

WPX4 - Some cautions and tips when using the Redaction tool

© 2008 by Barry MacDonnell, <http://wptoolbox.com>. All Rights Reserved. [Thanks to Roy ("lemoto") Lewis for his suggestions.]

The following information pertains to the **WordPerfect X4 initial release**. Future WPX4 service packs *may* address the issues described here.

"WordPerfect lets you create a redacted version of a document in which sensitive or confidential information is replaced by a black bar. The replaced text cannot be retrieved or revealed in the redacted version of the document. You can use redaction if, for reasons of privacy or security, you need to hide some text in a document that you need to share with others." [Corel Support database; <http://support.corel.com>.]

While the new redaction tool in WPX4 (Tools, Redaction) can be very useful to conceal sensitive material in a document, there are a couple of potential – and serious – problems when using it that you should know about.

One problem involves *using highlighting in the same document along with redaction codes*, and the other problem involves *using manual (or macro) search/replacement operations for highlighting codes or redaction codes*.

These two types of WordPerfect features might exist in the same document when you are creating rough drafts of the document: The new Redaction submenu gives you a couple of choices to "mark up" text that should be removed from the final redacted document; these menu choices will add *paired redaction codes* – visible in Reveal Codes as [Redact:Mark] – in the draft document. The standard Highlight tool also applies paired codes around selected text – visible in Reveal Codes as [Highlight].

Note that the redaction codes will not exist in the finalized redacted document (created with Tools, Redaction, Create Redacted Document). Hence there is no interaction with, or problems searching for, highlighting in a *redacted document*. It is only during the Mark-for-Redaction phase, where [Redact:Mark] codes are being entered in the *draft document*, and where regular highlighting is also desired, that you might have problems – and some of these can carry over to the redacted document.

This is especially worrisome if you share draft versions of a document with others who might want to highlight (or redact) sections of text that contains both highlight and redaction codes.

Problem #1 - Using *highlighting* in the same document along with *redaction codes*

Highlighting (i.e., using Tools, Highlight, On or using the Highlight button on the main toolbar) can interact with Redaction marks (i.e., using Tools, Redaction, Mark for Redaction or Tools, Redaction, Find and Mark) in non-obvious ways.

For example, suppose that while working on a draft of your document you (1) have marked some text to be redacted with the redaction tool (which applies a pair of [Redact:Mark] codes around the text); and then you (2) highlight some adjacent words (or the adjacent words were already highlighted) with the highlight tool; and then you (3) select all these items together as a block; and then you (4) *use one of the following tools*:

Tools, Highlight, On

Result: The selection is highlighted (as intended) but the text marked for redaction returns to normal text – that is, the [Redact:Mark] codes are removed from inside the selection (which was **not** intended).

Tools, Highlight, Remove

Result: The highlighted words in the selection are returned to normal text (as intended) and the text marked for redaction in the selection is also returned to normal text (which was **not** intended).

Tools, Mark for Redaction

Result: The entire selection is marked for redaction (as intended) and the highlighting disappears from the selection (which was **not** intended – but this probably not a major problem since you obviously wanted to redact the entire section anyway).

Not all operations with highlighting and redaction are problematic: If you Mark for Redaction a section of text *inside* an already highlighted block of text, the previously highlighted text is split into two sections of highlighted text with the (now unhighlighted) marked-for-redaction text in the middle. When you use Tools, Redaction, Create Redacted Document, this split survives the conversion to the final redacted document, where the text that was marked for redaction is now blacked out and the redaction codes are removed. So this type of operation works as intended. (*Hint: Keep this in mind.*)

The major Highlight-Redaction problem seems to occur only when some text is marked for redaction *first*, and *then* the section containing these redaction markers is selected and either

(a) highlighting is globally applied over (on top of) the redaction markers in that selected section, or (b) highlighting is globally removed from that entire selected section.

Tips

It appears from this and other observations that the new [Redact:Mark] codes are a subset of regular [Highlight] codes. In fact, opening a marked-for-redaction document in an earlier version of WordPerfect will convert the markers to simple gray highlighting.¹

Therefore: Be extra careful when using regular Highlighting in any document marked (or to be marked) for redaction. If both "marking" features are to be used in the same vicinity, you should take extra care when applying them (use Reveal Codes to see what is happening at those locations). Most likely, you want to retain redaction marks until the document is converted to a redacted version (with Tools, Redaction, Create Redacted Document) or until the redaction marks are removed (with Tools, Redaction, Remove all Marks), so you will not want to apply *highlighting* to a document that already contains redaction marks without understanding these possible interactions and being alert to their effects.

