in the regular expression itself and not as a separator, you can escape it with \

Match: (S)\(?X)(P)
Replace: A

will run only one match:
Match: (S)\(?X)(P)
Replace: A

Change Renaming Order

You can change the order in which the renaming criteria are processed instead of the default from 1 to 14. This will allow you to add a counter before processing a regular expression, for example, or add text before replacing.
Click on **Order** in *Special (14)* to change the default order.

The new settings will also be stored in the .bru file if you are saving your renaming criteria to a file.

The position of the Name Segment action can also be changed in the order.

Changing File Attributes

Bulk Rename Utility allows you to change the attributes of files and folders once they have been renamed. This can be useful if you wish to flag all renamed files/folders as requiring archiving, or if you want to hide all the renamed files/folders. As well as making a file read-only, hidden etc., you can also clear the same flags.

For each of the file attributes you can choose to **set** the attribute, **clear** the attribute, or **leave** it unchanged.

If you are updating the file's attributes information, this will be reflected in the application main window, as the attribute field will be colored in the file list.

Note

If you set the *Hidden* or *System* attribute for a file, it will disappear from the main window upon renaming if you do not have *Hidden* or *System* options ticked in the Filters group (group 12).

Changing File Timestamps

Bulk Rename Utility allows you to change the timestamp information for files and folders once you have renamed them. This is useful if you have certain applications which use a file's timestamp in order to identify if a file has been modified, of if you want to sort photographs in a particular sequence. Some older applications even use the timestamp to identify a product version.

You can choose a new date and timestamp for the *Created Date, Modified Date* and *Accessed Date*, and they are independent of each other. You can also choose a value of "Current" which will use the current date and time at the time of renaming.

You can also apply a ***Delta***. This is a value in days, followed by HH:MM:SS, and this value will be added (or subtracted) from the timestamp. So for example, if you rely upon timestamps and you realize your computer's clock is wrong, you can use this facility to batch-correct the timestamps. n.b. This will NOT update the "date taken" information held within a picture file.

The option "***Incr. By***" increments (or decrements, if negative) the timestamp of each file by a fixed number of seconds from the previous file. For example, with a value of +10 seconds:

File 1: original timestamp, File 2: +10s, File 3: +20s, File 4: +30s, ...and so on. Useful for creating sequential timestamps to maintain chronological order.

If you are updating the file's timestamp information, this will be reflected in the application main window, as the appropriate date field will be colored in the file list.

If you would like to set the Modified and Accessed timestamps to be the same as the Created timestamp then choose the *Created* option.

Note that the Date Taken (Original) is the *Date Time Original* EXIF flag. If that information is not available, for example for video files or some image formats, then the file 'Item Date' is used instead from the Windows File Properties.

You can also use the option *Exif Date Taken (Original) / Item Date set* to change the value of the Exif Date Taken to the file Created Timestamp or Modified Timestamp.

Character Translations

Character Translations allows you to enter a specific character or sequence of characters, and have that translated into a different character or sequence of characters. So for example, you could specify that you always want a $ sign to be converted into the word DOLLAR.

There are three ways to enter the replacement data:

1. As a character, e.g. A
2. As a hex value, e.g. 0F
3. As a decimal value, e.g. 065

Separate the FROM and the TO conversions with an equals sign (=). If you wish to actually convert an equals sign to/from something else then you can specify the hex or decimal value for the equals sign in your rules.

Bulk Rename Utility identifies the type of value entered by its length. So if your value is one character long then it's a direct character; two characters long and its a hex value; three characters long and its a decimal value.

In the following examples, every example is converting a capital "A" to a capital "B"

- A=B (direct expression of the characters to convert)
- 41=42 (two character long, therefore hex values)
- 065=066 (three characters long, therefore decimal values)
- A=066 (using a mixture of the above)
- 41=066 (using a mixture of the above)

If you wish to to convert several characters then you can separate the values by commas. So the following example will convert ABC to DEF:

- 41,066,C=D,E,070

If you wish to actually convert a comma sign to/from something else then you can specify the hex or decimal value for the comma sign in your rules.

Learn more on our website : **Character Translations HowTo**

Renaming Tags

Bulk Rename Utility employs a system of dynamic tokens referred to as **tags** that act like *variables*. These tokens let you insert file properties or external data into the new filenames automatically. Unlike static text that you type into the renaming fields, these tokens are **dynamically** replaced with corresponding values when the renaming operation is executed.

In Bulk Rename Utility, the use of angle brackets (**<** and **>**) around tags is a strict part of its syntax. For example <size>, <folder>, <clipboard>, etc. This notation is essential to differentiate between static text (literal characters in your new file names) and dynamic tokens (the "variables" that pull file properties or external information).

Here’s an in-depth look at how these tokens work.

Dynamic Replacement: Think of these tokens as variables in programming. When you include a token like <size> in your file naming rule, Bulk Rename Utility doesn’t treat it as literal text; instead, it computes the file’s size and inserts that value in its place during the renaming process.

Automation: This approach automates the renaming task by incorporating specific file attributes or external data (like clipboard content for example) into the filename.

Usage Scope: Tokens can be combined with fixed text. For example, if you input a pattern such as "Image_<size>_edited", each file will be renamed to reflect its own size embedded within the filename.

Formatting: Tokens (=renaming tags) can be formatted using formatters. For example, <name{upper}> will convert to the current name into upper-case. Or <($system.itemdate){YYYYDDMM}> will expand to the item date in format YYYYDDMM. See renaming tag formatting specifications.

Renaming Tags are supported in the following fields:

✓ **RegEx (1)** > Replace

✓ **Name (2)** > Fixed

✓ **Replace (3)** > With

✓ **Add (7)** > Prefix

✓ **Add (7)** > Insert

✓ **Add (7)** > Suffix

List of Renaming Tags

- Windows File Properties Tags - Examples: <(System.Photo.CameraManufacturer)>, <(Title)>, <(subject)>. To enable Windows File Properties tags support, activate the option 'Extract Windows File Properties' from the Renaming menu. This is required only in versions before 4.1.0.0. In 4.1.0.0 and later, Windows File Properties tags will work automatically when used, without enabling this option.

- EXIF image Properties Tags - Examples: <(exif:ImageWidth)>, <(Title)>, <(subject)>. <($exif: DateTimeOriginal)>. To enable EXIF Properties tags support, activate the option 'Extract EXIF data' from the Renaming menu. This is required only in versions before 4.1.0.0. In 4.1.0.0 and later, EXIF property tags work automatically without enabling this option.

- PDF and Video Tags - Examples: <(PDF:Title)>, <(PDF:Author)>, <(MediaInfo:FrameRate)>. <(MediaInfo:Duration/String)>. To enable these tags you must install two optional components: the PDF DLL and the MediaInfo DLL.

- **Other Tags**:

<clip>	A special tag which is replaced by the current text content of the Windows Clipboard.
<removed>	Use the <removed> tag to restore text that you removed with the Remove(5) function. For example, if you removed characters from positions 3 to 6, or you removed the first or last few characters, or used the crop function to remove part of the filename, you can later insert that same removed text elsewhere in the filename. You do this by placing the <removed> tag in one of the Add(7) fields, whether as a prefix (at the beginning), as a suffix (at the end), or as an insertion (in the middle). You can use the <removed> tag multiple times if you want the removed text to appear in several parts of the filename.
<name>	The current file name (without extension).
<ext>	The current file extension (without .)
<counter>	A counter that is incremented during the renaming operation, starting from 1.
<size>	The size of the file in bytes with no formatting (number).
<folder>	The containing folder name.
<numfiles>	The number of files in a folder. When applied to a file, this returns the number of files in its containing folder. Hidden and system files are not counted.
<ulid>	A ULID (Universally Unique Lexicographically Sortable Identifier) is a 26-character string that combines: • A timestamp (so you know roughly when it was generated) • Random data (so no two ULIDs collide, even if created at the exact same millisecond) Because of its design, ULIDs are: • Globally unique – you can generate one anywhere (on your computer, a server, a phone) and be confident it won’t clash with anyone else’s. • Sortable by creation time – when you list ULIDs alphabetically, they also go in the order they were created. Example: 01F4Z5X9A0B6C7D8E9F2G3H4J5
<uuid>	A UUID (Universally Unique Identifier) is a 128-bit value, typically represented as a 36-character string (32 hexadecimal digits plus 4 hyphens). For example: 550e8400-e29b-41d4-a716-446655440000 UUIDs come in several “versions” (most commonly v4, which is used by Bulk Rename Utility), but the gist is that each one is effectively guaranteed to be unique across time and space.

<(hash:crc32)> <(hash:md5)> <(hash:sha1)> <(hash:sha256)> <(hash:sha3)> <(hash:keccak)>	These tags are replaced by the corresponding hash value of the file and can be used to add a hash value to the file name.
<randnum>	Generates a random numeric string composed solely of digits (0-9). Formats: a) <randnum> - No additional parameters. - Default length: 6 digits. b) <randnum:X> - X is a positive integer indicating the number of digits. - The number of digits is capped at 1000. - Example: <randnum:8> produces an 8-digit numeric string. Usage Example: With <randnum>, you might get a result such as "093847". With <randnum:4>, a possible output would be "5483"
<randchar>	Generates a random alphabetic string containing only letters (a-z, A-Z). Formats: a) <randchar> - Default length: 6 letters. b) <randchar:X> - X is a positive integer that specifies the length of the generated string. - The string is formed from 52 letters (26 lowercase and 26 uppercase). - Example: <randchar:10> produces a 10-letter string. Usage Example: With <randchar>, the tag may generate "AbCdEf". With <randchar:3>, a possible output is "xyz" or "Qwe".
<rand>	Generates a random alphanumeric string containing digits (0-9), lowercase letters (a-z), and uppercase letters (A-Z). Formats: a) <rand> - Default length: 6 characters. b) <rand:X> - X is a positive integer that specifies the exact number of characters. - Valid lengths are from 1 to 1000.

	- Example: <rand:12> produces a 12-character alphanumeric string. Usage Example: With <rand>, the output might be "aB3dE9". With <rand:8>, you might see "9ZxY12Ab".
<randname>	Generates a random name string composed of valid characters from the 8-bit range (32 to 255). This function excludes characters that are forbidden in Windows file names (such as <, >, :, ", /, \, |, ?, *) and also filters out unprintable characters. Formats: a) <randname> - Default length: 10 characters. b) <randname:X> - X is a positive integer between 1 and 255. - Example: <randname:2> produces a 2-character string. c) <randname:X-Y> - X and Y are positive integers (with 1 <= X <= Y <= 255). - The function will generate a name whose length is randomly chosen between X and Y, inclusive. - Example: <randname:4-20> will produce a name with a length between 4 and 20 characters. Usage Example: Using <randname> might produce "h93K*gh@2Q". With <randname:8>, you might see "L7k#tP3s". With <randname:3-6>, the result may be a string of random length (for example, "jT9b" or "P3kL8").

Renaming Tags - Formatting

Bulk Rename Utility employs a system of dynamic tokens referred to as tags, which act like variables. These tokens let you insert file properties or external data into the new filenames automatically.
Unlike static text that you type into the renaming fields, these tokens are **dynamically** replaced with corresponding values when the renaming operation is executed.

In Bulk Rename Utility, the use of angle brackets (< and >) around tags is a strict part of its syntax. For example, <size>, <folder>, <clipboard>, etc. This notation is essential to differentiate between static text (literal characters in your new filenames) and dynamic tokens (the "variables" that pull file properties or external information).

Renaming tags can be formatted using "formatters". For example, <name{upper}> will convert the current name to uppercase. Or <($System.ItemDate){YYYYDDMM}> will expand to the item date in the format YYYYDDMM.

Here is a detailed look at the formatting syntax.

Formatters are small instructions that you add inside a tag to shape the value (text, number, or date) before it is inserted into the filename. You place formatters after the tag, inside curly braces.

Example: **<name{upper}>** makes the tag 'name' UPPERCASE - could be used in Name (2) -> Fixed.
Example: **<(PDF:Creator){lower}>** makes the tag name LOWERCASE.
Example: **<(System.Title){substr:1:10}>** extracts the first 10 characters of the document title.

