SUPERPET REFERENCE SHEET No. 1. Subject: DOS Commands, MicroEDITOR

The Intrinsic Commands (6809 mED), Version 1.1. Enter at Command Cursor (PF 5).

COPY (cop)	copy disk/1.oldtitle to newtitle	Drive 1 file 'oldtitle' copied to drive 0 as 'newtitle', SEQ format.
	cop orgy,prg to disk/1.neworgy,prg	File 'orgy' copied to drive 1 as 'neworgy', as a PRG file.
	cop newone to ieee4	File 'newone' is sent to an ieee4 printer from disk/0.
	cop short,usr to disk/1.short,usr	File 'short', USR format, copied to drive 1, under same name.
	cop short,usr to disk/1.short	ERROR. Copies USR file 'short' as SEQ file to drive 1.
	cop 'a man,prg to disk/1.a man,prg'	Apostrophes or quotation marks allow spaces in file names.
	cop origin to disk9/1.origin (This disk unit transfer not possible with 'g ieee8-15')	Copies file 'origin' to drive 1, device 9, under that name, as SEQ file, from drive 0, unit 8.
	cop origin,prg to origin.bak,prg	Creates new file, drive 0, named 'origin.bak', backing up 'origin'.
RENAME (ren)	rename oldname to newname	Renames the file on drive 0.
	ren disk/1.oldname to newname	Renames the file on drive 1. Do not specify disk/1 twice.
	ren disk9/1.oldname to newname	Renames file on unit 9, drive 1.
SCRATCH (scr)	scratch disk9/1.badfile	Scratches 'badfile' on unit 9, drive 1.
	scr disk/1.*	Scratches all of disk/1, unit 8.
	scr disk/1.assycode.*	Scratches all files on disk/1, unit 8, beginning with 'assycode.'
	scr *	Scratches all of drive 0, unit 8.

Warning: Do not scratch files marked with asterisk on directory (*SEQ, for example). The asterisk denotes an improperly closed file. See next page: use the VALIDATE (or COLLECT) command to remove such files.

MOUNT (mou)	mount disk/1	Identifies disk 1 to drive 1 (Same as 'Initialize')
	mou disk	Mounts disk in drive 0.

The mount command is not required for 8050 or 8250 drives.

```
---------------------------------------------------------------------------------
  BASIC 3.0 DOS Commands. Enter at Command Cursor (PF 5).   Preface each command
with 'g ieee8-15.', as shown in first example.  To save space, this prefix is at
times not shown, but it is required on ALL commands. Use CAPITALS for commands.

COPY      g ieee8-15.C1=0                   Copies all of drive 0 to drive 1.
                                            Destination always is left of '='.

       C1:orgy=0:neworgy                    File 'neworgy' copied to drive 1 as 'orgy'
     (prefix with g ieee8-15.)              in its original DOS format. You need not
                                            specify PRG, USR, REL, or SEQ.

  g 'ieee8-15.C1:a file=0:a file'           Demonstrates use of apostrophes to
                                            copy to drive 1 from drive 0 with
                                            spaces in filenames.

  g ieee8-15.C1:verylongfilename=0:verylongfilename   A truncation problem. The
           (Fails)                                    DOS will not accept over
                                                      40 characters, so this
                                            command will not work. Bypass the problem
  C1:x=0:verylongfilename                   by copying file as 'x' and then renaming it
    (Works. Then rename)                    as shown.
---------------------------------------------------------------------------------
DUPLICATE          g ieee8-15.D1=0          Backs up on drive 1 contents of drive 0.
(Backup)                                    Destination always is left of '='.
---------------------------------------------------------------------------------
NEW                g ieee8-15.N1:diskname,##      Formats or news disk in drive 1,
(Header)                                          with new diskname. Use any combin-
(Format)                                          ation of two letters and/or numbers
                                                  for ##. The preface 'i' is NOT used.

 g ieee8-15.NO:newname      Scratch directory and all files; rename disk. Do not
                            use id no. or comma. A very fast scratch. Handy.
---------------------------------------------------------------------------------
VALIDATE          g ieee8-15.V0      Validates (or collects unused space from) old
(Collect)              or            disk 0. Wipes out all files marked with *.
                  g ieee8-15.V1      Specify drive 0! If you do not, and used drive
                                     1 last, you may Validate drive 1.
---------------------------------------------------------------------------------
CONCATENATE       Because of the 40-character limit on DOS commands, commands
                  may be truncated. If so, concatenate in steps.

    g ieee8-15.C1:big=0:tiny,1:weeny,0:teeny      Creates file 'big' on drive 1,
                                                  composed of files 'tiny' from
                  drive 0, 'weeny' from drive 1, and 'teeny' from drive 0. Again,
                  the destination always is left of '='.
---------------------------------------------------------------------------------
RENAME            g ieee8-15.R0:newname=oldname   Again, the new name is placed
                                                  left of '='. Renames file on
                                                  drive 0 to 'newname'.
---------------------------------------------------------------------------------
INITIALIZE         g ieee8-15.I1                  Identifies disk 1 to drive 1.
                   g ieee8-15 IO                  Use 0; don't depend on default.
---------------------------------------------------------------------------------
```