To be safe: Before marking text for redaction, back up the original document (or make a copy of it) to safeguard it, *then* apply any required highlighting, *then* apply redaction marks to the working copy, and *then* create the redacted version with Tools, Redaction, Create Redacted Document. Once the redacted document is created, you can add more highlighting to it since the redacted text and the redaction marking codes have been removed. (The text will have been replaced by black rectangular symbols.)

¹ As noted, if a WPX4 document contains text marked for redaction and the document is saved with the [Redact:Mark] codes intact, and it is then opened in WPX3 or earlier version, the [Redact:Mark] codes are converted to normal [Highlight] codes. These codes have an RGB (red,green,blue) value of 173,176,174.

Roy ("Lemoto") Lewis has observed a related phenomenon: If a document in WPX3 has normal, gray highlighting applied with the same RGB values of 173,176,174 (or even 174,177,175), and the document is then saved and reopened in WPX4, the gray highlights are converted to [Redact:Mark] codes!

These observations seem to indicate that marking text for redaction creates codes that are (internally) a subset (or specific variation) of normal highlight codes – which helps explain the way they interact with each other and why users who want to apply highlighting or conduct search/replacement operations need to take this into account.

Problem #2 - Using manual (or macro) search/replace operations for *highlighting codes* when *redaction codes* are present

A. Manual searches (with Edit, Find and Replace, Match, Codes)

If you use a manual search for [Highlight] codes (e.g., [Highlight On] or [Highlight Off]) the program will find *both* [Highlight] *and* [Redact:Mark] codes, if any exist, during the same search. Find and Replace can't tell them apart, so do not use the global *Replace All* option. Use *Find Next* or *Find Prev* and examine each item found to be sure it isn't a [Redact:Mark] code.

Note that, as of this writing, there are no entries in the Edit, Find and Replace, Match/Replace, Codes list for [Redact:Mark On] and [Redact:Mark Off], so you cannot search for them separately from [Highlight] codes.

The tips in Problem #1 above also apply here – especially if you use Find and Replace to *replace* all [Highlight] codes. Such a process will replace [Redact:Mark] codes, too!

B. Macro searches and/or replacements

If you use a macro Search() and/or Replace() routine for [Highlight] codes, the macro will also find/replace existing [Redact:Mark] codes and (possibly) not proceed further once it comes across one. *In any case, you probably will want to skip past any text marked for redaction and proceed to the next [Highlight] code.*

Note that, as of this writing, there are no special macro redaction codes such as [Redact:Mark On] and [Redact:Mark Off] in the macro codes list, nor does a system variable exist (e.g., something like ?RedactText) that could return a value indicating whether the cursor is inside text marked for redaction.

Tips

Here are several ways to deal with these redaction codes when doing a search or replace with a macro. Assume here that the search or replace routine is for a normal [Highlight] code, and that you need to deal with any [Redact:Mark] codes the macro might find in the document.

[**Important:** Be sure to insert the actual binary code for highlighting between the double quote marks in the SearchString() commands below, using the macro toolbar's Codes button.]

```
// Snippet #1:
SearchString("[Highlight On]")
MatchPositionAfter
SearchNext
// To detect [Redact:Mark] "on" codes:
If(?LeftCode=44539)
    // do something here, such as jump to another code segment
Endif
// end of snippet
```

```
// Snippet #2:
SearchString("[Highlight Off]")
MatchPositionAfter
SearchNext
// To detect [Redact:Mark] "off" codes:
If(?LeftCode=44540)
    // do something here, such as jump to another code segment
Endif
// end of snippet
```

Assume you want to use a macro to find just the next [Redact:Mark] "on" code, here's a snippet from Roy ("lemoto") Lewis:

```
// Snippet #3:
Repeat
Call(FindNext@)
If(?NotFound)
    // do something
Else
    // do something else
EndIf
Until(?NotFound)
// do other things here
Return
//
Label(FindNext@)
SearchString("[Highlight On]")
MatchPositionAfter
OnNotFound(exit)
// To detect [Redact:Mark] "on" codes:
Repeat
    SearchNext
Until(?LeftCode = 44539)
Return
exit:
MessageBox(; "Oh"; "Not Found")
Return
// end of snippet
```

Alternative to the redaction tool

Users with WP9-WPX4+ can use the Redactor macro, found in the Library at <http://wptoolbox.com/library>, to create a *printed* version of a document with "redacted" words removed on the printout. When using the Redactor macro, *text is not removed from the document* (so be careful sharing such a document, even as a PDF); it is only covered over with black (or white) highlighting, which effectively prevents the underlying text from being *printed* on paper. The appearance on the printout is much the same as if you had applied a black marking pen to the "redacted" text (or clipped the text from the paper document). ■