You can chain several formatters; they run from left to right. This sequence is called a *pipeline*. For example: `<name{slice:before:-}{upper}>` - This focuses on the part of the name before the first hyphen and uppercases only that part.

Quick Syntax Rules

✓ Arguments are separated by : (colon). If you need a literal colon, wrap the argument in quotes: `{set:'Part:01'}`.
✓ Some formatters accept a case-insensitive flag: `ci` or `i`.
✓ Occurrences can be numbers (`1, 2, 3`) or words (`first, second, third, last`).
✓ “Slices” let you temporarily focus on a portion of the text; everything after the slice works on that portion only, and then it is spliced back automatically.
✓ When a tag value is a date, a bare pattern like `{YYYYMMDD}` is treated like `{date:YYYYMMDD}`.

A. Text Case and Cleanup

Formatter	What it does	Syntax	Example => Result
upper	UPPERCASE letters	`{upper}`	"Chapter one" => "CHAPTER ONE"
lower	lowercase letters	`{lower}`	"My FILE" => "my file"
titlecase	Title Case (smart)	`{titlecase}`	"an_example of NASA" => "An Example of NASA"

trim	Trim spaces at both ends	`{trim}`	" hello " => "hello"
default	If empty, use fallback text	`{default:Untitled}`	"" => "Untitled"

Examples:

`<(PDF:Creator){upper}>` => UPPERCASE the creator (e.g., "Adobe InDesign" => "ADOBE INDESIGN").

`<(PDF:Creator){lower}>` => lowercase the creator (e.g., "Microsoft Word" => "microsoft word").

`<(System.Title){titlecase}>` => Title Case the title (e.g., "introduction to pdf/a" => "Introduction To PDF/A").

`<(System.Title){trim}>` => remove leading/trailing spaces (e.g., " Project Plan " => "Project Plan").

`<(System.Title){default:Untitled}>` => if empty, use fallback (e.g., "" => "Untitled").

`<(System.Title){trim}{titlecase}>` => trim then Title Case (e.g., " draft – marketing overview " => "Draft – Marketing Overview"

B. Cutting by Position

Formatter	Purpose	Syntax	Example
substr	Take part by start and optional length (negative start counts from end)	`{substr:start[:len]}`	On "abcdef": {substr:2:3} => "bcd"; {substr:-3:2} => "de"
rsubstr	Take from the right, or a right-anchored window	`{rsubstr:N}` or `{rsubstr:anchor:len}`	{rsubstr:3} => "def"; {rsubstr:2:4} => "bcde"

Examples:

`<(System.ItemNameDisplay){substr:1:8}>` => first 8 characters (e.g., "ProjectPlan_v3" => "ProjectP")

`<(System.ItemNameDisplay){substr:-3:3}>` => last 3 characters via negative start (e.g., "Report_Q1" => "Q1")

`<(PDF:Title){substr:4:5}>` => 5 chars starting at position 4 (e.g., "DocTitle" => "Title")

`<(System.ItemFolderNameDisplay){rsubstr:4}>` => last 4 characters (e.g., "Marketing" => "ting")

`<(System.ItemFolderNameDisplay){rsubstr:2:5}>` => 5-char window ending 2 from right (e.g., "Campaign2025" => "ign20")

`<name{substr:1:1}{upper}>` => uppercase first letter only (e.g., "alpha" => "A")

<ext{rsubstr:3}> => typical 3-letter extension tail (e.g., ".jpeg" => "peg")

<(System.FileName){substr:1:999}> => safe clamp to "all" (e.g., unchanged)

C. Cutting by Delimiter

Add `:n` for the n-th occurrence (default 1), and optional `:ci` to ignore case.

Formatter	Keeps...	Syntax	Example (on "2024-Trip-Italy-RAW")
before	Left part excluding	`{before:-[:n][:ci]}`	{before:-:2} => "2024-

	delimiter		Trip"
after	Right part excluding delimiter	{after:-[:n][:ci]}	{after:-:2} => "Italy-RAW"
beforeinc	Left part including delimiter	{beforeinc:-[:n][:ci]}	{beforeinc:-:2} => "2024-Trip-"
afterinc	Right part including delimiter	{afterinc:-[:n][:ci]}	{afterinc:-:2} => "-Italy-RAW"

Examples:

<(System.ItemNameDisplay){before:}> => left of first “” (e.g., "IMG_4321_Raw" => "IMG")

<(System.ItemNameDisplay){after:}> => right of first “” (e.g., "IMG_4321_Raw" => "4321_Raw")

<(System.ItemNameDisplay){before:-:2}> => left of second “-” (e.g., "ACME-Project-Q3" => "ACME-Project")

<(System.ItemNameDisplay){after:-:2}> => right of second “-” (e.g., "ACME-Project-Q3" => "Q3")

<(System.Title){beforeinc: - }> => keep left part including delimiter (e.g., "Draft - Overview" => "Draft - ")

<(System.Title){afterinc: - }> => keep right part including delimiter (e.g., "Draft - Overview" => " - Overview")

<(MediaInfo:CompleteName){before:. :2:ci}> => before the 2nd dot (case-insensitive search)

<(exif:Model){after: :last}> => after the last space (e.g., "Canon EOS 5D" => "5D")

D. Slices (Format just a part)

Slices temporarily focus on a portion; all the following formatters act only on that slice. At the end the modified slice is automatically placed back into the original text.

Formatter	**Scope**	**Syntax**	**Notes & Example**
slice (by position)	Range by start and optional length	{slice:start[:len]}	1-based start; negative start counts from right. Example: <name {slice:1:3}{upper}> on "alpha" => "ALpha".
slice before	Left part before delimiter	{slice:before:delim[:n][:ci]}	n = occurrence (default 1). ci = case-insensitive. Example: <name{slice: before:-}{upper}> "holiday-photos-01" => "HOLIDAY-photos-01".
slice after	Right part after delimiter	{slice:after:delim[:n][:ci]}	Example: <name{slice: after:-}{titlecase}> "holiday-photos-01" => "holiday-Photos-01".
slice beforeinc	Left part including delimiter	{slice:beforeinc:delim[:n][:ci]}	Example: <(System. Title){slice:beforeinc: - }{upper}> "Intro - overview" => "INTRO - overview".

slice afterinc	Right part including delimiter	{slice:afterinc:delim[:n][:ci]}	Example: <(System.Title){slice:afterinc: - }{titlecase}> "Intro - overview" => "Intro - Overview".
slice between	Text strictly between left and right	{slice:between:left:right[:ci]}	Example: <name{slice:between:(:)}{set:2025}> "Report (Draft)" => "Report (2025)".
slicere before	Left part before regex match	{slicere:before:pattern[:n][:flags]}	Flags: i / ci. Example: keep before 2nd dot (case-insensitive): <(MediaInfo:CompleteName){slicere:before:\.:2:i}>
slicere after	Right part after regex match	{slicere:after:pattern[:n][:flags]}	Example: after last space via regex: <(exif:Model){slicere:after:\s+:last}>
slicere beforeinc	Left part incl. regex match	{slicere:beforeinc:pattern[:n][:flags]}	Example: include " - " and left part: <(System.Title){slicere:beforeinc:\s-\s}>
slicere afterinc	Right part incl. regex match	{slicere:afterinc:pattern[:n][:flags]}	Example: keep " - " and right part: <(System.Title){slicere:afterinc:\s-\s}>
slicere between	Between two regexes	{slicere:between:leftRegex:rightRegex[:flags]}	Example: text inside brackets (case-insensitive): <name{slicere:between:\[:\]:i}{titlecase}>
slicenum	Focus on one numeric run	{slicenum[:occurrence][:type]}	Occurrence: 1, 2, ... , last. Type: int (default), signed, float. Example: <name{slicenum}{math:+1}{pad0:3}> "Scene-002" => "Scene-003".

Basic slice (by position or delimiter)

```
{slice:start[:len]}
{slice:before:delim[:n][:ci]}
{slice:after:delim[:n][:ci]}
{slice:beforeinc:delim[:n][:ci]}
{slice:afterinc:delim[:n][:ci]}
{slice:between:left:right[:ci]}
```

Examples:

<name{slice:before:-}{upper}> turns "holiday-photos-01" into "HOLIDAY-photos-01".

<(System.Title){slice:before: - }{upper}> => uppercase the part before " - " only e.g., "intro - chapter one" => "INTRO - chapter one"

<name{slice:after:}{titlecase}> => Title Case the part after the first “” e.g., "photo_paris_day" =>

"photo_Paris Day"

`<name{slice:between:(:)}{set:2025}>` => replace the text inside parentheses with “2025” e.g., "Report (Draft)" => "Report (2025)"

`<(System.ItemNameDisplay){slice:1:3}{upper}>` => uppercase first 3 chars only e.g., "abc_project" => "ABC_project"

Regex slice (advanced)

```
{slicere:before:pattern[:n][:flags]}
{slicere:after:pattern[:n][:flags]}
{slicere:beforeinc:pattern[:n][:flags]}
{slicere:afterinc:pattern[:n][:flags]}
{slicere:between:leftRegex:rightRegex[:flags]}
```

Use `i`/`ci` for case-insensitive.

Number slice - focus on one number run within the text and operate on it only.

`{slicenum[:occurrence][:type]}`
Occurrence: `1, 2, …, last` (default 1). Type: `int` (default), `signed`, `float`.

Examples:
`<name{slicenum}{math:+1}>` => adds 1 to the first number (e.g., "Scene-002" => "Scene-003"). Try in Name (2) => Fixed.
`<name{slicenum:last}{z:0000}>` => formats the last number as 0000. Try in Name (2) => Fixed.

Important - "substr" vs "slice": `substr extracts a portion and replaces the whole value with that portion. slice temporarily focuses on a portion, applies subsequent formatters only to that part, and then splices the modified part back into the original text.`

`<name{substr:1:1}{upper}>` on "alpha" => "A" (only the extracted part remains).
`<name{slice:1:1}{upper}>` on "alpha" => "Alpha" (only the first letter is changed; the rest is preserved).

E. Replace and Set

Formatter	What it does	Syntax	Example => Result
replace	Replace all matches (literal); supports `:ci`.	`{replace:from:to[:ci]}`	{replace:RAW:JPG} on "IMG_RAW_01" => "IMG_JPG_01"
set	Overwrite the current value with given text (multiple args are joined with a colon)	`{set:text}`	{set:Chapter:01} => "Chapter:01"

Examples:

`<(System.Title){replace:raw:jpg:ci}>` => replace “raw” with “jpg”, ignore case e.g., "ACME raw export" => "ACME jpg export"

`<(name){replace:-: }>` => turn dashes into spaces e.g., "my-file-name" => "my file name"

`<(System.ItemFolderNameDisplay){replace: :_}>` => spaces => underscores e.g., "Project Files" => "Project_Files"

`<(PDF:Subject){set:Confidential}>` => overwrite subject with fixed text “Confidential”

`<(System.Title){slice:before: - }{set:Draft}>` => change only the left part to “Draft” e.g., "Intro -

Overview" => "Draft - Overview"

`<(MediaInfo:Format_Profile){replace:High@L4.2:High-4_2}>` => tidy special chars e.g., "High@L4.2" => "High-4_2"

F. Numbers - Padding, Grouping, Math

Formatter	Purpose	Syntax	Notes & Examples
int	Truncate to integer	`{int}`	12.9 => 12
num	Keep digits only, parse as integer	`{num}`	"v2.0-beta" => 20
math	Add/Sub/Mul/Div/Pow (integer math)	`{math:+K}` or `{math:add:K}` / `{math:sub:K}` / `{math:mul:K}` / `{math:div:K}` / `{math:pow:K}`	Division truncates; power uses non-negative exponent.
pad	Left-pad text to width with a character	`{pad:width[:char]}`	"7" with {pad:3:0} => "007"; "A" with {pad:3:_} =>"__A"
pad0	Zero-pad numbers to width (keeps sign)	`{pad0:width}`	-12 with width 3 => "-012"
padnum	Pad the first number run inside the string	`{padnum:width[:fill]}`	"ID-7a" => "ID-007a"
fixed	Format as fixed-point with exact decimals	`{fixed:places}`	12.3 with {fixed:2} => "12.30"
group	Add thousands separator (custom optional)	`{group[:sep]}`	12345 => "12,345"; {group:_} => "12_345"
z	Zero-pattern shorthand (integers or decimals)	`{z:000}` / `{z:00.00}`	3.1 with {z:00.00} => "03.10"; -12.34 with {z:000.0} => "-012.3"

Tip: combine `slicenum` + `math` + `pad0` to "find number => add 10 => zero-pad". Example: `<name{slicenum}{math:+10}{pad0:3}>` turns "file_220_rtu" into "file_230_rtu".