The Directory Commands (6809 mED), Version 1.1. Enter at Command Cursor (PF 5). (Not available in V1.0)

Command:	Result:
To: Disk	
di disk index	Puts directory of drive 0 on drive 0 as 'index'
di disk disk/1.index	Puts directory of drive 0 on drive 1 as 'index'
di disk/1 disk/1.index	Puts directory of drive 1 on drive 1 as 'index'
di disk/1 index	Puts directory of drive 1 on drive 0 as 'index'
di disk9/1 index	Puts directory of unit 9, drive 1 on drive 0, unit 8, as 'index' (default to unit 8, drive 0)
To: Printer	
di disk ieee4	Puts directory of drive 0 to ASCII printer on user/ ieee-488 bus
di disk/1 printer	Puts directory of drive 2 to Commodore printer on user/ieee-488 bus
di disk9/1 serial	Puts directory of unit 9, drive 1 to serial port (and/or to printer on that port)

Commands to Screen are identical, except that no destination device is specified, as in: "di disk9/1", which prints to screen directory of unit 9, drive 1.

General Screen-Handling Commands, microEDITOR, V1.0 and V1.1, given at Command Cursor (PF5). PF (Shifted Keypad) Commands are interleaved in BRACKETS [ ]

Command:	Result:
1,12d <RETURN>	**DELETES** relative lines 1 through 12, inclusive, from file. (No space allowed after comma in any delete command)
*d <RETURN>	**DELETES** entire file in microEDITOR.
d <RETURN>	**DELETES** line screen cursor is on.
+2d <RETURN>	**DELETES** second line below screen cursor. With a minus sign, deletes second line above screen cursor.
+2,$d	**DELETES** all text, from second line below screen cursor to end of file ($).
80,$d	**DELETES** file from line 80 through end of file ($)
[PF 2]	**DELETES** and excises line screen cursor is on. DANGEROUS. There is no recovery.
STOP	**RECOVERS** original line if errors are made and no RETURN has been entered on that line.

---

i or I <RETURN>	Automatically **INSERTS** new, blank line below screen cursor on each <RETURN>. Automatic exit from insert mode if CURSOR UP/DOWN key is used. Exit also with: . <RETURN> on new, blank line.
[PF 0]	**INSERTS** new, blank line immediately below screen cursor.
# <RETURN>	Returns **LINE NUMBER** of line on which screen cursor falls.
54 <RETURN>	**MOVES** screen cursor to line 54 of the file. DO NOT confuse RELATIVE lines of file with line numbers of microBASIC.
$ <RETURN>	**MOVES** screen cursor to last line of file.
1 <RETURN>	**MOVES** screen cursor to first line of file. (Use NUMBER 1)
[PF 9, PF .]	**MOVES** up-text by one screen (9), or down by one screen (.).
[PF 6, PF 3]	**MOVES** up-text by one line (6), or down by one line (3).
n <RETURN>	**NAMES** the last file command given at command cursor.
p name	**PUTS** the entire file to drive 0, unit 8, as 'name', in TEXT, SEQ format.
p disk9/1.name	**PUTS** the entire file to drive 1, unit 9 as 'name'.
g name	**RETRIEVES** or gets from drive 0 the file 'name', which is inserted, beginning on line just below screen cursor.
g name,rel	**RETRIEVES** or gets REL file 'name' from drive 0, unit 8. (Any file which contains ASCII representations may be retrieved. PRG files are not ASCII representations. Do NOT attempt to file a REL file from the microEDITOR.)
g disk9/1.name	**RETRIEVES** or gets file 'name' from disk 1, unit 9.
? <RETURN>	**RETRIEVES** or gets previous command at command cursor. DANGER. Function keys except L/R CURSOR or INS/DEL re-execute! Command may be re-executed with <RETURN> as well.
[PF 8]	Shifts to **SCREEN MODE** (single cursor, for editing).
[PF 5]	Shifts from **SCREEN MODE** to **COMMAND MODE** (two cursors).
/phrase <RETURN>	**SEARCH** for the first occurrence of 'phrase' in file.
+/ <RETURN>	**SEARCH** for the next occurrence of 'phrase'
-/ <RETURN>	**SEARCH** for the first preceding occurrence of 'phrase'
+/phrase <RETURN>	**SEARCH** from screen cursor forward for next 'phrase'
-/phrase <RETURN>	**SEARCH** from screen cursor backward for next 'phrase'

