

17 Editing – track changes

Track change on/off

(This macro gets a mention in video: youtu.be/_ijsRqUR1fE)

If you prefer to use a keystroke to switch track changes on and off, here's a simple macro that does it.

```
Sub TrackOnOff()
```

In Word 2010/2011 and later (and 2007?) there is no visible indication of whether track changes are off or on (well, not unless you happen to have the Review ribbon showing), so you might value some audible feedback to show whether track changes are being switched on or off. This macro beeps once if track changes are now off, and twice if they are now on.

```
Sub TrackOnOffAudible()
```

Don't track change this edit

Sometimes, I want to make a change that I don't want tracked, but all too often I forget to turn track changes back on again. So I thought it might be good to be able to delete and/or replace some text and have the track changes automatically turned off and then turned back on again.

This macro prompts you to type in some text to be inserted at the cursor. However, if some text is already selected when you run the macro, it first deletes the selected text before adding your new text. If you just want to delete some text, select it, run the macro and, when prompted to enter some text, just press Enter.

```
Sub ChangeNotTracked()
```

Accept or reject track change

(Video: <https://youtu.be/AGyrZbgoTD0>)

Yes, I know that Shift-Alt-A and Shift-Alt-Z allow you to accept or reject the selected track change, but I have these macros on Alt-A and Alt-Z, which I find easier, but also they use a little bit of intelligence.

If, when you run the macro, no text is selected then it selects the current line before accepting (rejecting) any track change(s) that it finds therein.

Also called Sub AcceptTracking(), Sub RejectTracking().

```
Sub TrackChangeAccept()
```

```
Sub TrackChangeReject()
```

Reject track change on specific word/phrase

If you have changed, say, the name of a company incorrectly, and you want to revert to the original name, then this macro searches for a given text (the company name), and undoes the track change.

```
Sub UndoSpecificTrack()
```

Track change markup showing or not

If you want to use a keystroke to make the track changes visible – or not – here’s a simple macro that switches them on and off.

```
Sub ShowHideAllTracking()
```

However, this shows/hides the track change marks *and* the comments. If you want to keep the comments visible but hide just the tracking, use the following macro.

```
Sub ShowHideTracksOnly()
```

Aspects of track change markup showing or not

(For Word 2016 or later there’s a fourth option – see below.)

The other way of doing it is to have a macro that switches between three different states:

- 1) Comments only
- 2) Nothing at all
- 3) Comments and insertions and deletions and formatting changes

So click, click, click with the same keystroke and it goes between the three states.

```
Sub TrackingShowSwitcher()
```

Even better, perhaps, is a four-way switch:

- 1) Comments and insertions and deletions (but not formatting changes)
- 2) Comments only
- 3) Nothing at all
- 4) The whole bang shoot

```
Sub TrackingShowSwitcherExtreme()
```

Yet another way of doing it..

I’ve also done another one, prompted by the fact that in Word 2010/2011, every time you reload a file, all the blankety-blank track changes are switched on, regardless of how you had them set when you saved the file.

It sets the showing and not showing options to whatever your taste is (i.e. you alter the True/False items in the macro). However, if you run it repeatedly, it also flips back and forth between Final and Final Showing Markup, so the one macro is both a track changes show/hide macro and also a macro that sets the various track change options on and off.

The three lines I’ve highlighted in yellow in the macro set whether this macro sets any or all of the comments, insertions/deletes and format changes showing. Set to true/false as required.

```
Sub TrackChangeShowHide()
```

With Word 2016 or later, this macro cycles between ‘Simple Markup’, ‘Full Markup’, ‘No Markup’ and ‘Original’. To make this one work, you need to put a ‘Private variable’ at the very top of the VBA window. So, paste in the new macro and then do a Ctrl-Home (or Cmd-Home on Mac) to go to the top of the VBA window and type (or paste):

```
Private pbMarkupMode As Integer
```