Examples:

`<(counter){pad0:3}>` => zero-pad a counter to width 3 (e.g., 7 => "007")

`<(System.Size){group}>` => add thousands separators (e.g., 12345678 => "12,345,678")

`<(System.Size){group:_}>` => custom separator underscore (=> "12_345_678")

`<(System.Media.Year){math:+1}>` => increment a year tag (e.g., 2024 => 2025)

`<(#System.Music.TrackNumber){pad0:2}>` => track number as two digits (e.g., 3 => "03")

`<(System.Video.FrameRate){fixed:2}>` => fixed 2 decimals (e.g., 29.97 => "29.97"; 30 => "30.00")

`<(name){slicenum}{math:+10}{pad0:4}>` => find first number in name, +10, pad 4 e.g., "build_23_alpha" => "build_0033_alpha"

`<(#exif:FNumber){z:00.0}>` => zero-pattern format for aperture (e.g., 2.8 => "02.8")

`<(MediaInfo:OverallBitRate){group}{set:kbps}{beforeinc: }{after:}}` => example combo: group, append unit Simpler: <(MediaInfo:OverallBitRate){group}{set: kbps}{afterinc:}> produces "1,024,000 kbps"

`<size{num}{math:/1024}{math:/1024}{fixed:2}>` => bytes => MB with 2 decimals e.g., 10485760 =>

"10.00"

G. Dates and Times

Use `{date:pattern}` to format date/time values (EXIF, file timestamps, etc.). Common tokens:

✓ YYYY (year, 4-digit), YY (2-digit)

✓ MM (month), DD (day)

✓ hh (hours 00–23), mm (minutes), ss (seconds)

✓ Use quotes for literal text, e.g., `{date:'Shot on 'YYYY-MM-DD}`

Examples:
`<($System.ItemDate){date:YYYY-MM-DD}>` => "2025-08-14"
`<($System.ItemDate){date:'Shot on 'YYYY-MM-DD 'at' hh-mm-ss}>` => "Shot on 2025-08-14 at 13-07-42"
Shortcut for date values: `<tag{YYYYMMDD}>` behaves like `<tag{date:YYYYMMDD}>`.
Default if pattern omitted: `%Y-%m-%d_%H-%M-%S` (e.g., 2025-08-11_13-07-42).

H. Length and Digit Removal

Formatter	What it does	Syntax	Example
len	Character length of the current value	`{len}`	"File-001" => 8
nodigits	Remove all digits 0–9	`{nodigits}`	"ABC123-45Z" => "ABC-Z"

Examples:

`<name{len}> =>` character count of the current file name (e.g., "Report_2025_Final" => 19).

`<(System.Title){len}> =>` character count of the title (e.g., "Quarterly Plan" => 14).

`<folder{len}> =>` length of the containing folder name (e.g., "Client_Acme" => 11).

`<(System.Image.Dimensions){len}> =>` length of a value like "4000 x 3000" (e.g., => 11).

`<(PDF:Title){trim}{len}> =>` trims edges first, then counts (e.g., " Design Spec " => 11).

`<name{nodigits}> =>` remove all digits (e.g., "IMG_0012" => "IMG_").

`<(System.Title){nodigits}> =>` drop digits but keep letters/symbols (e.g., "Q3 Plan 2025" => "Q Plan ").

`<folder{nodigits}> =>` e.g., "Release_2.0" => "Release_.".

`<(System.ItemNameDisplay){before:}{nodigits}> =>` remove digits only in the part before “” (e.g., "build12_alpha3" => before "" is "build12" => "build"; result spliced back: "build_alpha3").

`<(PDF:Subject){trim}{nodigits}> =>` first trim, then remove digits (e.g., " Rev2 Draft " => "Rev Draft").

I. Recipes

Try in *Name (2) - > Fixed:*

✓ **Uppercase the project code before the first dash:** `<name{slice:before:-}{upper}>` => "acme-Quarterly-report" becomes "ACME-Quarterly-report".

✓ **Add 10 to the first number and zero-pad to 3 digits:** `<name{slicenum}{math:+10}{pad0:3}>` =>

"clip_7_final" becomes "clip_017_final".

- ✓ **Get the text between [and], then Title Case it:** `<name{slice:between:[:]}{titlecase}>` => "photo_[the louvre]_01" becomes "photo_[The Louvre]_01".
- ✓ **Replace “RAW” with “JPG” ignoring case:** `<name{replace:raw:jpg:ci}>` => "IMG_raw_001" becomes "IMG_jpg_001".
- ✓ **Format a date as YYYYMMDD:** `<($System.ItemDate){date:YYYYMMDD}>` (or simply `<($System.ItemDate){YYYYMMDD}>`).
- ✓ **Uppercase the segment before the second underscore:** `<name{slice:before:_:2}{upper}>` => "acme_proj_phase-1" becomes "ACME_PROJ_phase-1".
- ✓ **Zero-pad the first number in the name to 4 digits:** `<name{padnum:4}>` => "IMG_7_final" becomes "IMG_0007_final".
- ✓ **Convert underscores to spaces, Title Case, then restore underscores:** `<name{replace:_ : }{titlecase}{replace: :_}>` => "my_project_notes" becomes "My_Project_Notes".
- ✓ **Replace only the text inside parentheses with “Final”:** `<name{slice:between:(:)}{set:Final}>` => "Spec (Draft) v2" becomes "Spec (Final) v2".
- ✓ **Keep only the part after the second dash and Title Case it:** `<name{after:-:2}{replace:-: }{titlecase}>` => "acme-project-q3-summary" becomes "Q3 Summary".
- ✓ **Use EXIF capture date as YYYY-MM-DD_hh-mm-ss:** `<($exif:DateTimeOriginal){date:YYYY-MM-DD_hh-mm-ss}>` => "2024:11:03 14:05:06" becomes "2024-11-03_14-05-06".
- ✓ **Round video frame rate (MediaInfo) to 2 decimals:** `<(MediaInfo:FrameRate){fixed:2}>` => "29.970" becomes "29.97".
- ✓ **Zero-pad document page count to 3 digits:** `<(#System.Document.PageCount){pad0:3}>` => "7" becomes "007".
- ✓ **Trim and replace spaces with underscores in PDF title:** `<(PDF:Title){trim}{replace: :_}>` => " Quarterly Report " becomes "Quarterly_Report".
- ✓ **Shorten SHA-256 hash to first 8 characters in uppercase:** `<(hash:sha256){substr:1:8}{upper}>` => "6fe8c1b3a9e4…" becomes "6FE8C1B3".

J. Full list of formatters

after, afterinc, before, beforeinc, date, default, fixed, group, int, len, lower, math, nodigits, num, pad, pad0, padnum, replace, rsubstr, set, slice, slicere, slicenum, substr, titlecase, trim, upper, z.

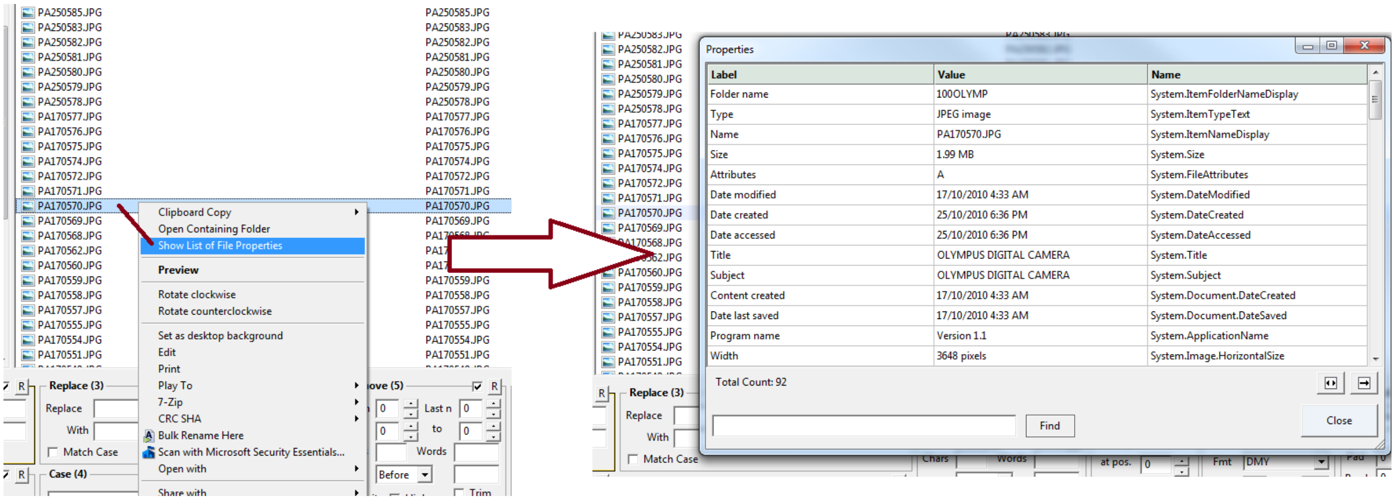
Using Windows File Properties to Rename Files

With Bulk Rename Utility you can read and use the Windows File Properties to rename your files. More information on the Windows Property System that can be set on Windows files can be found here: https://docs.microsoft.com/en-us/windows/win32/properties/props

Windows File Properties are only available on *Windows Vista or newer*.

*** To enable Windows File Properties tags support in Bulk Rename Utility, activate the option *'Extract Windows File Properties'* from the Renaming menu. This is required only in versions before 4.1.0.0. In 4.1.0.0 and later, Windows File Properties tags will work automatically when used, without enabling this option.

To show a list of all available file properties, for a file or a file type, in Bulk Rename Utility, right-click on a file in the main file list and select '*Show List of File Properties*' from the popup context menu. See image below.



Label	Value	Name
Folder name	100OLYMP	System.ItemFolderNameDisplay
Type	JPEG image	System.ItemTypeText
Name	PA170570.JPG	System.ItemNameDisplay
Size	1.99 MB	System.Size
Attributes	A	System.FileAttributes
Date modified	17/10/2010 4:33 AM	System.DateModified
Date created	25/10/2010 6:36 PM	System.DateCreated
Date accessed	25/10/2010 6:36 PM	System.DateAccessed
Title	OLYMPUS DIGITAL CAMERA	System.Title
Subject	OLYMPUS DIGITAL CAMERA	System.Subject
Content created	17/10/2010 4:33 AM	System.Document.DateCreated
Date last saved	17/10/2010 4:33 AM	System.Document.DateSaved
Program name	Version 1.1	System.ApplicationName
Width	3648 pixels	System.Image.HorizontalSize

Total Count: 92

Most rename text fields have two small icons on the right:
- *Chevron (arrow down)*: Opens a menu of Recent Entries for that field and, where supported, lets you insert Renaming Tags.
- *Six-dot button:* Opens the Zoom window. The Zoom window also shows the field’s Recent Entries and allows Renaming Tag insertion (where supported).

Tip: Right-click the six-dot button to go directly to the Insert Renaming Tags dialog.

Alternatively, you can also see the file properties of a file in Windows Explorer: right-click on a file and select "Properties" and then "Details".

Each file property has three entries: a LABEL, a VALUE and a NAME. You can select a file property by LABEL or by NAME and then use its VALUE to rename a file.

