

TRS-80 BASIC PROGRAM UTILITY
COMMAND PROCESSOR
"COMPROC"
USERS MANUAL

Written For RACET computers By

T.S. JOHNSTON

and

R.D. JOHNSTON

For Use On The Radio Shack® TRS-80™
Level II BASIC 16-48K Microcomputer System

© COPYRIGHT 1979, RACET computers, Orange, California

IMPORTANT NOTICE

ALL RACET computers programs are distributed on an "AS IS" basis without warranty. Neither RACET computers nor the contributor makes any express or implied warranty of any kind with regard to this program material, including, but not limited to the implied warranties of merchantability and fitness for a particular purpose. Neither RACET computers nor the contributor shall be liable for incidental or consequential damages in connection with or arising out of the furnishing, use or performance of this program material.

© 1979, RACET computers, Orange, California

The Government Law (Title 17 United States Code) has been amended by a recent Act of Congress, Public Law 92-140, protecting certain sound recordings against unauthorized duplication. It is an infringement of this law to copy any properly registered cassette designated with the copyright notice (e.g. © 1979 RACET computers, Orange, California).

TRS-80 BASIC PROGRAM UTILITY

COMMAND PROCESSOR

"COMPROC"

INTRODUCTION

COMPROC is a simple and easy to use system which enables the user to automatically execute a series of TRS-80 DOS system or BASIC commands. This provides the user with a convenient method for performing repetitive operations. Following are some of the key features of COMPROC:

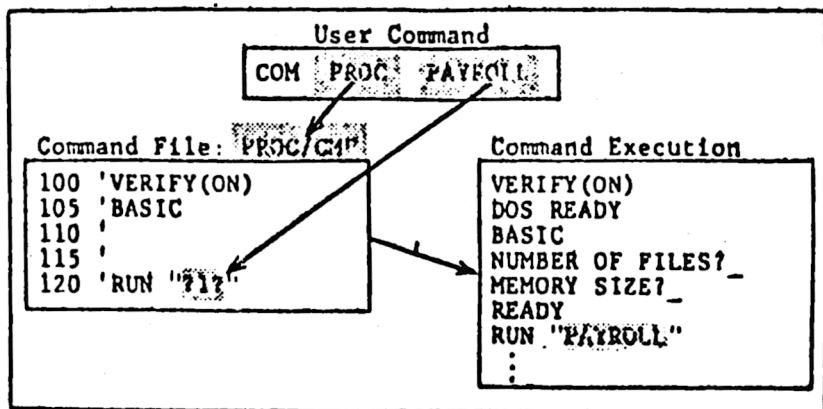
- Command files may be created containing any logical sequence of system commands, BASIC statements, or data.
- Command files may be selectively executed, including initiation by the AUTO command at power-up time.
- Creation, saving, and editing of any command file may be easily accomplished using standard NEW, SAVE, and EDIT operations of BASIC.
- Up to nine fields may be specified at execution time to be inserted at specified locations in the users command sequence.
- COMPROC will prompt the user for missing information at execution time, allowing the user to respond according to current requirements.
- Loading and relocating of the keyboard debounce routine may be optionally specified.
- COMPROC will run in any size TRS-80 DOS system and is automatically relocated so as not to conflict with other BASIC or machine language programs.

These, and many other features of COMPROC allow the user to create procedures that non-computer oriented personnel can easily use.

METHOD OF OPERATION

The following general steps are performed when using the COMPROC system:

- The user creates a command file(CMF) containing the list of commands or data to be processed. A very simple CMF is shown in Figure(1). Note that this is a standard BASIC file containing REM statements(the ' is an abbreviation for REM).
- The user initiates COMPROC by entering a standard DOS command as shown in the example.
- COMPROC scans the CMF, extracts information, and inserts optional fields to form a compressed control program.
- The compressed control program is relocated to a temporarily reserved area and passes the desired information to the system as if the user had entered it from the keyboard.
- After all statements have been processed COMPROC will release any reserved area and return control to the keyboard.



Figure(1). Simple COMPROC Example.

Note that in the above example the user command could be set with the AUTO command as follows:

```
AUTO COM PROC PAYROLL
```

This would execute the above CMF script at power-up time without any other user action required.