However, if you only want to loop between ‘Simple Markup’, ‘Full Markup’ and ‘No Markup’, not have Original as one of the options then you can change the line:

```
includeOriginal = True
```

to False.

```
Sub TrackChangeDisplaySelect()
```

With Word 2016 or later, this macro cycles **inline** markup and **in-balloon** markup.

```
Sub MarkupModeSwitch()
```

This final macro cycles EITHER between **simple** and **full** markup OR between **no** markup, **simple** markup and **full** markup, as set by the line: `includeNoMarkup = True` (or `False`).

```
Sub MarkupDisplaySwitch()
```

Track changes reminder

(Video: youtu.be/UMFSWIsW0Rc)

(Video [more recent]: youtu.be/P-6VdmT2BbE)

You're working with track changes switched on, but you want to make a small change that you don't want tracked. Fine, but it's all too easy to forget to switch track changes back on, and edit away merrily. Then you realise your mistake! So how about some sort of visual reminder?

This macro switches track changes on and off, alternately. And, at the same time, it flips the display into some obviously abnormal mode. This first version switches the text to embossed, but you can choose any font feature that is not used in the text.

```
Sub VisibleTrackOff()
```

For example, if you're not using underline in the document, you could use version 2. To tone it down even further, you can change `wdColorBlue` to `wdColorLightBlue`; or lighter still are `wdColorSkyBlue`, and then `wdColorPaleBlue`.

```
Sub VisibleTrackOff2()
```

And here's another one, sent in by Thiers Halliwell. Thiers found that, with large files, changing the font of the whole text was stretching the processor power of the computer a bit. We think that this one should take less processor time. It puts a wiggly border on the right-hand side of the whole text – which should be easily visible. (N.B. You obviously can't use this if your document uses borders for another purpose.)

```
Sub VisibleTrackOff3()
```

And here's yet another one, from an idea by Bruce Goatly. If you switch track changes off, it changes the document background colour to yellow – very clear, and almost instantaneous. I tried it on a 330,000-word file, and it's faster than Thiers' wiggly line.

```
Sub VisibleTrackOff4()
```

And if you want to fiddle with the colours more, here's one where you can specify the RGB values.

```
Sub VisibleTrackOff5()
```

I can't remember why I produced this next macro, but it's a version of `VisibleTrackOff4`, adding audible feedback as the change is made, but I think there was a document I was working on where, in some parts of the document, the background colour didn't change. So this was a modification to be sure that the background colour was clearly visible throughout the document.

And as with `VisibleTrackOff5`, it's easier to change the background colour by changing the line:

```
myColNumber = 200
```

Increase and decrease the number to taste.

```
Sub TrackOnOffVisible()
```

And there's another one for those **Macs** that don't seem to like some of the commands in the normal VBA that PCs use:

```
Sub TrackOnOffVisibleMac()
```

Turn the background colour of the screen on and off

Someone has asked for a macro to turn the background colour of the screen on and off anyway – without being related to the track changes. So here it is, in her very tasteful 'linen' colour:

```
Sub BackgroundColourOnOff()
```

Accept all formatting track changes

When using track changes, you sometimes want to keep the insertions/deletions for the author to see, but accept all the formatting track changes. This is a particularly useful macro if you've got one of those files where, every so often, when you edit something, it decides to generate track changes showing you where all sorts of formatting is switched on and off. Then this macro is really useful!

It hides all tracking *except* the format changes, then it accepts 'all changes', but that means 'all the changes **shown**' (i.e. just the formatting changes), and then it puts things back the way they were.