Using File Properties in Add (7), RegEx (1), Name (2) and Replace (3)

You can use the file properties in **Add(7), RegEx (1), Name (2)** and **Replace (3)** as tags.
To use a file property, specify the file property like this: <*(propertylabel)*> or <*(propertyname)*>*.* The tag will be replaced by the property value for that property.

For instance :
Prefix : <(System.Photo.CameraManufacturer)> might add "OLYMPUS" as a prefix.

Suffix : <(System.Image.HorizontalSize)> might add "3648 pixels" as a suffix.

Prefix : <(Title)> might add the file title as a prefix.

The file properties can be referred to by Label or Name. They can be used in Add(7) -> Prefix, Suffix, Insert. The file properties can also be used in RegEx(1) -> Replace and Replace(3) -> With.

IMPORTANT: You can also specify some extra characters in the property < > tag. For instance, "<(*Subject)* ->" will add the *Subject* and the - , but only if the *subject* is not empty. If (*Subject*) is empty, the whole tag < > will be empty. This is different from using "<(subject)> -". In this case - will always be added even if the file does not have a subject. For example: <[(System.Photo.ImageResolution)]> will resolve to the image resolution wrapped by [ ], if System.Photo.ImageResolution is not empty, otherwise the whole tag is empty. This will work only for renaming tags that include ( ), like <(Ttile*)*>, but not for tags without (), like <counter> or <name>, eg <name-> will not work use <name>- instead.

File Properties as Dates and Numbers

If you want to add a file property to a file name as a date or as a number, use the property formatting markers.

will format a property as a number. If a file property value has numbers and letters, the letters will be ignored.
$ will format a property as a date, according to the date and time formatting options as specified in Auto-Date (8).

For instance :
Prefix : <($System.Photo.DateTaken) - > will add the date taken as prefix followed by - , formatted according to values in Auto-Date (8).

Suffix : < - (#ISO speed)> might add " - 100" as suffix. The file property is added as number only.

Suffix : < - (ISO speed)> might add " - ISO100" as suffix. The file property is added as a text string.

Formatting: File properties can be formatted and changed using 'formatters'. For example, <(System. Photo.CameraManufacturer){upper}> will convert it into uppercase. Or <($system.itemdate){YYYY_DD_MM}> will expand to the item date in format YYYY_DD_MM. See renaming tag formatting specifications with full list.

Clipboard

A special tag is <clip> which is replaced by the current text content of the Windows Clipboard.

Using File Properties in Javascript

For more complex uses, besides just adding certain file properties to file names with Add (7), also Javascript and Javascript Conditions support Windows file properties, via the functions **fileProperty**, **filePropertyDate**, **filePropertyNum**. See Javascript Renaming for more information.

Using EXIF Properties to Rename Files

With Bulk Rename Utility you can read and use the EXIF file properties to rename your image files.

Exchangeable image file format (officially EXIF, according to JEIDA/JEITA/CIPA specifications) is a standard that specifies the formats for images, sound, and ancillary tags used by digital cameras (including smartphones), scanners and other systems handling images.

Bulk Rename Utility supports the latest EXIF metadata and attributes **version 2.2.** The following file types are processed: .jpg, .jpeg, .jpe, .jfif, .tif, .tiff, .dng, .heic, .heif, .avif. The following camera RAW formats are also supported: .crw, .cr2, .cr3, .nef, .nrw, .arw, .sr2, .srf, .srw, .orf, .rw2, .raf, .pef, .rwl, .3fr, .erf, .kdc, .dcr, .mef, .mos, .mrw, .x3f, .bay

*** To use EXIF Properties tags in Bulk Rename Utility, activate the option *'Extract EXIF data'* from the Renaming menu. This is required only in versions before 4.1.0.0. In 4.1.0.0 and later, EXIF property tags work automatically without enabling this option.

You can show a **list of all available EXIF attributes** for a file by **right-clicking on the file** in the main file list and select '*Show EXIF info (.JPG files)*' from the context menu.

Most rename text fields have two small icons on the right:

- *Chevron (arrow down)*: Opens a menu of Recent Entries for that field and, where supported, lets you insert Renaming Tags.

- *Six-dot button:* Opens the Zoom window. The Zoom window also shows the field’s Recent Entries and allows Renaming Tag insertion (where supported).

Tip: Right-click the six-dot button to go directly to the Insert Renaming Tags dialog.

Using EXIF Properties in Add (7), RegEx (1), Name (2) and Replace (3)

You can use the EXIF properties in **Add(7), RegEx (1), Name (2)** and **Replace (3)** as tags to add to file names.
To enable EXIF tags support in Bulk Rename Utility, activate the option *'Extract EXIF data'* from the Renaming menu.

To use a property, specify the property like this: *<(exifproperty)>.* The tag will be replaced by the property value for that file.

For instance :
Prefix : <(exif:ImageWidth)> might add "800" as a prefix.

Suffix : <(exif:ImageResolution)> might add "640x480 pixels" as a suffix.

Prefix : <(exif:Software)> might add the software used to create the image as a prefix.

The EXIF properties can be used in Add(7) -> Prefix, Suffix, Insert.
The EXIF properties can also be used in RegEx(1) -> Replace and Replace(3) -> With.

IMPORTANT: You can also specify some extra characters in the property < > tag. For instance, "<*(exif:Make)* ->" will add the *exif:Make* and the - , but only if the *exif:Make* is not empty. If (*exif:Make*) is empty, the whole tag < > will be empty. This is different from using "<(*exif:Make*)> -". In this case - will always be added even if the file does not have a camera make attribute. You can also specify multiple properties: <*(exif:Make)* - *(exif: ImageResolution)*>. For example: <[(exif:ImageResolution)]> will resolve to the image resolution wrapped by [ ], if exif:ImageResolution is not empty, otherwise the whole tag will be empty. This will work only for renaming tags that include ( ), like <*(exif:Software)*>, but not for tags without (), like <counter> or <name>, eg <name-> will not work use <name>- instead.

EXIF Properties as Dates and Numbers

If you want to add a property to a file name as a date or as a number, use the property formatting markers.

will format a property as a number. If a file property value has numbers and letters, the letters will be ignored.
$ will format a property as a date, according to the date and time formatting options as specified in Auto-Date (8).

For instance :
Prefix : <($exif:DateTimeOriginal) - > will add the date taken as prefix followed by - , formatted according to values in Auto-Date (8).

Suffix : < - (#exif:Flash)> might add " - 16" as suffix. The file property is added as number only.

Formatting: EXIF properties can be formatted and changed using 'formatters'. For example, <(exif:Software){lower}> will convert it into lowercase. Or <($exif:DateTimeOriginal){YYYY_DD_MM}> will expand to the item date in format YYYY_DD_MM. See renaming tag formatting specifications with full list.

Using EXIF Properties in Javascript

For more complex uses, besides just adding certain file properties to file names with Add (7), also Javascript and Javascript Conditions support Windows file properties, via the functions **fileProperty**, **filePropertyDate**, **filePropertyNum**. See Javascript Renaming for more information.

Example of EXIF Properties for an Image File JPG

exif:ImageResolution	640x480
exif:ImageWidth	640
exif:ImageHeight	480
exif:Make	NIKON
exif:Model	COOLPIX P6000
exif:Orientation	1
exif:XResolution	300
exif:YResolution	300
exif:ResolutionUnit	2
exif:Software	Nikon Transfer 1.1 W
exif:DateTime	2008:11:01 21:15:09
exif:DateTimeOriginal	2008:10:22 16:46:53
exif:DateTimeModified	2009:11:21 10:40:50
exif:DateTimeDigitized	2008:10:22 16:46:53
exif:ExposureTime	0.015432
exif:FNumber	5.6
exif:ExposureProgram	2
exif:ISOSpeedRatings	64
exif:FocalLength	22.1
exif:Flash	16
exif:MeteringMode	5

exif:LensInfo.FocalLengthIn35mm	103
exif:GeoLocation.Latitude	43.468243
exif:GeoLocation.Longitude	11.880172
exif:GeoLocation.GPSMapDatum	WGS-84
exif:GeoLocation.GPSTimeStamp	14 45 20.91
exif:GeoLocation.GPSDateStamp	2008:10:23
exif:OffsetTime	+01:00
exif:OffsetTimeDigitized	+01:00
exif:OffsetTimeOriginal	+01:00
exif:PixelXDimension	640
0exif:PixelYDimension	480

Using PDF and MediaInfo Properties

Bulk Rename Utility can read metadata from PDF files and from audio/video files via MediaInfo and expose those properties through renaming tags.
To enable these tags you must install two optional components: the **PDF DLL** and the **MediaInfo DLL**.

Installation (PDF DLL & MediaInfo DLL)

✓ Download the PDF DLL and the MediaInfo DLL from the Bulk Rename Utility website. The DLLs are available on this page.

✓ Close Bulk Rename Utility, then copy the DLLs into the application folder (next to the Bulk Rename Utility executable, e.g., 'Program Files' folder).

✓ Restart Bulk Rename Utility.

When are these tags available?

✓ PDF tags appear for .pdf files once the PDF DLL is installed.

✓ MediaInfo tags appear for audio/video files (e.g., .mp4, .mkv, .mp3) once the MediaInfo DLL is installed.

✓ Files without the relevant metadata (or file types not supported) will return an empty value for those tags.

Sample PDF renaming tags

Purpose	**Tag & Example**
PDF title (plain)	`<(PDF:Title)>`
Uppercase author	`<(PDF:Author){upper}> => "ADOBE SYSTEMS"`
Creation date as YYYY-MM-DD	`<($PDF:CreationDate){date:YYYY-MM-DD}>` e.g. `"2025-08-14"`
Keywords with underscores	`<(PDF:Keywords){replace: :_}> => "design_review_Q3"`
Creator with fallback	`<(PDF:Creator){default:Unknown Creator}>`

Sample MediaInfo renaming tags

Purpose	**Tag & Example**
Video format (tidy spaces)	`<(MediaInfo:Video_Format_List){replace: :_}> =>` e.g., `"AVC_MPEG-4"`
Frame rate with 2 decimals	`<(MediaInfo:FrameRate){fixed:2}> => "29.97"`
Overall bitrate with grouping	`<(MediaInfo:OverallBitRate){group}> => "12,000,000"`
File created date (local) as YYYYMMDD	`<($MediaInfo:File_Created_Date_Local){YYYYMMDD}> => "20250814"`
Duration (string as provided by MediaInfo)	`<(MediaInfo:Duration/String)> => "1 h 42 min"`
Audio codec list (uppercase)	`<(MediaInfo:Audio_Format_List){upper}> => "AAC"`

Examples you can try

- ✓ **PDF - Title then Author (uppercase author):** `<(PDF:Title)> - <(PDF:Author){upper}>`
- ✓ **PDF - Creation date as YYYYMMDD:** `<($PDF:CreationDate){YYYYMMDD}>`
- ✓ **MediaInfo - Format & frame rate:** `<(MediaInfo:Video_Format_List)>_<(MediaInfo:FrameRate){fixed:2}>fps`
- ✓ **MediaInfo - Grouped bitrate with unit:** `<(MediaInfo:OverallBitRate){group}>_kbps`
- ✓ **MediaInfo - File created date (local) as YYYY-MM-DD:** `<($MediaInfo:File_Created_Date_Local){YYYY-MM-DD}>`

Troubleshooting

- ✓ If tags return empty values, confirm the DLLs are present in the application folder.
- ✓ Restart Bulk Rename Utility after adding or replacing DLLs.
- ✓ Verify the file type actually contains the metadata you're asking for (e.g., PDFs may lack Title/Author; some videos lack codec info).

Listing available file properties & inserting tags

- ✓ Right-click a file in the main list and choose **List of File Properties** from the context menu to see all properties available for that item.
- ✓ From renaming input that support renaming tags, click the arrow-down chevron next to the field to open the **Recent Entries** menu, then select **Enter Renaming Tag...** to browse/insert a tag directly.
- ✓ PDF and MediaInfo properties will appear here only after installing their optional DLLs (PDF DLL and MediaInfo DLL).

