

Use find and replace

In some cases, you can use find and replace to delete items, and avoid using `repeat` loops and backwards iteration.

This macro deletes empty paragraphs from a document. It also has extra code to remove empty lines from within and around tables in the document, which does require using backwards `repeat` loop iterations by -1.

There is a problem: this macro does not actually work in Microsoft Word 2004 for Mac. This occasionally happens when you use macros written for Microsoft Word for Windows, so let's try to deal with that, too.

In Word for Windows, you can work around the fact that `^p` — a special character combination that means "paragraph mark" — is not allowed in the **Find** box when you also use wildcards. You want to use wildcards to represent {2}, a wildcard meaning two or more occurrences of the paragraph mark. In Word for Windows, in the **Find** box, you simply type `^13` instead of `^p` in the in the expression `^13{2,}`, and then select **Use wildcards**.

But that doesn't work in Word 2004, neither in the UI, nor in Visual Basic for Applications (VBA), nor in AppleScript. You can use `^13` without wildcards, but not `^13{2,}` with wildcards. If you do, in the UI, you receive an error message that you are using an expression that is not valid. In a script, you receive an error message that the find does not understand the execute find message, which is actually the same error. On a Macintosh, `^13` means the same as `^p` and is subject to the same constraints.

The solution is to look for two paragraph marks together, `^p^p`, and repeat the **Replace All** until there are no more double paragraph marks. Using find and replace is still so much faster than looping through all the paragraphs that this is a reasonable workaround. Below is the VBA macro and the modified AppleScript version. Compare the following examples:

VBA

```
'Note that using Find and Replace is dramatically faster
'than cycling through the Paragraphs collection

'Replace: ^13{2,} with ^p, which replaces all occurrences
'of two or more consecutive paragraph marks with one paragraph mark

With ActiveDocument.Range.Find
    .Text = "^13{2,}"
    .Replacement.Text = "^p"
    .Forward = True
    .Wrap = wdFindContinue
    .Format = False
    .MatchCase = False
    .MatchWholeWord = False
    .MatchAllWordForms = False
    .MatchSoundsLike = False
    .MatchWildcards = True
    .Execute Replace:=wdReplaceAll
End With

'However, you can't use Find & Replace to delete the first or last
'paragraph in the document, if they are empty. To delete them:

Dim MyRange As Range
Set MyRange = ActiveDocument.Paragraphs(1).Range
If MyRange.Text = vbCr Then MyRange.Delete

Set MyRange = ActiveDocument.Paragraphs.Last.Range
If MyRange.Text = vbCr Then MyRange.Delete
```

AppleScript

```
tell application "Microsoft Word"
    -- replace ^p^p with ^p to replace all occurrences of two
    -- consecutive paragraph marks with one paragraph mark
    -- repeat until done

    repeat
        set textObject to (text object of active document)
        -- redo each time
        if (content of textObject)
            does not contain (return & return) then
                exit repeat -- done
        end if

        set findObject to (find object of textObject)
        -- we need a separate executefind on it,
        -- so best set a variable to it so we just get
        -- it once
        tell findObject
            set {content, content of its replacement,
                forward, wrap, format, match case,
                match whole word,
                match all word forms, match sounds like,
                match wildcards} to {"^p^p", "^p", true,
                    find continue, false, false, false,
                    false, false, false}
        end tell
        execute find findObject replace replace all

        -- Find/Replace cannot delete first or last
        -- paragraph if empty, so:
```

```

        set myRange to text object of paragraph 1
        of active document
        if content of myRange = return then
            delete myRange
    try
        set myRange to text object
        of paragraph -1 of active document
        if content of myRange = return then
            delete myRange
    end try
end repeat
end tell

```

Note that when replacing, you need the special ^p character to insert a proper paragraph end. Using ^13 inserts some sort of corrupted version, though it looks all right at first glance. It might be just a carriage return without all of the extra info that is stored in a Word paragraph mark.

You use a plain `repeat` block because you do not know how many times you'll need to run it. If you use the following statement or a variable set to the same thing, it does not reevaluate each time and the `repeat` loop never finishes:

```
repeat while (get text object of active document does not contain return & return)
```

Instead, reset the `textObject` variable just inside the `repeat` loop to force a reevaluation, and then exit `repeat` under that condition.

That means that you also need to run the code inside the loop that deletes the first and last paragraphs if they are empty. Otherwise, if the last paragraph is empty, the text range continues to have two paragraph marks at the very end, and you never get out of the `repeat` loop.

Finally, you need a `try/end try` block around the code that deletes the last empty paragraph since it errors when it tries to delete `myRange`. The code actually does delete the paragraph (just a carriage return), but still errors. So a simple `try/error` gets around that.

To use the above script to get find and replace to work on selected text, rather than on a whole document, use the following statement in place of `text object of active document`:

```
set textObject to (text object of selection) -- redo each time
```