SUPERPET REFERENCE SHEET No. 3. Subject: Metacharacters in MicroEDITOR

Ref. p. 58+ (par. 7.4.2), Version 1.1. System Overview Manual:

Searches/replaces in the microEDITOR may be supplemented by metacharacters which generalize commands. The most useful are briefly summarized below. We do not repeat examples from the manual where those examples are clear. 'Metacharacter' is abbreviated to MC. %, /, ^, $, * and & are MC. The '.' is not.

MC and Commands:	Function and Examples of Use:
%	Represents any character, even if that character itself is a metacharacter, or is part of the search/replace character set, as in the examples following:
/%%	Finds any occurrence of '%', as in '/b%%', which finds b%.
/%%.	Finds a % followed by any character but a metacharacter. %. represents 'any character but a metacharacter'.
/%.%%	Finds a % preceded by any character, as in a%, b%, endd%. Most useful for locating integers in microBASIC.
/%%*	Finds a '%*'. This form finds any metacharacter, as in '/%/', which finds '/'.
/%%^	Finds any occurrence of '^'. (%^ alone finds start of line.)
/%.	%. finds any character. <u>Useless</u> unless in combination.
/9-%.%.-82	Finds any two characters following '-' and preceding the next '-', as in 9-nn-82, where nn is any two-character string. The period is essential. Will <u>not</u> find 9-n-82.
/%*	Finds zero or more repititions of the character which <u>precedes</u> the %*. <u>Useless</u> unless prefaced by a true search character (as below), for it finds <u>zero</u> or more repititions.
/9-%.%*-82	Here,'%.' represents any character, and %* locates zero or more repititions, as in 9--82, 9-n-82, or 9-nn...n-82.
/%^	Represents the start of any line.
/%^start	Finds the word 'start' wherever it starts a line.
/%$	Represents the end of a line. Requires <u>inversion</u> of order of entry in search string. See below:
/end%$	Finds any 'end' at the end of a line. Note search characters appear first.
/82.%$	Finds the three-character string '82.' at end line. The last character on the line <u>must</u> be included in the string.

Ref. p. 49 (par. 7.4.2), Version 1.1. System Overview Manual:
Test complex searches/replaces before use, or errors will occur which cannot be corrected without a manual search of the entire text. Search/replace commands are extremely precise, powerful, and dangerous.

There are four forms of the replace command: *c* changes all occurrences; *c changes all first occurrences in all lines; c* changes all in current line. The c alone changes only 1st occurrence in the current line. Only c* and c can be used with a specific line range, as in: 1,10 c*/ a / b /, which will change all occurrences of 'a' in the line range shown, while 1,10 c changes only 1st 'a'.