Using the Custom Column

The custom column of the file list in Bulk Rename Utility is a column whose content can be defined by the user using renaming tags.

You can set the content of this column from the Display Options -> List menu.

The following tags can be used:

- Windows File Properties tags. For example: <(System.Photo.CameraManufacturer)>, <(System.Image.HorizontalSize)>, <(Title)>, etc.
- Version 2 EXIF tags. For example: <(exif:ImageWidth)>, <(exif:ImageResolution)>, <(exif:Software)>, etc.
- Hash tags. For example: <(hash:crc32)>, <(hash:md5)>,<(hash:sha1)>, etc.
- All other renaming tags.

Multiple tags can also be used, for example: <(Title)> - <(exif:ImageResolution)>.

The items in the file list can be sorted under this column by clicking on the column header.

Prevent Duplicates Format

When enabled, the **Prevent Duplicates** option allows to you to overcome the situation whereby a rename would fail because a file with the same name already exists: if you try to rename a file, and there's already a file with the same name, the software will make a subsequent attempt to rename the file but with a "_1" suffix. If this fails, it will try with "_2" as the suffix, and will continue up to "_99". The limit of 99, and the separator character underscore (_) are used by default.

You can change the default format that is used to prevent duplicates.
You can also set the duplicate counter to *Start From 1*, and you can prevent duplicates across folders.

To change the format, enable the option **Prevent Duplicates** in the Renaming Menu. A message will be shown that the option is now enabled, and clicking on "*Change Format*" will open a further dialog where you can specify the format string.

Supported tags:

TAG	Explanation
%name%	This tag identifies the file name that is duplicate (without extension)
%count%	This tag identifies a simple numeric counter with no padding: 1, 2, 3, ...
%count00%	This tag identifies a numeric counter padded with zeroes to 2 digits: 01, 02, 03, ...
%count000%	This tag identifies a numeric counter padded with zeroes to 3 digits: 001, 002, 003, ...
%count0000%	This tag identifies a numeric counter padded with zeroes to 4 digits: 0001, 0002, 0003, ...
%counta%	This tag identifies an alphabetic counter: a, b, c, d, ...
%countA%	This tag identifies a capitalized alphabetic counter: A, B, C, D, ...

Examples:

- The default format in Bulk Rename Utility is: *%name%_%count%* - Suffix is added to prevent duplicates: _1, _2, _3, ...
- To change the separator to be a # use: *%name%#%count%* - Suffix is added to prevent duplicates, #1,#2, #3, ...
- The Windows File Explorer uses the following format: *Copy (%count%) of %name%* - e.g. a prefix is added: Copy (1) of, Copy (2) of, ...
- You can use padded numbers: *%name%_%count00%* - Suffix is added to prevent duplicates: _01, _02, _03, _10, ... _99
- You can use a prefix instead: *%count00%%name%* - Prefix is added to prevent duplicates: 01, 02, 03, _10, ... 99
- You can use an alphabetic counter: *%name%%counta%* - Suffix is added to prevent duplicates: a, b, c, d, ... z
- You can use a capitalized alphabetic counter: *%name%%countA%* - Suffix is added to prevent duplicates: A, B, C, D, ... Z

Auto Refresh On / Off

In the main Bulk Rename Utility window, you have the option to turn Auto Refresh On or Off.
When Auto Refresh is Off, then the new names are not automatically recalculated each time a renaming criteria changes, but only when clicking on Preview, Rename, or switching Auto Refresh back to On.
This is helpful when a large number of names are currently selected and recalculating each file name while the renaming criteria are changing takes time. In this case, Auto Refresh can be switched off while renaming criteria are changed and then switched back on.

Quick Access to .bru Files

In the main Bulk Rename Utility window, positioned just above the Preview button, the button "open" is present to quickly open certain .bru files for quick access.
You can add specific .bru files to this menu and organize them into the right order as needed.

A .bru file or favourite file is a collection of configuration values that can be used to provide quick access to commonly-used criteria. For example, if you frequently want to rename a bunch of pictures in one folder, and you frequently want to rename a bunch of sound files in another folder, then you could create two Favourites called Pictures and Sound Files. Each Favourite holds with it the selection criteria, rename criteria, and optionally the current folder.

Detached Renaming Criteria

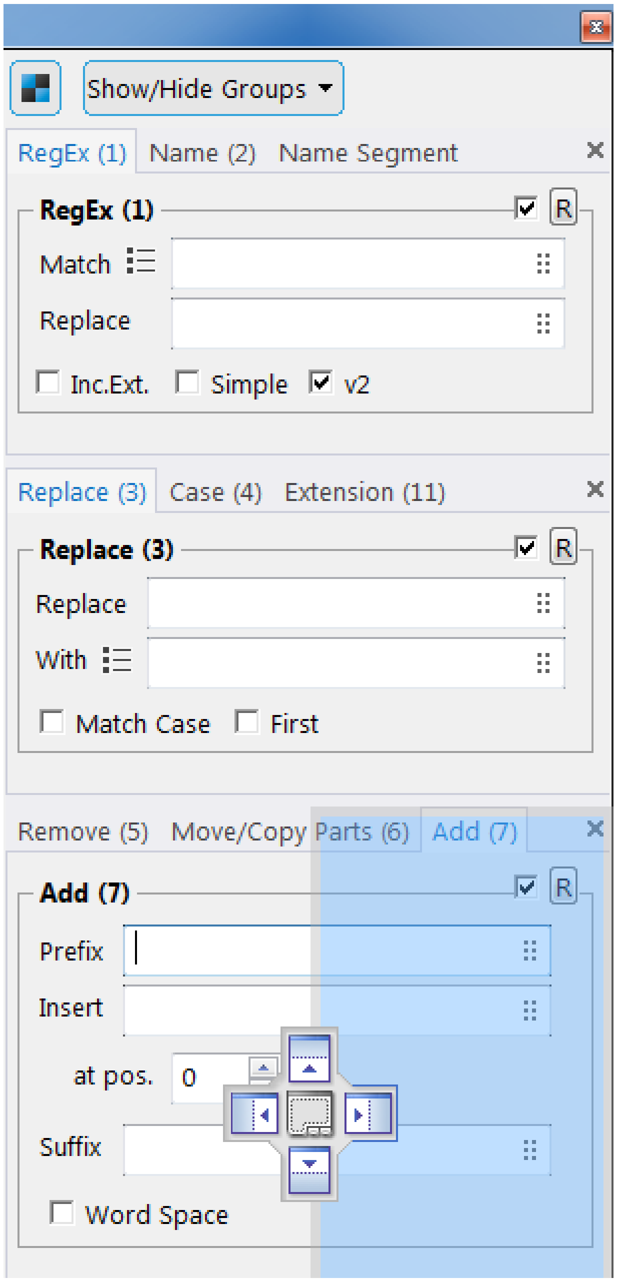
The renaming criteria, or renaming options, can be displayed in a separate window to help manage the Bulk Renaming Utility space and user interface.

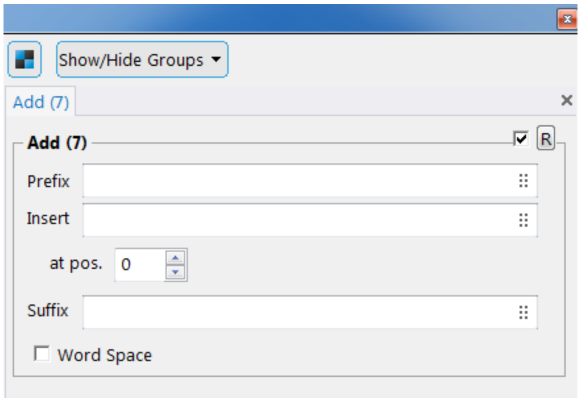
Any modifications made to the renaming criteria in the detached window are likewise mirrored in the main window, as this separate window is synchronized with it.

To maximize and enlarge the file list on the main window to display more items while the detached window is open, see the Display Menu->List settings.

All the renaming criteria in the detached window are organized in separate tabs and can be moved around, opened, or closed if not in use, as shown in the pictures.

To open the detached renaming criteria window, select Display Options -> Detach Renaming Criteria, or press F12, or click on the icon next to the Preview button.





JavaScript Renaming

Javascript renaming gives you total flexibility and full control of your file renaming needs by using Javascript code.

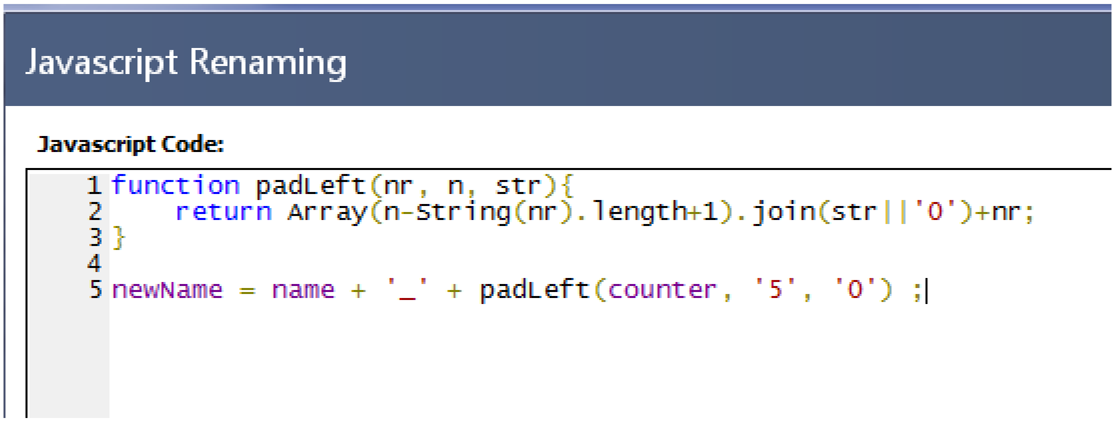
Bulk Rename Utility uses the V8 Javascript, Google's high performance Javascript engine also used in Google Chrome. V8 implements ECMAScript as specified in ECMA-262, 5th edition.

All standard Javascript syntax and functions are supported. Additional javascript libraries can be included, see below.

Javascript code is processed as the last step after all other renaming options have been applied. The javascript code is applied to each object as the last step, before it is renamed.

A commercial license for Bulk Rename Utility is required to use Javascript Renaming.

Here is an example of a Javascript script which changes a file name in Bulk Rename Utility. The script is applied to each selected file. This script adds a counter, padded with up to 5 zeros, after each file name:



```
function padLeft(nr, n, str){
    return Array(n-String(nr).length+1).join(str||'0')+nr;
}

newName = name + '_' + padLeft(counter, '5', '0') ;
```

Explanation:

- The function padLeft is defined.
- **newName** represents the new name of the file.
- **name** represents the current name of the file.
- **counter** is a counter that starts from 1 and it's incremented for each file.

More examples can be found on our website: Javascript Renaming Examples

Javascript code can be easily loaded from or saved to a file, or copied and pasted into the editor using Ctrl+C and Ctrl+V.

Javascript Bulk Rename Utility Constants and Variables

The following special constants and variables are available in Bulk Rename Utility javascript.

Variables can be modified, while constants have a fixed value and can not be changed.

