Table of Contents

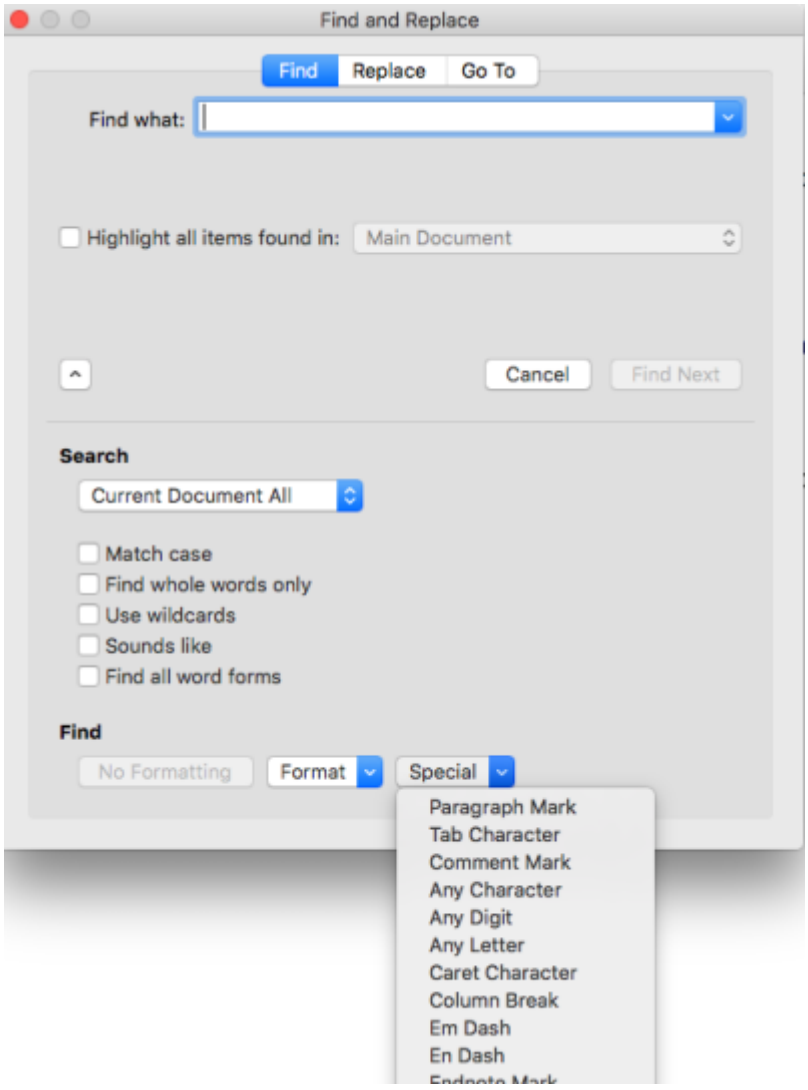
Regular Expressions

Regular expressions (aka regex) is useful for replacing patterns of text, such as headers/footers with page breaks or simply removing them, or replacing line breaks as is common when text is converted from a PDF (to remove middle of word or middle of sentence breaks).

Tips

Using wildcards in Microsoft Word (this is similar to regular expressions, but Word has a lot of its own syntax)

- Word has a lot of options to find letters (^$) and numbers (^#) but these only work with the wildcard option *off* (which it is by default). Only turn the wildcard option on if you're using regex options. Read the info page carefully on when things apply with the wildcard option on/off.
- A lot of the codes for special characters (e.g. page break) are under the "Special..." button.



In LibreOffice & OpenOffice

Make sure that the `Regular expressions` box is checked on the Alternative Find & Replace dialog for all of the search and replace actions below.

Regular expressions in LibreOffice Regular Expressions in OpenOffice

Conversion Fixes

The following fixes assume you are using Word, unless otherwise stated.

Contribute your problems and regex solutions below. Attach your screenshots of both the problem and solution.

PROBLEM: Each line ends with a paragraph break.

SOLUTION: There is no single solution to this, but the typical pattern is to search for the pattern not a period, followed by paragraph break, followed by letter and replace with the same thing minus the paragraph break.

In Word, this will only work with wildcards turned on.