Search/Replace Command:	Effect:
`*c*/able/ible`	Replaces the phrase 'able', wherever found, with the phrase 'ible'. Dangerous! Changes occur within words.
`*c*/ no / yes /`	Replaces the word 'no' with 'yes'. Note spaces. Fails at very start or end of line (there are no spaces!). Also fails when 'no' is immediately followed by any punctuation mark. Last, always fails when 'no' is the last word on a line (the rest of the line is not spaces, but nulls. There, / no/ yes/ is required. This command changes all occurrences, all lines.
`*c/no /yes`	Changes all 'no', first occurrence, all lines, with 'yes'. Fails to insert space after 'yes'. For space: *c/no /yes /.
`c*/no /yes /`	Replaces as above, but only on current line, all occurrences.
`c/no /yes /`	Changes only first occurrence on the current line.
`10,12c/ no / yes /`	Replaces 'no' with 'yes' at first occurrence, from lines 10 through 12. Will not change 'no' at column 1 or 80 (no spaces), or if 'yes' is last printable character on a line.
`c*/ no / yes /`	If phrase 'no demons' is at end line (column 80), replaces 'no' with 'yes', but forces the 's' in 'demons' off screen.
`.,+10 c/ a / b /`	From current line '.' to ten lines forward, replaces first occurrence of ' a ' to ' b '.
`c/%^obo /bob /`	Replaces 'obo' at start of line with 'bob'. Fails if 'obo' is only word on line, where /%^obo/bob/ is required.
`*c/%^"/`	Deletes any '"' at start of all lines. Pulls all text left.
`1,.c/ d /`	Deletes 'd', first occurrence, line 1 to current line.
`c*/  / /`	Replaces all double spaces in current line with a single space. Useful for changing spacing in captions and headings.
`10,12c/  / /`	As above, but only 1st occurrence, lines 10 through 12.
`c*/ /  /`	Inserts double spaces for single in current line. Useful for captions and headings. Changes all occurrences.

SUPERPET REFERENCE SHEET No. 4. Subject: ASCII Codes in SuperPET

The Roman character set in SuperPET is pure ASCII (not PET ASCII). Any ASCII table defines the printable character set and the CONTROL functions of the non-printable codes. If you add 128 to its ASCII code, you will print any character to screen in reverse field. POKE, PEEK, and print codes, wherever used, are identical for a given character or control function.

The fourteen CONTROL codes summarized below apply to SuperPET's screen, to screen printing, disk files, and general operations, in all languages, including APL (in APL, []IO, index origin, must be 0 for the codes to work). In APL, []IO is set to 1 when you load that language. Allow for it.

ASCII Code:	Key to use/comments:	Screen Effect:
0	NUL. If in disk file, deletes rest of line in microEDITOR.	Prints small square in language.
1	HOME. HOME key or in program.	HOMEs the Cursor
2	RUN. SHIFT/RUN in microEDITOR starts any program.	None
3	STOP. STOP key pauses program; in program, ends execution. ASCII 3 in disk file stops a read in V 1.0 microBASIC, but not in V1.1.	None
4	DELETE key deletes character the cursor is on, pulls characters from the right.	DELETE right.
5	Usual Commodore SHIFT/INSERT.	INSERT space
6	Use ESCAPE key or in program.	ERASE from cursor to end line
7	CURSOR key and in program.	CURSOR right
8	SHIFT/CURSOR key and in program.	CURSOR left
9	TAB key tabs to default or user-set tabs. Useful in program.	TAB to next preset stop.
10	CURSOR key and in program.	CURSOR down
11	SHIFT/CURSOR key and in program.	CURSOR up
12	SHIFT/CLR key and in program.	CLEAR screen, home cursor
13	RETURN key and in program.	CARRIAGE return
127	REPEAT key (old Commodore DELETE)	DELETES char. to its left

(All SuperPET keys repeat, so you don't need the old REPEAT key function.)

Printer Filenames

SuperPET recognizes three printer filenames: 'serial' for any ASCII printer on the serial (RS-232) port; 'ieee4' for any ASCII printer on the user/ieee bus, and 'printer' for any Commodore printer using PET ASCII on the user/ieee bus. So far as we know, all letter-quality Commodore printers (such as 8300P) are ASCII printers and are addressed as either 'ieee4' or 'serial', depending on which bus or port they use. Examples follow, showing how to list a program to printer from the microEDITOR, which runs in all languages but APL:

```
p printer       (Lists contents of mED to a Commodore printer on ieee.)
1,45 p printer  (Lists lines 1 through 45 to a Commodore printer on ieee.)
p ieee4         (Lists mED contents to an ASCII printer on ieee, device 4.)
45,$ p serial   (Lists lines 45 through end of file to printer on serial port.)
```

Definition of Filenames in SuperPET:

```
          disk address - channel/drive.(Waterloo type)file-designator, DOS format
Reference      1              2     3       4                4.1              5
below:
```

Ref 1 Includes two designators: the word 'disk' plus the device number, in the form 'disk8' for device 8, 'disk9' for device 9, etc.