Constant / Variable	Type	Explanation
name	Constant	This constant contains the name of the object (i.e. file or folder name) being processed, after all Bulk Rename Utility renaming options 1 to 13 and all specials have been applied. If the Bulk Rename Utility renaming

Constant / Variable	Type	Explanation
		options 1 to 13 and specials did not modify the name, then *name* will be the same as *origName* (see below).
newName	Variable	This variable contains the new name to be applied to the object. Modify this variable to modify the object name. If you do not modify the variable newName, then the javascript code has no effect on the name of an object.
origName	Constant	This constant contains the original name of the object before all Bulk Rename Utility renaming options 1 to 13 and all specials.
ext	Constant	This constant contains the extension of the object (i.e. file or folder name) being processed, after all Bulk Rename Utility renaming options 1 to 13 and all specials have been applied. If the Bulk Rename Utility renaming options 1 to 13 and specials did not modify the extension, then *ext* will be the same as *origExt* (see below).
newExt	Variable	This variable contains the new extension to be applied to the object. Modify this variable to modify the object extension. If you do not modify the variable newExt, then the javascript code has no effect on the extension of an object.
origExt	Constant	This constant contains the original extension of the object before all Bulk Rename Utility renaming options 1 to 13 and all specials.
location	Constant	This constant contains the location of the object as specified in *Location (13)*
newLocation	Variable	This variable contains a modified *location (13)* for the object. If you do not modify the variable newLocation, then the javascript code has no effect on the location (13) of an object.
subDir	Constant	This constant contains the object sub directory (it matches the subdir column in the file list). It is only applicable if subfolders have been included in Selection (12).
counter	Constant	A counter that is incremented during the renaming operation, starting from 1. A script can be run multiple times in a batch rename and the counter is incremented each time starting from 1. It is an integer that represents the index of the current rename iteration.
newModified	Variable	This variable (JavaScript Date) holds the new modified timestamp for the object. Assign a new date (and time) to newModified to update the object's modified timestamp. If you leave it unchanged, no update will be applied.
newCreated	Variable	This variable (JavaScript Date) holds the new created timestamp for the object. Assign a new Date (and time) to newCreated to update the object's created timestamp. If you leave it unchanged, no update will be applied.

Javascript Bulk Rename Utility Functions

The following special functions are available in Bulk Rename Utility javascript.

Function	Explanation	Details / Examples
object("value")	This function returns specific values for the object being processed.	The following *values* are supported. **object**("*folder*") : returns the folder for the current object being processed. **object**("container") : returns the name of the containing folder

Function	Explanation	Details / Examples
		for the current object being processed. **object**("container[*num*]") : *num* index can be a number from 1 to 9. Returns the name of the containing folder at *num* level for the current object being processed, where 1 is the first containing folder (same as just using *container* without index), 2 is the second folder, etc, up to a maximum of 9. If the containing folder at that level does not exist, then an empty string is returned. **object**("containers") : returns the total number of the containing folders for the current object being processed. This includes also the drive, e.g., c: **object**("*autonumber*") : returns the autonumber for the current object being processed, taken from Numbering (10). Use the **counter** variable above unless you need the autonumber from Numbering (10). **object**("*autodate*") : returns the autodate for the current object being processed, taken from Auto Date (8). **object**("*isdir*") : returns true if the current object being processed is a directory. **object**("*size*") : returns the file size for the current object being processed in bytes. **object**("attr") : returns a string containing the Windows file attributes for this object, encoded as single letters. Multiple attributes may be present at the same time; they are returned as a concatenated string. D (Directory), R (Read-only), H (Hidden), S (System), A (Archive), C (Compressed), E (Encrypted), I (Not content indexed), O (Offline), P (Sparse file), L (Reparse point / symbolic link), T (Temporary). If no attributes are set, object("attr") returns "-". Example to select read-only files: object("attr").indexOf("R") !== -1 **object**("*modified*") : returns the modified timestamp for the current object being processed. (Javascript Date) **object**("*created*") : returns the created timestamp for the current object being processed. (Javascript Date) **object**("*accessed*") : returns the accessed timestamp for the current object being processed. (Javascript Date)
exif("value")	This function returns EXIF values for the object being processed. See details.	The following values are supported. **exif**("%d") : Date/Time Taken (Javascript date) **exif**("taken") : Date/Time Taken, same as %d (Javascript date) **exif**("%a") : Aperture **exif**("%c") : Comments **exif**("%e") : Exposure **exif**("%f") : Focal Length **exif**("%xb") : Exposure Bias **exif**("%ma") : Camera Make **exif**("%mo") : Camera Model
fileProperty ("name") **fileProperty ("label")**	This function returns the Windows File Property value for the object being processed, as a *text string*, by name or by label.	**fileProperty**("Height") **fileProperty**("Dimensions") **fileProperty**("Total bitrate") **fileProperty**("System.Media.Duration") **fileProperty**("Channels")

Function	Explanation	Details / Examples
	See details. This function also returns the EXIF Properties of an image. If the property is not found, the function will return *null*.	**fileProperty**("exif:XResolution") **fileProperty**("exif:ISOSpeedRatings") **fileProperty**("hash:sha256") **fileProperty**("hash:md5") etc. You can list all file property labels or names that can be used in Bulk Rename Utility from the context menu of a file in the main file list (right-click) or in Windows Explorer, right-click on a file and select "Properties". Each file type might have different properties available.
filePropertyDate ("name") **filePropertyDate ("label")**	This function returns the Windows File Property value for the object being processed, as a *date object*, by name or by label. See details. This function also returns the EXIF Properties of an image. If the property is not found, the function will return *null*.	**filePropertyDate**("System.Document.DateCreated") **filePropertyDate**("exif:DateTimeOriginal") You can list all file property labels or names that can be used in Bulk Rename Utility from the context menu of a file in the main file list (right-click) or in Windows Explorer, right-click on a file and select "Properties". Each file type might have different properties available.
filePropertyNum ("name") **filePropertyNum ("label")**	This function returns the Windows File Property value for the object being processed, as a *number*, by name or by label. See details. This function also returns the EXIF Properties of an image. If the property is not found, the function will return *null*.	**newName = filePropertyNum**("Width") + **name**; **newName = filePropertyNum**("exif:Orientation") + "-" + **name**; You can list all file property labels or names that can be used in Bulk Rename Utility from the context menu of a file in the main file list (right-click) or in Windows Explorer, right-click on a file and select "Properties". Each file type might have different properties available.
clipboard()	This function returns the text that is currently in the Windows Clipboard.	**newName** = **clipboard()** + " - video";
alert("text")	This function shows a popup message.	**alert**("object new name is: " + newName);
regEx("text", "match", "replace")	This function processes a regular expression in the same way as it is done in section RegEx (1). Using this function is possible to process multiple regular expressions.	This code will switch a file name around "-". Please note that the character \ must be doubled \\ in Javascript strings. **newName** = **regEx**(**name**, "(.*)-(.*)", "\\2\\1");
removeExt ("text")	This function removes the file extension from the given text, if an extension is present.	MyStr = **removeExt**("file.txt");

Function	Explanation	Details / Examples
countFilesInFolder("pattern", includeHidden, includeSystem)	This function counts the files in a folder. You can specify a search pattern (such as "*.jpg" or "*.pdf"). If no pattern is specified, then all files are counted. *includeHidden*: Choose whether to include hidden files in the count. Use true to count hidden files or false to exclude them (the default is false). includeSystem: Choose whether to include system files in the count. Use true to count system files or false to exclude them (the default is false).	**newName** = **name** + "-" + **countFilesInFolder();** **newName** = **name** + "-" + **countFilesInFolder(**"*.pdf"**);** // counts PDF files only **newName** = **name** + "-" + **countFilesInFolder(**"", true**);** // include hidden files (but not system files) **newName** = **name** + "-" + **countFilesInFolder(**"*.*", true, true **);** // counts all files, including hidden and system files **newName** = **name** + "-" + **countFilesInFolder(**"*.jpg", true, false**);** // counts JPG files only including hidden files (but not system files) Appends the number of files in a folder to the name. If this function is applied to a folder name, it counts the files in that folder; for a file name, it counts the files in the folder containing that file.

listFilesInFolder("pattern", includeHidden, includeSystem)	This function returns an array of file names present in a folder. The array indices are zero based.
	You can specify a search pattern (such as "*.txt" or "*.pdf"). If no pattern is specified, then all files are listed.
	includeHidden: Choose whether to include hidden files in the list. Use true to list hidden files or false to exclude them (the default is false).
	includeSystem: Choose whether to include system files in the list. Use true to list system files or false to exclude them (the default is false).
		newName = **removeExt**(**listFilesInFolder()**); // the new name is set to the first file name in a folder, without its extension
		newName = **removeExt**(**listFilesInFolder(**"*.pdf"**)**); // the new name is set to the first PDF file name in a folder, without its extension
		newName = **removeExt**(**listFilesInFolder()**[0]); // the new name is set to the first file name in a folder, without its extension
		
		newName = **removeExt**(**listFilesInFolder(**"*.*", true, true**)**[9]); // the new name is set to the 10th file name in a folder, without its extension. The file list includes system and hidden files.

listFoldersInFolder("pattern", includeHidden, includeSystem)	This function returns an array of folder names present in a folder. The array indices are zero based.
	You can specify a search pattern (such as "A*" or "*B"). If no pattern is specified, then all folders are listed.
	includeHidden: Choose
		// add a tag if the folder contains at least one 4-digit year subfolder
		// e.g. folder has subfolders "2023", "notes" -> "Photos [has year]"
		if (**object**("isdir")) {
		    var folders = **listFoldersInFolder**("*", false, false);
		    var hasYearFolder = false;
		    var yearRegex = /^[0-9]{4}$/;
		    for (var i = 0; i < folders.length; i++) {

Function	Explanation	Details / Examples

	whether to include hidden folders in the count. Use true to list hidden folders or false to exclude them (the default is false).
	includeSystem: Choose whether to include system folders in the list. Use true to list system files or false to exclude them (the default is false).
		if (yearRegex.test(folders[i])) {
		hasYearFolder = true;
		break;
		}
		}
		if (hasYearFolder) { **newName** = **name** + " [has year]"; }
		}

isFirstRename()	Returns a Boolean indicating whether this is the first rename operation in a multiple renaming operation (i.e. when the counter equals 1). Use it to run code only on the very first rename.	// Prepend "FIRST_" only to the very first file
		if (**isFirstRename()**) { **newName** = "FIRST_" + **name + counter** ; } else
		{ **newName** = **name** + **counter**; }

setState ("varname", value)	Stores or updates a named state entry (string, number, or Date) in the script's persistent state. Calling with only key (or with value null/ undefined) removes that entry. Persists across multiple renames in the same operation and can be recalled with getState.	**setState**("current", 10); // Store a numeric value
		setState("now", new Date()); // Store the current date
		setState("start"); // Removes the "start" entry
		setState("currname", **name**); // Remember the current name
		// The values persist across multiple renames in the same operation and can be recalled with getState.

getState ("varname", defaultValue)	Retrieves the value previously stored with setState or setStateAtStart. Returns the stored type (Date, Number, or String). If the key varname isn't found, returns defaultValue (if provided) or undefined.	// Suppose you previously did setState("mynumber", 5):
		count = **getState**("mynumber", 0); // returns 5
		// If the key doesn't exist:
		newName = **getState**("username", "guest"); // returns guest

setStateAtStart ("varname", value)	Like setState, but only executes on the very first run of the script in a batch (i.e. when counter is 1). Subsequent invocations are no-ops. Persists across multiple renames in the same operation and can be recalled with getState.	**setStateAtStart**("initialDate", new Date()); // This will only set "initialDate" once, at the first rename. Later in the same batch, calling this again does nothing and the value stored remains the same.
		setStateAtStart("initialNumber", 10); // This will only set "initialNumber" once, at the first rename. Later in the same batch, calling this again does nothing and the value stored remains the same.
		setStateAtStart("firstname", **name**); // This will only set "firstname" once, at the first rename. Later in the same batch, calling this again does nothing and the value stored remains the same.

