

## Turbo Macro Pro editor commands:

[slark.me/c64](http://slark.me/c64)

Special editor commands are accessed by pressing the 'command key', followed by another key to invoke a specific command. Certain commands (**R**, **P**, **B**) will themselves invoke a submenu of additional commands. Several other commands may prompt for user input such as filenames, memory locations, and so on. The command key in all versions of TA (Turbo Assembler), TAM (Turbo Assembler Macro), and TMP (Turbo Macro Pro) is the **←** key, which is located to the immediate left of the number 1 key.

The list below shows all the commands that are invoked by first pressing the command key, followed by the key in the left column:

- 1** exit-basic: Leaves TMP to a BASIC prompt. TMP can be re-entered immediately after doing this with **SYS 8\*4096**.
- !** view-seq: Prompts for a SEQ filename, then displays the contents of the file on screen (**C** pauses output, RUN/STOP aborts).
- 2** paste-separator: Outputs a 'separator' comment line to the current cursor line, overwriting anything else there.
- 3** assemble: Assembles to the assembly bank, which is bank 0 in all mods supporting extra RAM, or simply directly to C64 main memory on the unexpanded C64 mod.
- #** assemble-to-object: Assembles to the current object bank; only enabled on mods supporting extra RAM.
- 4** print-listing: Prompts for a filename to which a source listing print out will be saved as a SEQ file. This invokes the assembler and the resulting listing shows a side-by-side output of code lines preceded by the object code the line assembles to. Instead of a filename, enter **?** by itself to print directly to a printer (device 4) or **\*** by itself to output to the screen (**C** pauses output, RUN/STOP aborts).
- 5** assemble-to-disk: Prompts for a filename to which assembled code is saved on disk.
- %** assemble-to-slave: Assembles across a cable to a slave machine; only enabled in the X2/R2 mods.
- 6** make-data: Prompts for a region of memory (start and end address). Data in that region are read and translated into **.byte** statements which are inserted into the source code at the current cursor line. In mods supporting extra RAM, the data is read from the current object bank; otherwise the data is read from the C64 main memory.
- 7** set-tab-return: Takes the current cursor location and sets that as the column at which the cursor is placed after pressing return.
- 8** set-tab-source: Takes the current cursor location and sets that as the column at which the source code (opcode/pseudo) is aligned at.
- +** add-hex: Prompts for two 16 bit values, adding them and outputting the result in hex and decimal.
- sub-hex: Prompts for two 16 bit values, subtracting the second from the first and outputting the result in hex and decimal.
- DEL** delete-line: Removes the entire line under cursor, moving subsequent lines up by one.
- INST** toggle-line-insert: Switches the editing mode for line insertion; when off, hitting return does not insert a new blank line.
- ↑** copy-line-buffer: Copies the line under cursor to the line buffer.
- £** paste-line-buffer: Pastes the line buffer to the current cursor line.
- ¶** cursor-left-edge: Move cursor to column 0 (the leftmost column).

**w** write-seq: Prompts for a filename to which the current source is saved to disk as a SEQ file.

**e** enter-seq: Prompts for a filename from which lines of source code are read off disk and inserted at the current cursor line.

**r** replace-string: Prompts for search and replacement strings and then finds the first occurrence of the search string, leaving the cursor at that occurrence.

**R** ram-submenu: Activate the expansion RAM submenu (where applicable), with these additional key commands:

**R 1** load-to-ram: Prompts for a filename. A second prompt is given for a load address, using the file's actual load address as a default. Then the file is loaded into the current object bank at the given address.

**R b** backup-to-bank: Prompts for an available source bank number, and then copies the active source code into that bank.

**R o** set-object-bank: Prompts for an available RAM bank and uses that as the new object bank.

**R s** swap-to-bank: Prompts for an available source bank number; if the selected bank contains a TMP/source instance, then the active source code is backed up to its assigned bank and the selected bank is swapped into C64 main memory, becoming the new active source code. TMP will not allow swapping in a bank that doesn't already have a TMP instance in it.

**R j** set-jumpback: Prompts for an address, which will be the location to which the jumpback routine will be copied when executing an assembled code. If the address is set to \$0000 then TMP will not copy the jumpbacker at all. The default address is \$0140 .

**t** replace-one: Executes one replacement of a just-found search string and then searches for the next occurrence, moving the cursor to it.

**y** replace-all: Executes replacements for every occurrence of the search string located from the current cursor location to the bottom of the source code.

**u** list-labels: Prompts for a filename to which a listing of each label and the memory address that it resolves to will be saved to disk as a SEQ file. Instead of a filename, enter **?** by itself to print directly to a printer (device 4) or **\*** by itself to output to the screen (**C** pauses output, RUN/STOP aborts). Note that the output is based on the label resolutions from the last time the source code was assembled, and if list-labels is invoked prior to assembling any source code, the output will be empty.