LOADING COMPROC TO DISK

COMPROC is distributed on cassette tape which must first be loaded to disk prior to use. The recommended procedure is as follows (user input is shown underlined):

1. Bring in TRSDOS
2. BASIC (user enters BASIC command)
3. HOW MANY FILES? _
4. MEMORY SIZE? _
5. CMD"T" (must turn off clock!)
6. SYSTEM (executes SYSTEM command)
7. ?COM (COMPROC name=COM)
8. ?bk (bk = break key)
9. CMD"S" (go back to DOS)
10. DUMP COM/CMD (START=X'7000',END=X'7700',TRA=X'7000')

The above procedure will load COMPROC into a file named COM/CMD, although any other file name could be used.

EXECUTING COMPROC

COMPROC is a machine language program which is executed in DOS command mode. The general format of the user command statement is shown below (assuming COM/CMD is used as the program file name):

```
COM filespec(ln) parm#1 parm#2 ... parm#n
```

where

filespec The name of the command file(CMF) containing the users script to be executed. The file extension name will default to /CMF if not otherwise specified.

(1n#) optional specification immediately following and enclosed within parenthesis indicating the lowest line number in the CMF to be processed. The first line in the CMF will be used if (1n#) is not present.

parm#1 Optional parameter fields which are to be inserted at
 parm#2 specified locations in the users CMF. Each field must
 : be separated by one or more blanks and contain no
 parm#n blanks within the parm# field.

Trailing fields may be left unspecified on the COM command statement. The COMPROC analysis program will prompt the user for any missing fields at execution time. This allows the user to interactively supply parameters based upon current requirements.

COMMAND FILE DESCRIPTION

A. General Format

The CMF is a BASIC program file containing a sequence of REM and END lines. The information used by COMPROC follows immediately after each REM(or its abbreviation "'"). Trailing blanks are removed and replaced by one carriage return(ENTER-key)-character. The user must anticipate all responses needed to accomplish a given task and include the appropriate lines in the CMF script.

Several scripts may be placed in the same CMF. Each script(except the last) must be terminated with an END line or a ?X? function code. COMPROC will process all lines greater than or equal to the (1n#) option up to the end-of-script or end-of-file, whichever comes first.

B. Parameter Fields

Optional parameter fields may be specified on the COM user command as discussed previously. These fields will be inserted in the users script at locations identified by the sequence:

?n?

where "n" is the relative parameter field number(1,2, ... n). Each occurrence of the ?n? sequence in the CMF is replaced by the corresponding parameter field from the COM user command.

C. Function Codes

The user may place special function codes in the CMF to perform a number of actions. The function code is specified by the sequence:

?ff?

where "f" is a one letter code summarized in Figure(2).

D. Memory Size Specification

The compressed COMPROC control program produced from the users CMF is relocated to a temporarily reserved area just below the highest value available to DOS or BASIC. This is specified by:

1. The first line in the script containing the memory size. This line is then deleted from the script to be executed.
2. The memory size entered in the normal sequence in the script when initiating BASIC.
3. The highest memory location if neither of the above is given.
4. The lowest value if specified more than once.

Code	Description
7B?	<u>Break-key</u> - This enters a break key character in the script. This can be used, for example, to terminate an AUTO command.
7C?	<u>Clear-key</u> - This enters a clear-key character in the script. This performs the same action as if the user had pressed the clear key at that point.
7D?	<u>Down Arrow-key</u> - This enters a down arrow-key(↓) character in the users script.
7I?	<u>Interactive Input</u> - This specifies that COMPROC is to revert temporarily to the keyboard for data input. This mode will be terminated when the user presses the carriage return (ENTER-key). The carriage return character is <u>not</u> passed on to the system. After the users carriage return COMPROC will continue processing any remaining script.
7L?	<u>Left Arrow-key</u> - This enters a left arrow(←) backspace key character in the users script.
7P?	<u>Pause</u> - This enters a pause code into the CMF. COMPROC will temporarily suspend processing until the user presses any key(similar to the shift @ key in BASIC).
7R?	<u>Right Arrow-key</u> - This enters a right arrow(→) tab key character in the users script.
7U?	<u>Up Arrow-key</u> - This enters an up arrow-key(↑) character in the users script.
7X?	<u>Exit</u> - This terminates the current script similar to an END line except no trailing carriage return is generated. This allows the user to continue the last command on the same line(e.g. RUN "7X?").