Many thanks to Jessica Weissman for showing me this idea – it's much, much faster than the one I used to use.

```
Sub AcceptTrackedFormatChanges()
```

Accept specific types of track changes

When using track changes, you sometimes get swamped by unwanted ones such as 'Formatted: Not Highlight' or 'Formatted: Bullets and Numbering' – a real pain, but a pain no more! Simply click on one of the track changes you want to accept and run this macro; it and all identical track changes will be accepted.

```
Sub AcceptSpecificTrackChange()
```

Accept/reject track changes

(Video: <https://youtu.be/hqPVJSZsFDk>)

These macros allow you to accept/reject track changes. They work on any text that is selected, but if no text is selected, they work just on the individual line of text.

```
Sub AcceptTracking()
```

```
Sub RejectTracking()
```

Accept track change of specific author/editor

If you have a document with changes by different people, maybe an author and an editor, you might want to accept the changes of one, but keep the other(s) as still tracked.

(If you also want to delete their comments, you can use DeleteCommentsSelectively.)

```
Sub AcceptTrackingSelectively ()
```

```
Sub DeleteCommentsSelectively ()
```

Accept tracking and move on to the next

If you place the cursor in the middle of a track change, or you select a track change and run the macro, it accepts the change and moves on to the next.

This might seem a little bit of overkill in terms of macro use but if you apply a keyboard shortcut and have a lot of comments to accept and need to look at each one individually, then this macro will definitely speed up your work and you will quickly learn whatever keystroke you assign with all the repetition.

If you select an area of text you select doesn't contain a tracked item, or if the file doesn't have any more tracked items to move on to it will beep at you. (It will also beep if you put the cursor in a comment.)

```
Sub TrackAcceptMoveOn ()
```

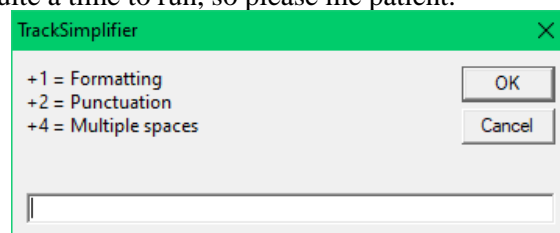
Simplify track changes

(Video: <https://youtu.be/AGyrZbgoTD0>)

(Mac users: Unfortunately, this macro uses certain features of Visual Basic that are not available on some versions of Word for Mac. Nothing I can do about that, sorry.)

If your document has too many track changes, it can become difficult to see the really important ones. This macro allows you to choose to accept the tracking of (a) format changes, and/or (b) punctuation changes, and/or (c) multiple space changes.

On a large document, it can take quite a time to run, so please me patient.



When you run the macro, you enter a number, made up by adding together 1 and/or 2 and/or 4 (you do the maths!), to determine which of the clean-ups you want to run.

If you use this macro, you can leave track changes on the whole time you are editing, and then simplify the tracking later, before sending the file back to the client.

Indeed, if you have a client who absolutely insists you must show ALL track changes, you can educate them by sending a copy of the file with ALL the changes tracked, plus one with the simplified track changes, then they can see how much easier it is for them to see the **important** changes you have made.

```
Sub TrackSimplifier()
```

Consolidate track changes

(Video: <https://youtu.be/AGyrZbgoTD0> + youtu.be/_ijsRqURIfE + <https://youtu.be/hqPVJSZsFDk>)

If you're making two or three edits to a phrase or sentence, the result is a bit of a jumble of tracking, so it's difficult for the author to see what has been changed to what. I therefore have a macro that consolidates the tracking, making it one delete and one insertion. So you select the section of text, and run this macro. OK, all the macro does is to delete the selected text and type it back in again, but this macro means I don't have to think, just press (in my case) Ctrl-Shift-Alt-C.

If you don't actually select anything, but just click in a word, it will select that word for deletion and repasting.

Sub ConsolidateTracking()

Navigating track changes

When you want to jump from track change to track change, use *NextChange* and *PreviousChange* (see above, under the heading: Jump from Edit to Edit).

Compare documents by using track changes

(Video: youtu.be/5QxjCuQjdI and <https://youtu.be/EaFJuKTbhGw> and <https://youtu.be/AGyrZbgoTD0>)

(**Mac users:** Unfortunately, this macro uses certain features of Visual Basic that are not available on some versions of Word for Mac. Nothing I can do about that, sorry. You'll just have to try it and see if it works for you.)