Find: `([A-z] )^13([A-z])`

Replace with: `\1\2`

This looks for the pattern: any-letter space paragraph-break any-letter

The parentheses are used to group what it finds, so \1 refers to the first "any-letter" group and \2 refers to the second "any-letter" group.

In this way, you are putting back exactly what it found minus the paragraph break.

PROBLEM: Hyphenated words that break over two lines.

SOLUTION: Replace with the same text minus the hyphen.

Find: `([a-z])-^13([a-z])`

Replace with: `\1\2`

Using a-z restricts what it finds to lowercase.

You will likely have to do it again for lines that end with a comma, and possibly en and em dash. Look

through your document for patterns of anything else it might have missed.

PROBLEM: There are extra paragraph breaks. We want to keep the real paragraph breaks and remove the fake extra paragraph breaks.

SOLUTION: Use MS Word's find and replace to remove the extra paragraph breaks using special Word symbols.

Find: ^p^p (you can also search for more than 2 paragraph breaks, i.e. ^p^p^p)

Replace with: ^p

PROBLEM: There are newlines/line breaks (↵) instead of paragraph marks (¶).

SOLUTION: Find and remove all line breaks and replace with a single paragraph break.

Find: ^m

Replace with: ^p

In LibreOffice, replace all \n with \p to convert them to paragraphs.

If you understand the deleted notes below, please attach a screenshot of the problem and of the solution!

~~Check to see if there is a paragraph marker at the end of each line, if so, there is a multi-step process to clean them up: - Paragraphs will be separated by a blank line. replace those with a unique set of characters that won't be in the text, e.g. \p\p → - If the lines all end with a space, replace all \p with nothing, otherwise replace them with a single space. - Finally, replace all with \p. * If the lines wrap properly but there is still a blank line between paragraphs, then a simple replace \p\p with \p will suffice, rather than the above procedure.~~

~~We have to convert the double paragraphs breaks into something else unique, remove the single paragraph breaks and then convert the unique characters that were double paragraph breaks into new single paragraph breaks. It is best to do this at the beginning of the text correction stage as it appears to mess with existing formatting styles. - Find and replace all double paragraphs * initiate a find for, ^p^p - Replace with a unique symbol or code, eg, ' xswedc ' * (I found placing a space before and after helps make it even more unique and avoid it bunching up with other double paragraphs) this isn't anything special about these letters, other than that they are a unique string of letters we can search on later - Find and replace all remaining single paragraphs, find = ^p, replace = [single keyboard space] - Find and replace all the double paragraphs you previously changed into a special symbol or code and change back to a single paragraph - Find and remove all line breaks, change into double or single paragraphs instead (find = ^m, replace = ^p)~~

PROBLEM: Running headers. Example, where the first three numbers and the three numbers after the filename is the page number: `231(paragraph break)MacG_9781770494220_5p_all_r1.indd 231(paragraph break)10/27/14 11:56 AM(paragraph break)`

SOLUTION: Without using wildcards:

Find: `^#^#^#^pMacG_9781770494220_5p_all_r1.indd ^#^#^#^p10/27/14 11:56 AM^p`

Replace with: nothing. If you're doing a paginated title, replace with page breaks.

You will need to remove one of the `^#` at the beginning and after the .indd to remove it for 2 digit page numbers, and one last time for single digit page numbers. The following screenshot is an example with a 1-digit page number (see below), followed by the command used to isolate all such instances.

Find: ^#^pMacG_9781770494220_5p_all_r1.indd ^#^p10/27/14 11:56 AM^p

You will also need to do it with the leading `^#^p` to catch the footer text that do not have any page numbers with it.

In LibreOffice:

- Verso (left hand)
- `\p[0-9OoIil]{1,3}\s+.+\p`
    - taken piece-by-piece, this means:
    - `\p` : a paragraph marker
    - `[0-9OoIil]{1,3}` : between one and three numbers or "number like" symbols. (OCR programs often mistake `o` or `O` for `0` and `I`, `i`, or `l` for `1`.)
    - `\s+` : one or more whitespace character (spaces, tabs, etc.)
    - `.+` : one or more of any character
    - `\p` : a final paragraph marker
- Recto (right hand)
- `\p.+\s+[0-9OoIil]{1,3}\p ### Detect bad line breaks ###`
- `[^\."?!]$`