**U** list-labels-vice: Identical to list-labels, except the format of the output is compatible for loading into the VICE emulator monitor.

**i** find-label: Prompts for a string to search for as a label, then places the cursor at the first occurrence of that string where it is used as a label definition.

**P** preferences-submenu: Activates the preferences editor submenu which can be used to modify the separator style and color scheme:

**P 0** Edits the color used for \$d020 (border).

**P 1** Edits the color used for \$d021 (background).

**P 2** Edits the color used for the message line.

**P 3** Edits the color used for the status line.

**P 4** Edits the color used for regular source.

**P 5** Edits the color used for source with a syntax error.

**P 6** Edits the color used for source in the current marked block.

**P 7** Edits the separator template.

**P ↑** Prompts for a filename, to which a complete copy of the newly customized TMP program is written to disk.

**C** disk-command: Prompts for a string that is sent to the current serial device as a disk command, the results of which are displayed on the message line. Entering a **C** by itself will send a status command to the device.

**\*** view-directory: Outputs a directory listing from the current serial device to the screen (**C** pauses output, RUN/STOP aborts).

**a** petSCII-mode: Sets the editor into a literal PETSCII mode; the cursor flashes slightly faster to indicate the setting. During this mode, all keypresses are taken literally as PETSCII sequences which are stored on the line. This mode is useful for entering color or cursor movement codes into .text strings that can be 'printed' from within your program. Hitting the **←** key will quit this mode and return to normal editing.

**s** save-source: Prompts for a filename, to which the current source code is saved to disk as a binary PRG file.

**d** increment-device: Selects the next available device number on the bus between 8 and 15.

**f** find-string: Prompts for a string and then searches for the first occurrence in the source code.

**g** goto-mark: Prompts for a mark (**1-9, s, e**) and moves the cursor to the line number corresponding to the selected mark.

**h** find-next: Finds the next occurrence of the search string.

**k** define-fkeys: Prompts for a redefinable function key (**F3-F6**), then allows the user to enter a new definition for the selected key.

**K** reset-fkeys: The function keys (**F3-F6**) are all reset to TMP's internal defaults. This is useful when loading a source code that has a different set of key definitions than TMP.

**l** load-source: Prompts for a filename, from which source code is read. Note, the loaded source code will **\*replace\*** any current source!

**:** list-marks: Displays the current line numbers associated with each mark.

**;** kill-mark: Prompts for a mark (**1-9, s, e**) and disassociates the selected mark with whatever line number it had been pointing to.

**Z** undo-edit: Undo whatever current line editing has occurred. Note: This can only be used as long as the cursor has not left the current line where editing has taken place!

**c** cold-start: This performs a hard reset of TMP, re-initializing and erasing the current source code.

**b** block-submenu: Activates the block commands submenu, with these additional key commands:

**b w** write-block: Prompts for a filename to which the source code within the current marked block is saved to disk as a SEQ file.

**b c** copy-block: Copies the current marked block to the current cursor position.

**b M** move-block: Moves the current marked block to current cursor position.

**b k** kill-block: Erases the current marked block.

**n** goto-line: Prompts for a line number and then moves the cursor to the entered line.

**M** set-mark: Prompts for a mark (**1-9, s, e**) and then sets the selected mark to the current cursor line. **s** and **e** stand for start and end; setting these marks defines a contiguous set of lines upon which subsequent block operations (invoked by the block-submenu) will operate.

**=** join-line: The remainder of the current line (everything to the right and under the current cursor location) is moved into the same location on the line immediately above, deleting the old line in the process.

**/** blank-to-end: Blanks out everything on the current line to the right and under the current cursor location.

- RETURN** split-line: The remainder of the current line (everything to the right and under the current cursor location) is moved into a new line which is inserted immediately below the current line.
- SPACE** blank-line: Blanks out the current line.
- CRSR⇒** cursor-right-edge: Moves the cursor to column 39.
- ⇐CRSR** cursor-left-edge: Moves the cursor to column 0.
- CRSR↓** Moves the cursor down by 200 lines.
- ⇧CRSR** Moves the cursor up by 200 lines
- ←** The command key, when pressed twice in a row, outputs an actual ← character.

The list below shows all the commands that are invoked by pressing the key in the left column (without first pressing the command key):

- INST** toggle-char-insert: Switches the editing mode for character insertion; when off, new key presses overwrite instead of insert.
- F1** Moves the cursor up by 20 lines.
- F2** Moves the cursor to line 0 (top of the source).
- F7** Moves the cursor down by 20 lines.
- F8** Moves the cursor to the bottom of the source.

Function keys **F3-F6** are redefinable, and have these default operations:

- F3** Identical to ← + ⇧**CRSR** (cursor up by 200 lines).
- F4** Assembles (as with ← + **3**) and then auto executes the code.
- F5** Identical to ← + **CRSR↓** (cursor down by 200 lines).
- F6** Identical to ← + **R** which invokes the RAM submenu.