The bad news:

If you want to see the changes between two documents, you can use Word's Compare function off the Review tab on the ribbon. But, by default, this shows you every single change including, for example, multiple word spacing and multiple line spacing and tabs (so-called 'white space'). These track changes can completely obscure the more significant changes.

The good news:

Word's Compare window has a 'More' button; if you click it, you can choose, say, *not* to show the changes in white space.

You can also select the 'granularity' – do you want to show changes character by character, or will word by word do? For example, if 'practice' has been changed to 'practise', it could track it as:

Deleted: c

Inserted: s

or it could show:

Deleted: practice

Inserted: practise

The even better news:

If the comparison is done using a macro, you can decide, by using the macro's options, whether or not certain types of changes are shown. So once you've set these options how you want them, you can just load the files and run the macro.

What's more, the macro allows you not just to compare *whole* files but to apply a selection to either or both of the files, and the macro will compare the text that is selected (or the whole text) of one file with the text that is selected (or the whole text) of the other file.

When you run the macro, it creates a menu of all the open files (although any files such as zzSwitchList or zzFReditList, i.e. names starting 'zz', will be ignored) and presents them as a numbered list:

- 1 – Notes for the author
- 2 – Chapter 03_PB_06

3 – Chapter 03
4 – Hyphenation list

It asks which files you want to compare, so you can enter, say, 32, which means take file 3 as the original, and show the changes between that and file 2.

Press Enter and the comparison appears – job done!

The options available are set at the beginning of the macro (by moving the apostrophe from the `True` line to the `False` line, or vice versa):

```
' checkFormatting = True  
checkFormatting = False
```

i.e. do you want to show formatting changes, such as bold/italic and style changes?

```
' checkSpaces = True  
checkSpaces = False
```

i.e. do you want to show the changes to 'white space', i.e. word spacing, line spacing and tabs?

```
checkCase = True  
' checkCase = False
```

i.e. do you want to show changes made to capitalisation?

```
checkTables = True  
' checkTables = False
```

i.e. do you want to check the tables, or is it OK to just ignore them? (I think this means whether you ignore rows and columns having been added or deleted.)

```
' showMoves = False  
showMoves = True
```

i.e. do you want the moves shown as moves (in green) or as deleted and retyped text?

```
showDetail = wdGranularityWordLevel  
' showDetail = wdGranularityCharLevel
```

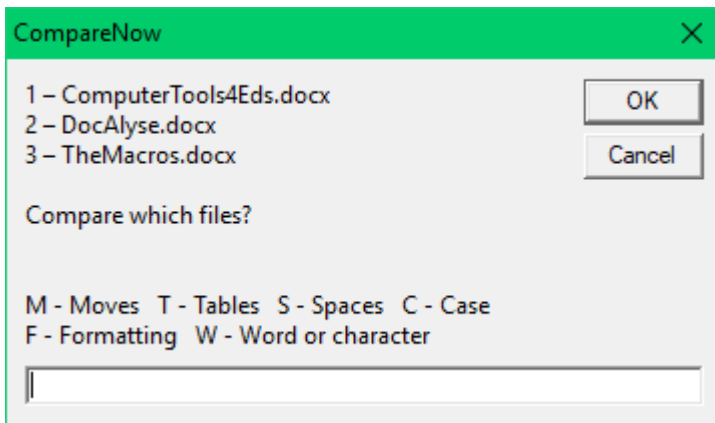
i.e. what granularity do you want? This sets the way the tracking works when you make changes to a part(s) of a word. Here's an example:

For example:

Character granularity: I feel particularly happy today

Word granularity: I feel particularly happy today

So that's how you set the options you prefer, but when you enter your file choice, you can change any one (or more) of them by using codes:



So you can follow your '21' by any or all of those following letters, it will change to the *opposite* of however the options are set at the top of the macro.

