

## TECO-64 – Command Line Options

### Overview

This document describes the command-line options that can be used to start TECO-64. Generally, only one file can be edited at a time, although it is possible to read one file and write another. Also, as noted in other documentation, it is possible to switch between primary and secondary input and output streams.

As TECO processes the user's options, it builds a command string that is processed when initialization has completed. However, this command string is not created until all options have been processed, and it therefore does not matter in what order they occur.

In addition to the options described below, there are also several optional environment variables that may be used to affect processing of these options, or which affect TECO's behavior after initialization.

TECO_INIT	<p>This defines the name and location of a file containing TECO commands which are executed at start-up. This is often used to set desired features, such as enabling window mode on startup.</p> <p>If TECO_INIT translates to a string in single or double quotes, that string is executed as TECO commands at start-up. Note that depending on the shell or command-line interpreter in use, the quotes, and possibly other characters, may need to be escaped so that they are passed to TECO.</p>
TECO_LIBRARY	<p>This defines the location of a library of TECO macros (i.e., <i>.tec</i> files) that is searched by the <b>EI</b> command if the specified indirect file is not found in the local directory. If the file specification includes a relative path, that will be appended to the library directory (so if TECO_LIBRARY translates to <i>lib</i>, and the indirect command file is <i>test/foo</i>, then the EI command will search first in <i>test/foo</i>, and then in <i>lib/test/foo</i>.</p>
TECO_MEMORY	<p>This is used by TECO to determine the name of the last file edited. With some operating systems, this environment variable will contain that name directly; with others (such as Linux, Windows, and MacOS), it contains the name of a file that in turn specifies the last file edited.</p>
TECO_PROMPT	<p>This defines a string used for TECO's prompt (instead of the standard asterisk prompt).</p>

## Basic Invocation

`teco foo`

Open *foo* for input and output (using the **EB** command), then read in first page.

`teco`

Open last file edited, by reading its name from the file pointed to by the *TECO\_MEMORY* environment variable. If that variable is not defined, TECO will start without opening any file.

## Options

`-Ann, --argument=nn`

Store *nn* in Q-register A before starting TECO (using the **UA** command). Normally used in conjunction with the `-execute` option.

`-Bstring, --buffer=string`

Insert *string* into text buffer before TECO starts (using the **I** command). Normally used in conjunction with the `-execute` option.

`-C, --create (default)`

If *foo* does not exist, then create it (using the **EW** command). This is effectively a *make* command with other versions of TECO.

`-c, --nocreate`

If *foo* does not exist, then issue an error and exit.

`-Emungfile, --execute=mungfile`

Execute *mungfile* as an indirect command file (using the **EI** command) which is executed at start-up. This is effectively a *mung* command with other versions of TECO. If file does not exist, TECO will try opening *mungfile.tec*.

If *mungfile* is a string in single or double quotes, then that is treated as a command string instead of a file name. Note that depending on the shell or command-line interpreter in use, the quotes, and possibly other characters, may need to be escaped so that they are passed to TECO.

**-Iinitfile, --initialize=initfile**

Open *initfile* as an indirect command file (using the **EI** command). If file does not exist, TECO will try opening *initfile.tec*.

If *initfile* is a string in single or double quotes, then that is treated as a command string instead of a file name. Note that depending on the shell or command-line interpreter in use, the quotes, and possibly other characters, may need to be escaped so that they are passed to TECO.

**-I, --initialize (default)**

Use the *TECO\_INIT* environment variable to find the name of the initialization file. If *TECO\_INIT* is not defined, then no initialization occurs before TECO starts.

If the value of *TECO\_INIT* is in single or double quotes, then that is treated as a command string instead of a file name.

**-i, --noinitialize**

Do not use any initialization file, and do not check *TECO\_INIT*.

**-Llogfile, --log=logfile**

Open *logfile* as a log file for TECO input and/or output (a flag variable controls whether either or both are logged).

**-M, --memory (default)**

If no file is specified, then use the *TECO\_MEMORY* environment to find the name of a file which contains the name of the last file edited.

**-m, --nomemory**

Do not use *TECO\_MEMORY* to find the name of the last edited file.

**-Obaz, --output=baz**

Open *foo* for input (using the **ER** command), and open *baz* for output (using *EW* command), then read in first page.

**-o, --nooutput (default)**

Use the same name for the output file as the input file (unless **-readonly** is specified).

**-R, --readonly**

Open *foo* for inspection (using the **ER** command), and read in first page.

-r, --noreadonly (default)

Open *foo* for input and output (using the EB command).

-Snn, --scroll=nn

Start TECO in split-screen window mode, with *nn* lines for a scrolling region (using the 1W and nn, 7:W commands).

-W, --window

Start TECO in window mode (using the 1W command).

-X, --exit

Used with -E to exit from TECO once the indirect command file has been processed.

-Z, --zero

Start TECO with strict enforcement of command syntax (using the -1EW command).