Figure(2). Function Code Summary

COMPROC will subtract the amount of memory required for the compressed control program from the memory size as determined above. This extra space will then be released by COMPROC when the last line of the users script has been executed. This space will be added to the users string space if BASIC is in control.

E. Key Debounce Routine Specification

An optional routine is available in COMPROC that will help reduce the double striking or "key-bounce" problem on the TRS-80. This routine will be loaded into memory by specifying:

'DEBOUNCE or REMDEBOUNCE

in the first line of the script, or the second line if a memory size is specified on the first line(See D.1 above). This routine will be relocated just below the memory size value determined in D.1-D.4 above.

If 'DEBOUNCE is not specified then any existing keyboard debounce routine is deactivated.

DETAILED EXAMPLE

Figure(3) illustrates a more complex use of COMPROC. Shown at the top is the user command, associated prompting messages, and user input responses. On the left is the CMF containing the statements to be processed by COMPROC. The resulting command execution sequence is shown on the right. User input is shown underlined, and shaded areas represent parameter fields and function codes.

Summarized below is an explanation of the features demonstrated by this example.

1. The user initiated COMPROC by entering only "COM". COMPROC then prompted the user for the missing CMF procedure filespec and other fields needed. The user could have provided all information on the initial command line, ie
COM PROC(200) PAYROLL/BAS GSF
2. The CMF shown contains two scripts. Line #140 indicates the "END" of the first script. The second script continues through Line #400. Note that Line #340 is not an END statement for the second script since it contains a leading REM(').
3. The user specified PROC(200) indicating that processing should start at Line #200 in file PROC/CMF.
4. Lines #200 & #280 both indicate memory size to be reserved for user protected area. The smaller value(62168) is further reduced by a small amount for COMPROC and the DEBOUNCE routine. This value(61822) replaces the corresponding value at Line #280.
5. Line #210 specifies that the keyboard "DEBOUNCE" routine is to be loaded and relocated(just below 62168 in this case). Note that Lines #200-210 are eliminated from the script.
6. Lines #250 & #290 show the use of parameter fields. In this case ?1? is replaced by PAYROLL/BAS and ?2? by GSF from the user command prompt responses.
7. Lines #230, #240, & #360 illustrate the use of the interactive function code ?1?. Upon encountering this function code COMPROC temporarily activates the keyboard to allow the user to continue the sequence. In this case the user enters the time, date, and line # to be "RUN". COMPROC continues with the remaining script after the user presses the ENTER-key.
8. Line #350 contains a break-key function code. This is required to terminate the AUTO mode active in Lines #300-340.
9. Line #320 shows the use of the right arrow(→) function code ?R?. This inserts a tab in the resulting script.
10. Line #360 specifies a pause function code ?P?. The system pauses at this point until the user presses any key.
11. Lines #370-380 illustrate that the users script can also contain data.
12. Line #400 shows the use of the terminate function code ?X?. This is an alternate method of terminating the script which results in no trailing carriage return(ENTER-key). This allows the user to complete the last line from the keyboard. In this example another COM command is entered which executes the first script in the CMF at Lines #100-140.

HINTS AND SUGGESTIONS

The user should read very carefully the following hints and suggestions in order to effectively utilize COMPROC.