Compare parts of the same document

This macro needs to compare two different documents, but someone asked if it was possible to compare two sections of the **same** document. No, but how about this idea?

Load your test file; do a Save As, and call the new file 'aaTempFile'.

Delete all the text in your aaTempFile and resave it. This can now be used any time you want to compare any two parts of your test file.

Test process:

1. Make sure that both your test file and your aaTempFile are open on screen.
2. Select the first section of text in your test file and run *CopySelectedForCompare*. This will copy the selected text, and paste it into your aaTempFile, having first deleted what was in that file from any earlier comparison.
3. Select the second section of text in your test file and run *CompareNow*.

Complicated to explain, but if you do this a lot, it's a real time-saver.

```
Sub CompareNow()
```

Using find and replace despite track changes being present

(Video: youtu.be/AqREu_iJ2Yg)

Problem: Suppose you have 'colour' and have deleted the 'u'. It will look like this: color.

Now, if you try to find that word 'color' using Word's Find and Replace function (Ctrl-H), it won't find it (though it will find the two occurrences of 'color' in this sentence). This is because, in a sense, the deleted 'u' is still there.

And even if you hide track changes, it still can't do this find and replace.

It's no problem if you only want to *find* the word 'color' because both my SmartFinder macro and Word's own Navigation pane, are able to find it. You just won't be able to *replace* it. Nor will you, for example, be able to highlight every occurrence of 'color'.

Solution: I've created two macros that, like my SmartFinder, can find (and replace) things despite the presence of track changes. There are two macros, one for global and one for selective (though you'll see that they are actually the same macro, just with the initial line changed!).

As it stands, these two macros allow you to do a replace and also to add italic, or bold, or highlight.

To use them, use Ctrl-H to set up the F&R and click Find Next, and it will say that it can't find any. So then run the relevant macro. It uses the Find and the Replace, and it notes whether you've asked for bold and/or italic and/or highlight to be added.

```
Sub FandRdespiteTCsGlobal()
```

```
Sub FandRdespiteTCsSelective()
```

Count the number (of words) of track changes in a document

(Video: <https://youtu.be/gOBpOMbIogU>)

This macro... er... counts the number of track changes in a document! And also the number of words (and characters) added and deleted.

```
Sub TrackChangeCounter()
```

Create a file of all sentences containing any tracking

This macro, er, creates a file of all sentences in a document that contain any tracking, including in endnotes and/or footnotes. It was created because an editor had to do a re-edit of a document, and wanted to know how much of the text contained edits, i.e. how many words.

```
Sub CopyAllEditedSentences()
```

Lists the date and time of all track changes

(Video: <https://youtu.be/gOBpOMbIogU>)

This macro, er, lists the date and time of all the track changes in a document!

Its genesis was when an editor realised she had forgotten to switch her timer on for the editing she had just done.

Bother! I know what time it is now, but I can't remember what time I started, or what time I had a break or how long that break was – roughly 15 minutes, I guess.

Hang on! All the changes were being tracked **and timed** as I worked. *Paul!* Please could you please create a list of all the times of all the tracking in this document?

No problem! It's a very simple macro. It creates a new file into which it puts all the track dates and times, in the order they occur in the document; then it creates a copy of that list in another new file and sorts it into order. Search through that and you can see the time period where there **are** tracks being made, and where there **aren't**. Here's a sample:

```
07/10/2019 14:15:00
07/10/2019 14:16:00
07/10/2019 14:16:00
07/10/2019 14:16:00
07/10/2019 14:17:00
07/10/2019 14:17:00
07/10/2019 14:17:00
07/10/2019 14:18:00
07/10/2019 15:07:00
07/10/2019 15:09:00
```

07/10/2019 15:09:00
07/10/2019 15:09:00
07/10/2019 15:25:00
07/10/2019 15:25:00

Sub TrackDateTimeList()