Function	Explanation	Details / Examples

fileMatch ("pattern", "flags", occurrence, maxSizeMB)	Reads the current file line by line (auto-detecting its text encoding), applies a regex search pattern, and returns the Nth occurrence (1-based). If no such match exists, returns null. Supports text files, PDF files, Office files.
	pattern (String) - regex match pattern (boost regex without /…/)
	flags (String, default "") - e.g. "i" = case insensitive match.
	occurrence (Number, default 1) - which match to return
	maxSizeMB (Number, default 30) - abort if file exceeds this size (MB)
		// Step 1: get "Username = Alice" from file
		line = **fileMatch**('^Username = .*');
		// Step 2: split and trim to get "Alice"
		username = line ? line.split('=')[1].trim() : null; // username set to "Alice"
		
		// Example: extract values from EML file and use for a new name
		// 1) grab the entire header line
		rawDate = **fileMatch**('^Date:.*', 'i'); // e.g. "Date: Tue, 08 Jul 2025 15:30:00 +0000"
		rawSubject = **fileMatch**('^Subject:\\s*.*', 'i'); // e.g. "Subject: Meeting Reminder"
		// 2) split on ":" and trim to get only the value
		dateValue = rawDate ? rawDate.substring(rawDate.indexOf(':') + 1).trim() : null;
		subjectValue = rawSubject ? rawSubject.substring(rawSubject.indexOf(':') + 1).trim() : null;
		// 3) Combine into newName
		var dt = new Date(dateValue);
		var formatted = **formatDateTime**(dt, "YYYY-MM-DD HH-mm-ss");
		combined = formatted + '_ ' + subjectValue;
		newName = combined.trim();

formatDateTime (dateObj, "formatStr", "localeName")	Converts a JavaScript Date (or millisecond timestamp) into a formatted String. Supports dates back to January 1, 1601 and respects the computer locale or an optional locale override.
	dateObj (Date|Number) A JS Date object or a number (ms since epoch).
	formatStr (String)
	Template using tokens:
	YYYY 4-digit year
	YY 2-digit year
	MMMM full month name
	MMM short month name
	MM 2-digit month
	M month number
	DD 2-digit day
	D day number
	HH 2-digit hour (00–23)
	hh 2-digit hour (01–12)
	H/h hour number
		// Example: ISO-style date/time in system locale
		now = new Date();
		s1 = **formatDateTime**(now, "YYYY-MM-DD HH:mm:ss"); // returns "2025-07-09 17:45:02"
		
		// Example: Full month and weekday in French
		dt = new Date(Date.UTC(2025, 6, 8, 12, 0, 0));
		s2 = **formatDateTime**(dt, "WD, D MMMM YYYY", "fr-FR"); // returns "mardi, 8 juillet 2025"
		
		// Example YYYY-MM-DD: Create a date for July 9, 2025 (months are 0-based)
		dt = new Date(2025, 6, 9);
		// Format as "YYYY-MM-DD"
		isoDate = formatDateTime(dt, "YYYY-MM-DD"); // isoDate = "2025-07-09"
		
		// Example: retrieve the object's modified timestamp, format it as YYYY_MMMM_D (e.g. 2025_July_9), and then prepend it, separated by an underscore, to the original filename
		// 1) Get the object's modified Date
		var modDate = object("Modified");
		// 2) Format as "YYYY_MMMM_D" so each part is joined by underscores
		var dateStr = **formatDateTime**(modDate, "YYYY_MMMM_D"); // e.g. "2025_July_9"
		// 3) Prepend to the existing name with another underscore

Function	Explanation	Details / Examples
	mm 2-digit minutes ss 2-digit seconds A/a AM / PM WD3 short weekday (Mon) WD full weekday (Monday) *localeName* (String, optional) IETF language tag (e.g. "en-US", "fr-FR"). Defaults to the system locale.	**newName** = dateStr + "_" + **name**;

padNum(n, width, "padPattern")	Left-pad the string form of *n* (a number or text) so that its total length is at least *width*. The padding is built by repeating *padPattern* (default "0") on the left and truncating it if it overshoots. Returns a String.
	
	n: A Number or String.
	width: A Number >= 0, max 1000. The minimum total length of the returned string.
	padPattern (optional): A String to use for padding. Defaults to "0". If its length is greater than 1, its characters are repeated as needed (and truncated if necessary) to reach exactly width - length(n) characters.
		// Pad a number with zeros to 5 digits:
		padNum(42, 5); // "00042"
		// Pad text with spaces to length 10:
		padNum("apple", 10, " "); // " apple"
		// multi-character pad pattern:
		padNum("A", 6, "01"); // padPattern "01" repeated: "010101", truncated to 5 chars, returns "01010A"
		// If width is less than length(n), returns n unchanged:
		padNum("1234567", 5); // "1234567"

askOnce ("prompt", "defaultValue")	Shows a prompt dialog with the given *prompt* text (and optional *defaultValue*) and returns the answer as string, but only on the very first invocation of that exact prompt during a batch. Subsequent calls with the same prompt text immediately return the stored answer without showing the dialog again. If the user cancels	// Example 1: One-time filename prefix
		var prefix = **askOnce**("Enter filename prefix:", "IMG_");
		// Returns user's input (or "IMG_" if they left it blank).
		// Further calls to askOnce with the same prompt will return the same prefix.
		// Apply prefix to every file in this batch
		newName = prefix + **name**;
		
		// Example 2: Ask for a multiplier only once
		var factor = parseFloat(**askOnce**("Enter scale factor:", "1.0"));
		if (isNaN(factor)) { throw new Error("Invalid factor"); }

Function	Explanation	Details / Examples

	the dialog, a JavaScript exception is thrown and execution stops.
	prompt (String) The message to display in the input dialog.
	defaultValue (optional, String)
	The initial value shown in the input field. If omitted or left blank by the user, the returned string will be empty.
		// Apply to each file's numeric suffix
		newName = **name**.replace(/(\d+)$/, function(_, num){
		  return Math.round(parseInt(num,10) * factor);
		});

setEnv ("varname", "value")	Sets the value of an environment variable	**setenv**('BRUNAME', **name**); // saved across multiple renaming operations until the application closes

parseTags ("tags")	Parse a renaming tag with formatting. *Note*: the function parseTags is not available for the Javascript Filter Condition.	**newName** = **parseTags**("<(System.Title){default:Untitled}>");
		newName = **parseTags**("<name{substr:1:10}>");

getEnv ("varname")	Gets the value of an environment variable	var value = **getenv**('USERNAME'); // available across multiple renaming operations until the applications closes

include ("filename")	Includes and runs an extra javascript file. The file location is relative to the current working directory, usually the Bulk Rename Utility installation folder, unless Bulk Rename Utility is running from a different directory.	**include**('js/sugar.js');
		include('file.js');
		include('../afile.js');

require ("filename")	Same as include but if the file is not found, javascript execution is stopped and an error is reported.	**require**('js/date.js');
		require('myfile.js');

About Conditional Renaming

Setting the variable **newName** back to the **origName** is like canceling the renaming of an object, as the new name is set back to the object's original name. This feature allows you to perform conditional renaming: for example, you could set **newName** back to **origName** only for objects with a certain modified timestamp. The modified timestamp is accessed via **object("modified")**.

The same is valid for **newExt** and **newLocation**.

Javascript renaming examples can be found on our website: Javascript Renaming Examples

Javascript Extension Libraries

Extension javascript libraries and files can be included with the functions **include** and **require**, see above.

There are two Javascript libraries that are already included in Bulk Rename Utility. They are saved in the *js* folder in the installation directory. File names are *sugar.js* and *date.js*.

These two libraries can be made easily available to javascript code in Bulk Rename Utility using the options '*Javascript Libraries...*" in the '*Special*' menu: activating the option 'Include sugar.js' is equivalent to adding **require('js/sugar.js')** at the start of your javascript code and activating the option 'Include date.js' is equivalent to adding **require('js/date.js')** at the start of your javascript code. Using the menu options is much easier.

About Sugar.js library

Information on the **sugar.js** javascript library and its supported functions/API can be found on the Sugar library website: http://sugarjs.com/

Sugar is a very powerful library. It adds many useful functions to work with dates, text and more in Javascript, see: http://sugarjs.com/api

The Sugar library is Copyright (c) 2014 Andrew Plummer. It is licensed under the MIT license.

Include Sugar.js library support in Bulk Rename Utility using the menu option '*Include sugar.js*' in the '*Special->Javascript Libraries...*'

About Date.js library

More information on the **date.js** javascript library and its supported functions/API can be found on the Date library website: http://datejs.com/

Date.js is a very powerful library. It adds many useful functions to work with dates in Javascript.

The Date.js library is copyright (c) 2006-2007, Coolite Inc. all rights reserved, licensed under the MIT license.

Include Date.js library support in Bulk Rename Utility by using the menu option '*Include date.js*' in the '*Special->Javascript Libraries...*'

JavaScript Filter Condition

The Javascript filter condition is part of the Filter (12) box.
In the Javascript filter condition you can use Javascript syntax. If the condition evaluates to true for an object in the file list, then that object is included, otherwise it is not. The Javascript condition supports all the functions, variables and constants that are available for Javascript Renaming.

The Javascript condition allows selection/inclusion of files/folders based on objects's name and also date, size, time, exif, attributes, etc.

Here is are some examples of Javascript conditions that can be used:

Javascript Filter Condition Examples

Javascript Filter Condition	**Requires sugar.js (*)**	**Requires date.js (**)**	**Result**
name.endsWith(/[q-z]/)	Y	N	Include all objects that end with 'q' to 'z'.
name.startsWith(/[a-d]/, null, false)	Y	N	Include all objects that start with 'a' to 'd', case insensitive.
ext.isBlank()	Y	N	Include all objects that have no extension.
object('modified').getTime() == **exif**('%d').getTime()	N	N	Include all files that have the Windows modified timestamp matching the EXIF date taken.
object('modified').getTime() != **exif**('%d').getTime() && **exif**('%d').getTime() != 0	N	N	Include all files that have the Windows modified timestamp not matching the EXIF date taken and the EXIF date taken is not empty.
object('size') == 0	N	N	Include all objects with zero size. Disable option to include folders in Filters (12) to include only files.
object('size') > 10000	N	N	Include all files larger than 10000 bytes.
object('modified').daysAgo() < 31	Y	N	Include all objects with the Windows modified timestamp more recent than 31 days ago. daysAgo() requires sugar.js, see note below.
object('modified').weeksAgo() < 4	Y	N	Include all objects with the Windows modified timestamp more recent than 4 weeks ago. weeksAgo() requires sugar.js, see note below.
object('modified').isBetween('yesterday', 'tomorrow');	Y	N	Include all objects with the Windows modified timestamp between yesterday and tomorrow. isBetween() requires sugar.js, see note below.
object('modified').isBetween('the beginning of last month', 'today');	Y	N	Include all objects with the Windows modified timestamp between the beginning of last month and today. isBetween() requires sugar.js, see note below.
exif('taken').isBetween('the beginning of last month', 'today');	Y	N	Include all objects with the EXIF taken date between the beginning of last month and today. isBetween() requires sugar.js, see

Javascript Filter Condition	**Requires sugar.js (*)**	**Requires date.js (**)**	**Result**
			note below.
var objsize = object('size'); objsize !=0;	N	N	Include all non-empty objects. This examples shows that multiple statements separated by ';' can be included in the condition.
include("myfile.js"); myFunction(**name**) == true;	N	N	Includes myfile.js which defines myFunction then include object if myFunction is true.
if (**filePropertyDate**('System.Document.DateCreated')) { **filePropertyDate**('System.Document.DateCreated').yearsAgo() < 2 }	Y	N	Include all objects with the Windows File Property date 'System.Document.DateCreated' more recent than 2 years ago. yearsAgo() requires sugar.js, see note below.
filePropertyNum('Width') == 3840;	N	N	Include all images with Width (pixels) equal to 3840.
name.startsWith(**clipboard**());	Y	N	Include all files (or folders) with name starting with the text content of the Windows Clipboard.

More examples can be found on our website: Javascript Examples

----

(*) *sugar.js* support can be activated in Bulk Rename Utility, menu 'Special' -> 'Javascript Libraries'. See sugar.js

(**) *date.js* support can be activated in Bulk Rename Utility, menu 'Special' -> 'Javascript Libraries'. See date.js

Link Files By Extension

You can link files by their extension in the Renaming Options Menu.

This feature allows you to link files by their extension, where you specify a **Master File Extension** and one or more **Linked File Extensions**.
Once linked, the file names with the master file extension will determine how files with the linked extensions are renamed.

If you want to link files across folders (not just within the same folder), enable the **Allow Link Across Folders** option.

Note: Linked extensions can include multiple values, separated by commas.

