

Debug Macros with MCRCHECK

By Niek Campagne (2005, updated 2016)

MCRCHECK

The macros belonging to this package can be used to debug WordPerfect 5.1(+) macros.

Macro *mcrcheck* discovers, within seconds, syntax errors in a macro, in particular – but not only – tilde errors ('List of Detected Errors' below).

Its add-on *mcrindex* generates a list of the varnames and labelnames in a macro, with the line numbers where they occur. The length of varnames in the index is limited to 7, the maximum number of relevant characters in varnames; labelnames are limited to 15 relevant characters.

INSTALLATION

The downloaded file *mcrcheck.zip* contains the WP51 macros MCRCHECK.WWPM, MCRINDEX.WPM, and GOTOLINE.WPM; the Editor macros MCRCHECK.EDM and MCRINDEX.EDM; the Shell 3.x macro MCRINDEX.SHM; and the text files README.WP5 (WordPerfect file format, this file) and QUICKLY.TXT (DOS text, for a quick start).

Copy the macros you plan to use to their respective mmacro directories. This completes the installation.

USE

There are two roughly equivalent versions, *.*edm* and *.*wpm*, of both *mcrcheck* and *mcrindex*.

*.*edm* are Editor macros. (You can find information on how to install ED.EXE on E. Mendelson's most informative website www.wpdos.org.) They are language-independent. (One may run these macros, with some caution, on Editor macros and possibly (not tested) on those of other WPCorp. programs with closely related macro languages.)

*.*wpm*, being WordPerfect 5.1 macros, can be used on decompiled (text-version) macros like those produced by David Seidman's *macascii*, or *macrolst* from Jeffrey Kane's MacroAde (see the www.wpdos.org website). *mcrcheck.wpm* is specific for (the *wp.mrs* and *keys.mrs* of) the English versions of WordPerfect 5.1, and checks for incorrectly spelled English macro and key commands.

If you write your macros in WordPerfect Editor (ED.EXE), your natural choice will be to use the *.*edm* macros, but if you write them as (an English) text document in WordPerfect and use a tool to convert them to the macro format, you may prefer the *.*wpm* macros to check the text version. The instructions for both sets of macros are essentially the same.

mcrcheck:

The location of most errors in a macro (list of detected errors below) is within two or three lines above the position where *mcrcheck* stops upon detection of the error. *mcrcheck* does not itself correct any error but displays an informative message before quitting. It thus offers an opportunity to correct the error immediately after its detection.

With the macro to be checked retrieved, start *mcrcheck*. It will walk, starting from the cursor position, along the macro until it finds either a located error, or a Small House (△), or the macro's end. In every case, it will stop with an informative message (and then quit upon a keystroke). Use Small Houses – Alt+Num-127 – to check sections of a macro, for instance to locate an {END IF} error.

mcrindex:

In WordPerfect, with document 2 cleared, run *mcrindex.wpm* in document 1 with the retrieved macro (already debugged with *mcrcheck*). In WordPerfect, that's all: you'll find the index of varnames, sysvars, labelnames, and macro names in document 2.

Editor does not possess a Sort function. Therefore, the index has to be made in WordPerfect. *mcrindex.edm* saves the data in a text file. Close Editor after running *mcrindex.edm* on a macro, and start WordPerfect. In a cleared document screen now start *mcrindex.wpm* to retrieve the data file and complete the index.

If both WordPerfect and Editor run under Shell 3.x (and *mcrindex.shm* has been properly installed), the entire process runs completed automatically after *mcrindex.edm* has been started. (You may find information on Shell 3.x at the www.wpdos.org website.)

Macro *mcrindex.edm* uses a temporary DOS text file for its collected data. This file is saved in the disk root directory, typically C:\. To minimize the risk of erasing existing files, the filename of this temporary file begins with a '#@&' sequence. The file is deleted immediately after retrieval of its data.

gotoline

The locations mentioned in the index are sequence numbers of macro lines. Editor displays the line number on the status line, but WordPerfect doesn't. To find a line in the latter, run *gotoline.wpm* and enter the line number when prompted.

HOW ERRORS ARE REPORTED

- Most errors are located, and can be pinpointed more or less precisely, like the tilde errors. *mcrcheck* will stop with an informative message as soon as a located error has been detected.
- The other errors depend on counts, and are reported at the macro's end (or at {LABEL} commands).

LIST OF DETECTED ERRORS

'Located' errors reported at, or a short distance beyond, the location where they actually occur

- Missing and redundant delimiter tildes in syntaxes. (Replace a non-syntax tilde occurring somewhere in a macro by a tilde command {~} (Ctrl-v,~) or {NTOK}126~. to avoid a false error).
- Missing or unequal quotation marks – ' or " – delimiting a string. If a quotation mark itself is the 'string', it should be enclosed between quotation marks of the other kind: "" and """"; the enclosed quotation mark will not be considered as a delimiter by *mcrcheck*.

- Unequal numbers of opening and closing brackets. Single brackets between quotation marks are not counted.
- Illegal commands within a macro command syntax (often caused by a tilde error).
- Non-numeral characters where only numerals are allowed (in the syntaxes of {NTOK}, {SPEED}, {WAIT} and {MID}).
- Letters and other illegal characters in numeric expressions, as in a {FOR} syntax.
- No control-character following {^N} or {^O} in a message.
- Less than two (lower) ASCII characters, Ctrl-characters or {NTOK}'s following {^P} in a message.
- In the sysvar following a {SYSTEM} command, a spelling error (other than concerning font case) or a non-English sysvar name. Unlike sysvar names, sysvar numbers are universal.
- With {CASE} and {CASE CALL} (apart from tilde errors):
 - Syntax ends with a value instead of a labelname.
 - Within an {IF} - {END IF} loop, an {ELSE} command is not allowed because it will function as belonging to that {IF} loop. Therefore, it should be replaced by {OTHERWISE}.
 - {OTHERWISE} is not the last value in the {CASE} syntax.
 - Quotation marks, if present, should be placed around both the expression and all values except {OTHERWISE}.
- Absent or wrong option number following {Macro Commands}. Only 3, and a or A(ssign), are allowed; 3 is universally valid.
- {Replace} Yes/No question not answered by y, Y, n, N, +, -. The latter two – '+' and '-' – are universally valid.
- Name search code other than n or =. Character = is universally valid.

'Counted' errors

Reported at next {LABEL}:

- Odd number of {Search} commands (single commands)
- {Replace} lacking one or two {Search} commands

Reported at macro's end or at a Small House:

- Missing and redundant {END IF}, {END FOR} and {END WHILE} codes are calculated by subtracting the total numbers of {IF (EXISTS)}, {FOR (EACH)} and {WHILE} codes, respectively, from their counted numbers.

Faulty command codes

In addition to these errors, *mcrcheck.wpm* reports incomplete and misspelled command codes. Intentional single opening and closing curly braces (except those assigned to a variable, or in a {CASE (CALL)} syntax, or in a 'string' enclosed in quotation marks) therefore will cause false errors, which can be avoided by using {KTON}123~ and {KTON}125~, respectively, instead.

Errors revealed by the list generated by mcrindex

- Single occurrences of varnames and labelnames may point to typing errors and other discrepancies.
- Labelnames in a {LABEL} command are indicated by a bolded 'L' after their line number, permitting the identification of missing and double {LABEL} commands.