

Cleaning It Up:

Now that you've got the routines "popping up all over the place", you must handle any other things which will prevent **CTools™** C Source Formatter from giving a clean output, which is the next goal of editing. The reason you do this now is because it takes care of a few things which would become compiler errors, and it makes the C source (it's not Pascal any more) easier to read and follow.

These preventers are:

- (1) Last right paren (')') missing from the end of a do-while loop.
- (2) Improper or absent switch components (i.e. no "case " or "break;").
- (3) No end-of-line semicolon (;) before "else" keywords.

For the "do-while" in **CMaster™**, you simply find "while" and using the up/down selection search function to find them. Start at the top of the file, click the down button, if it is a "while" at the end of a do/while loop, hit Command leftArrow, then rightArrow once and type a ')' there. If there are comments on the line, you must manually place the insertion point. Do this until EOF or it wraps back to the first one. If you don't have **CMaster™** use Command G and the left/right arrows as above.

Improper switch components are no "case " before the cases, no "break;" between the cases commas where ":RETURNcase " should go. If you have **KodeKeys**, you've been "saved" from a monotonous, drudgery. Use my KodeKeys macro set for these (does one intermediate comma).

Use Find to find the first occurrence of the word "switch" in your **CTools™** output. In **CMaster™**, get that word into the selection search queue. You can see that the word "case " is missing from the switch components. Put them there.

With my macros, you simply put the insertion point before the first one and type "ctrl[3]" (control key, numeric pad 3).

Move down to where the next "case " goes, type "break;", hit RETURN twice and type "case " again.

With my **KodeKeys** macros, simply put the insertion point where "case " goes and type "ctrl-option-[3]" (control, option, numeric pad 3). It's automagically and correctly updated.

If there are two case possibilities, with a comma, like:

```
keyDown, autoKey:
```

then hit "ctrl-shift-[3]" using the **KodeKeys** macros. It will turn the above into:

```
break;
```

```
case keyDown:  
case autoKey:
```

in one shot!

Otherwise, just type in "case " and a space and replace the comma with a ':' and return and another "case" and space, so that it looks like the above example.

When you get down to the last one, you type "break" after it. The macro for that is "ctrl[2]" alone.

It may be that `break;` does not belong there. Maybe it has `return`, instead. Determine this as you go.

Lastly, find all `else` with **CMaster**'s selection/search function and add a ';' to the end of the previous code line (where they are missing).

Now, if you are in a big merged file, format the source later - after a complete compile and you've split the files back up into their individual .c files. On a single file, run the **CTools™** C Source Formatter now, to sort out the appearance. This editing step will not make the source proper for the compiler, but will make it easier and ready to edit for compilation.

Be sure not to feed the formatter too big a chunk of code to format at one time.