Examples:

Master File Extension: jpg
Linked File Extensions: raw
Result:
If filename.jpg and filename.raw are selected, any renaming applied to filename.jpg will also be applied to filename.raw.

Master File Extension: jpg
Linked File Extensions: raw,png
Result:
If filename.jpg, filename.raw, and filename.png are selected, any renaming applied to filename.jpg will also be applied to both filename.raw and filename.png.

Important Notes:

- All linked files must appear in the file list of Bulk Rename Utility and must be selected for the renaming to be applied correctly.
- For example, if filename.jpg and filename.raw are linked and need to be renamed with the same name, both files must be in the file list and be selected.
- You can verify if the files are correctly linked by checking the New Name column or using the Preview feature.

Exporting the File List

You can export the current list of files and folders from Bulk Rename Utility to use it in another program, such as a spreadsheet or text editor.
To export the list of files and folders, select them in the Bulk Rename list (Ctrl+A to select all present) and then right-click to open the context menu.
From the context menu, select "*Copy to Clipboard*" and then select what information you would like to export, such as pathname, filename, extension, new filename, or all.
Once copied into the Windows Clipboard, you will be able to paste the list into a spreadsheet, text editor, or other software.
For example, you can apply renaming changes as required to this exported list and then re-import it in Bulk Rename Utility using the "Import Rename Pairs" functionality.

Importing Files to Rename

Bulk Rename Utility provides two convenient methods for importing lists of files you want to rename.
These methods are ideal when you're working with large batches of files or have predefined rename pairs.

Importing From a Text File (CSV)

This method allows you to import a CSV (Comma-Separated Values) file that contains pairs of original filenames and their corresponding new names.
Each line in the file should follow this format:

original_filename.ext,new_filename.ext

The CSV file can be created using any text editor (e.g., Notepad) or spreadsheet application (e.g., Microsoft Excel).

More info here: Renaming From A Text File (CSV)

Once imported, you can then review, select and apply the changes.
This method is especially useful when you already have a list of filenames to be changed and want to ensure exact replacements.

Importing Using the Clipboard

You can also import files directly from the clipboard using the Paste from Clipboard feature. Copy full file paths into the clipboard from Windows File Explorer, Text documents, Excel or other editors.
More details here: Renaming Using The Clipboard

These importing options make it easier to manage complex renaming tasks, especially when the files to be renamed are located in different folders or when the new names are predefined.

Command Line Parameters

On the command line you can specify the name of a favourite file to open at start-up or a directory path to scan, i.e.:

"Bulk Rename Utility.exe" *filename.bru*

or

"Bulk Rename Utility.exe" *directorypath*

Other command line options

Command Line Parameters for automatic license code registration:

"Bulk Rename Utility.exe" /writeregkey:"AAAA|BBBB" [/elevated]

/writeregkey:"AAAA|BBBB" : Enter a license code via the command line, where AAAA is the registration text and BBBB the registration key, separated only by the character |. The whole text must be surrounded by quotes ".
Program will return 0 if the operation was successful.

[/elevated] : Optionally instructs to prompt for elevation to administrator if needed (to register for all users on computer).

Examples:
"C:\Program Files\Bulk Rename Utility\Bulk Rename Utility.exe" /writeregkey:"Paul|1234"
"C:\Program Files\Bulk Rename Utility\Bulk Rename Utility.exe" /writeregkey:"Paul|1234" /elevated

Support

If you have any comments or suggestions, please get in touch. The latest version of the application can always be downloaded from the website.

Remember, the easiest way to find your way around the application is to experiment. You won't do any harm unless you hit the Rename button, so you can play around with as many settings as you like!

The application's homepage is at www.bulkrenameutility.co.uk

Contact information is at www.bulkrenameutility.co.uk/Contact.php

The Support Forums are available at the same website.

If you get in touch with a problem then please email the following details:

- Your version of Windows (e.g. Windows 7, Windows 10, Windows 11, Windows Server, etc.)
- The version of the utility (found at Help -> About)
- How to recreate the problem

Assistants

Rename Assistant

The AI-powered assistant guides you through creating rule-based renaming. Just describe what you want (for example, “prepend today’s date” or “replace spaces with underscores”) and it will generate and preview a set of renaming steps you can apply to your files and folders. Rename Assistant

RegEx Assistant

Use natural-language prompts (for example, “match a number and move it to the front” or “remove all digits”) and the AI will build a regular-expression pattern that you can immediately try on your file list. RegEx Assistant

JavaScript Assistant

Describe the custom logic you need (for example, “increment a year in each filename,” “swap two words around,” or “increment number by 10”) and the AI will generate ready-to-paste JavaScript code for Bulk Rename Utility’s scripting interface. JavaScript Assistant

Ask a Question

Ask a question about Bulk Rename Utility and get a quick, AI-powered answer. Ask a Question

End-User License Agreement

Bulk Rename Utility

Copyright© TGRMN Software

Bulk Rename Utility - End-User License Agreement

This is a license agreement between you, a final user of computer software, and TGRMN Software (hereinafter "TGRMN", "TGRMN Software" or "TGRMN Software Company").

Please read carefully the following terms and conditions before using this software. Unless you have a different license agreement signed by TGRMN Software your use of this software indicates your acceptance of this license agreement and warranty. Installing and using Bulk Rename Utility signifies acceptance of the terms and conditions of this license agreement. If you do not agree with the terms and conditions of this license agreement you must remove Bulk Rename Utility files from your storage devices and cease to use the product.

* Usage *

Bulk Rename Utility is free of charge for personal, private use at home. To use Bulk Rename Utility for or within a business entity, company, or commercial environment and/or for commercial purposes, or within government departments or agencies, a commercial license is required for each computer where Bulk Rename Utility is installed. The commercial licenses can be purchased from our website: http://bulkrenameutility.co.uk

One commercial license is required for each computer where Bulk Rename Utility is installed. You may access Bulk Rename Utility through a network, provided that you have obtained individual licenses for the software covering all the computers and users that will access the software through the network. For instance, if 10 different computers will access Bulk Rename Utility on the network, each computer must have its own commercial license, regardless of whether Bulk Rename Utility is used at different times or concurrently.

Academic Use: the Bulk Rename Utility academic license allows educational institutions and their members (students, faculty, and researchers) to use the software for non-commercial, academic, and research purposes. This license is restricted to faculty, staff, and students of educational institutions. The Software may not be used for any commercial purpose under this license. One academic license is required for each computer where Bulk Rename Utility is installed. The academic licenses can be purchased from our website: http://bulkrenameutility.co.uk

Home Use: Bulk Rename Utility is free of charge for personal, private use at home. To unblock all the features of the software and to support software development, a home license can be purchased from our website. A home license unblocks all the features of the software and covers personal, private use at home for non-commercial purposes.

* Disclaimer of Warranty *

THIS SOFTWARE IS PROVIDED BY TGRMN SOFTWARE "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE

IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TGRMN SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Because of the various hardware and software environments into which Bulk Rename Utility may be used, NO WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE IS OFFERED.

Good data processing procedure dictates that any program be thoroughly tested with non-critical data before relying on it. The user must assume the entire risk of using Bulk Rename Utility.

* Copyright and Trademark Rights *

The Software is owned by TGRMN Software. This Agreement does not grant you any intellectual property rights in the Software. TGRMN Software owns Copyright and all Intellectual Property Rights to the Software and to any copies made of the Software.

* Restrictions *

You agree not to modify, adapt, translate, reverse engineer, decompile, disassemble or otherwise attempt to discover the source code of the Software. You may not use, copy, modify or transfer copies of the Software except as provided in this licence. You may not decompile, disassemble, or create derivative works based upon the Software. You may not modify, adapt, translate, or create derivative works based upon the written documentation. You may not sub-license, rent, lease, sell or assign the Software to others. All explicitly non-granted rights are reserved.

This license is provided personally to you and for that reason it does not allow you to make any duplicate (copy) to be sold, borrowed, assigned, leased or transferred in any manner whatsoever to another person. Any transfer executed in violation of this provision shall be deemed invalid and constitute a reason for termination of your license validity.

* Governing Law *

This Agreement shall be governed by, construed and enforced in accordance with the internal substantive laws (and not the laws of choice of laws) of South Australia, Australia, without giving effect to the conflict of laws provisions. Sole venue shall be in the applicable state and federal courts of South Australia.

Buy Commercial License

Bulk Rename Utility is free of charge for personal, private use, at home. To use Bulk Rename Utility for or within a business entity, company or commercial environment and/or for commercial purposes, or within government departments or agencies, a commercial license is required for each computer where Bulk Rename Utility is installed.

The commercial licenses can be purchased from our website http://bulkrenameutility.co.uk and are delivered instantly via e-mail.

The commercial license also unlocks Javascript Renaming, Javascript file selection, and other features that are not available otherwise.

Buy Home/Academic License

Bulk Rename Utility is free of charge for personal, private use, at home. To use Bulk Rename Utility for or within a business entity, company or commercial environment and/or for commercial purposes, or within government departments or agencies, a commercial license is required for each computer where Bulk Rename Utility is installed.

Academic Use: The Bulk Rename Utility academic license allows educational institutions and their members (students, faculty, and researchers) to use the software for non-commercial, academic, and research purposes. This license is restricted to faculty, staff, and students of educational institutions. The Software may not be used for any commercial purpose under this license. One academic license is required for each computer where Bulk Rename Utility is installed. The academic licenses can be purchased from our website: http://bulkrenameutility.co.uk

Home Use: Bulk Rename Utility is free of charge for personal, private use at home. To unblock *all the features of the software* and to support software development, a home license can be purchased from our website. A home license unblocks all the features of the software and covers personal, private use at home for non-commercial purposes.

The Bulk Rename Utility licenses can be purchased from our website, http://bulkrenameutility.co.uk, and are delivered instantly via e-mail.

Copyrights and Third Party Licenses

Bulk Rename Utility License
Copyright 2001-2025 - TGRMN Software.
All rights reserved.

Bulk Rename Utility is free of charge for personal, private use, at home.
To use Bulk Rename Utility for or within a business entity, company or commercial environment and/or for commercial purposes, or within government departments or agencies, a commercial license is required for each computer where Bulk Rename Utility is installed. The commercial licenses can be purchased from our website http://bulkrenameutility.co.uk

One commercial license is required for each computer where Bulk Rename Utility is installed. You may access Bulk Rename Utility through a network, provided that you have obtained individual licenses for the software covering all the computers that will access the software through the network. For instance, if 10 different computers will access Bulk Rename Utility on the network, each computer must have its own commercial license, regardless of whether they use Bulk Rename Utility at different times or concurrently.

Academic Use: the Bulk Rename Utility academic license allows educational institutions and their members (students, faculty, and researchers) to use the software for non-commercial, academic, and research purposes. This license is restricted to faculty, staff, and students of educational institutions. The Software may not be used for any commercial purpose under this license. One academic license is required for each computer where Bulk Rename Utility is installed. The academic licenses can be purchased from our website: http://bulkrenameutility.co.uk

Home Use: Bulk Rename Utility is free of charge for personal, private use at home.
To unblock all the features of the software and to support software development, a home license can be purchased from our website. A home license unblocks all the features of the software and covers personal, private use at home for non-commercial purposes.

THIS SOFTWARE IS PROVIDED BY TGRMN SOFTRWARE "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TGRMN SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The Bulk Rename Utility code makes use of certain software components and these are their copyrights / licenses:

Javascript V8 License
Copyright 2014, the V8 project authors. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- * Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- * Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- * Neither the name of Google Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License for Scintilla and SciTE
Copyright 1998-2002 by Neil Hodgson <neilh@scintilla.org>
All Rights Reserved.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation.

NEIL HODGSON DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL NEIL HODGSON BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES

WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

PCRE License - Perl Compatible Regular Expressions
Written by: Philip Hazel
Email local part: ph10
Email domain: cam.ac.uk
University of Cambridge Computing Service, Cambridge, England.
Copyright (c) 1997-2015 University of Cambridge All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of the University of Cambridge nor the names of any contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Index