1. User written machine language programs may bypass COMPROC when calling the system keyboard routine for data. COMPROC only intercepts calls to 0049H and 0384H (or routines that call these addresses). Any calls to 002BH as documented in the TRS-80 Editor Assembler Manual will bypass COMPROC and expect data to be entered directly from the keyboard.
2. While COMPROC is in control all keyboard keys are inactive except the Break-key. This may be used to terminate COMPROC when it is in control.
3. User programs that utilize maximum memory may return an "OUT OF MEMORY" message when using COMPROC. This is due to the small amount of extra space required by COMPROC. This can usually be avoided by only using COMPROC to issue the RUN statement for the large program. This will release space occupied by COMPROC just prior to the start of execution.
4. The COM user command may be initiated by the AUTO feature. Refer to the DOS Users Manual for details on the use of the AUTO command. Also see the example associated with Figure(1) on Page(2).
5. The last line of the users script may contain another COM user command. This provides a method of linking from one script to another. The system must be in DOS mode when this statement is finally executed. The COM command may not appear anywhere else in the users script.
6. Any user command sequence which would allow the debounce routine to be destroyed, such as a change in memory size not under the control of COMPROC, may cause unpredictable results. The user can deactivate the debounce routine by executing COMPROC with no DEBOUNCE statement.
7. Loading a machine language program may destroy COMPROC, DEBOUNCE routine, or other system data if an inadequate memory size parameter is specified. The user is responsible for the correct use of protected memory.
8. COMPROC cannot anticipate or detect system or user application errors. As a result the script may become unsynchronized causing incorrect results. For example, if a "PROGRAM NOT FOUND" error message results from a LOAD operation the following commands, such as a RUN statement, will probably fail. The user should try to keep scripts small and uncomplicated. For example, the script shown in Figure(3) is unnecessarily complicated for most practical for most practical uses.

User Command

COM

ENTER CMF PROCEDURE FILESPEC?PROC(200)

ENTER PARAMETER FIELD #1?PAYROLL/EBS

ENTER PARAMETER FIELD #2?GSF

Command File: PROC/CMF

```

100 'BASIC
110 '
120 '
130 'RUN "PAY/BAS"
140 'END
200 '62168
210 'DEBOUNCE
220 'VERIFY (ON)
230 'TIME ?I?
240 'DATE ?I?
250 'LOAD ?2?/OBJ
260 'BASIC
270 '4
280 '62718
290 'LOAD "???"
300 'AUTO 1000
310 'FOR I=1 TO 2
320 '  ?R?INPUT J: ?J+13
330 'NEXT
340 'END
350 ' ?B?SAVE "PAY/BAS"
360 ' ?F?RUN ?I?
370 '153
380 '192
390 'CMD "S"
400 'COM ?X?

```

Command Execution

```

DOS READY
VERIFY (ON)

DOS READY
TIME 17:58:19

DOS READY
DATE 04/22/79

DOS READY
LOAD GSF/OBJ

DOS READY
BASIC
HOW MANY FILES? 4
MEMORY SIZE? 61922
RADIO SHACK DISK BASIC...
READY...
LOAD "PAYROLL/EBS"
READY
AUTO 1000
1000 FOR I=1 TO 2
1010   INPUT J: ?J+13
1020 NEXT
1030 END
1040
READY
SAVE "PAY/BAS"
READY
RUN 1000
? 153
166
? 192
203
READY
CMD "S"

DOS READY
COM PROC
BASIC
HOW MANY FILES?
MEMORY SIZE? 65397
RADIO SHACK DISK BASIC...
READY
RUN "PAY/BAS"
: (under user control)

```

Figure(3). Complex COMPROC Example.

COMPPROC Version 1.2 Addenda

COMPROC Version 1.2 added features not found on Version 1.1 as follows:

1. COMPROC now works with the Apparat NEWDOS. To use with NEWDOS, the first line of the Command File (CMF) must specify APPARAT, e.g.
10 'APPARAT
2. Lower case driver object code has been added to Version 1.2. To request lower case, on the first line of the Command File, include the words LOWERCASE
Examples include:

10 'APPARAT, LOWERCASE

OR

10 '62137, DEBOUNCE, LOWERCASE

Optional do not use with Apparat NEWDOS

Note: Lower case hardware modification must be installed for the lower case code to function properly.

3. A screen print command has been added to COMPROC for use when using TRSDOS. To use the screen print command, specify DEBOUNCE on the first line of the Command file. When it is desired to print the screen contents, simultaneously press the SHIFT, down arrow, and P keys. This feature (JKL) keys along with a keyboard debounce routine are incorporated in the Apparat NEWDOS, and, therefore are not included in Version 1.2 of COMPROC.

When loading COMPROC to disk (Ref. pg 2 of manual), end dump at 7900 hex, i.e. line 10 should read:

10. PUMP COM/CMD (START=X'7C00',END=X'7900',TRA=X'7000')