2 Automatically assigned by the interpreter; <u>DO NOT SPECIFY</u> without good reason (channels 2-14 are used when a designator is specified; channel 15 only when there is no designator. You may override auto assignment, but run serious risk of two open files having the same channel number.)

3 Drive Number: 0 or 1; default is to 0

4 Select one of three file types: text, fixed, variable. Default to text.

4.1 Commonly miscalled 'filename'. The TITLE of the file on directory. Waterloo calls it the file-designator.

5 DOS format; default is to sequential. 'rel' must be used for relative files, 'prg' for program files; 'usr' for user files.

EXAMPLES OF FILENAMES: File title is 'example 1'

<u>For Device No. 8:</u>

'disk8-12/0.(text)example 1,seq'	[full filename; no use of defaults; a 'text' file, 'SEQ' DOS format]
'disk/0.(t)example 1'	[same filename, some defaults]
'example 1'	[same filename, all defaults]

* * *

<u>For Device No. 9:</u>

'disk9-12/1.(f:80)example 1,rel'	[full filename; no defaults]
'disk9/1.(f)example 1,rel'	[same filename, all defaults]

SUPERPET REFERENCE SHEET No. 5 Subject: Condition Code Register

Flags in the CC register of the 6809 are defined on page 93, Assembler Manual. The flags are summarized in a two-digit hex number under CC when registers are dumped in the monitor. For a value of $C9 in that register, see example below on how to convert the value to binary and assign value to each flag.

For effect of 6809 instructions on CC register, see p. 120, Assembler Manual.

Conversion Table : Hex to Binary

Hex	Binary	Hex	Binary	Hex	Binary	Hex	Binary
0	0000	4	0100	8	1000	c	1100
1	0001	5	0101	9	1001	d	1101
2	0010	6	0110	a	1010	e	1110
3	0011	7	0111	b	1011	f	1111

	$C				$9			
$C9 allocated:	1	1	0	0	1	0	0	1
Flag ID:	E	F	H	I	N	Z	V	C
Bit Number:	7	6	5	4	3	2	1	0
Full Flag name:	Entire State Flag	FIRQ mask bit	Half-carry Flag	IRQ mask bit	Sign Flag	Zero Flag	Over-flow Flag	Carry Flag
Logic operations affect?				No	Yes	Yes	Yes	No

<u>Bit:</u> <u>Data on the Flags:</u>

0 **Carry.** Holds most significant bit from a carry operation. Inverts actual carry after subtraction, so it acts as a borrow. MUL causes this bit to be bit 7 of a 16-bit result.

1 **Overflow** (V). Flag is 1 when result of two's complement arithmetic overflows.

2 **Zero flag.** Set 1 when an operation produces zero; 0 for non-zero.

3 **Sign** (N). Takes value of most significant bit of a result.

4 **IRQ Mask bit** (I). Set to 1 when the 6809 is not to recognize IRQ requests.

5 **Half-carry** (H). Holds any carry from bit 3 to bit 4 caused by execution of an 8-bit instruction (ADC or ADD). Simplifies BCD operations.

6 **FIRQ Mask bit** (F). When set 1, prevents fast interrupts.

7 **Entire flag** (E). Discriminates between regular and fast interrupts. E is set to 1 for any interrupt which stacks all registers. E is set 0 at FIRQ, as it stacks only PC and CC.

Flags do not change until the processor executes an instruction which requires a change. Any bit can be changed by ORing or ANDing. OR CC with 1 will set flag if flag is clear; AND CC with 0 will clear a flag if that flag is set.

The manuals do not document a number of ROM library routines. This reference sheet lists some of those routines. Note that the 'fpp.lib' exports on disk are floating point math routines; those below are only for integer arithmetic. Be warned: All routines listed below adjust the stack after parameters are passed, and a 'leas' to correct the stack is not required. All are found in watlib.exp. Long dashes (___mul) are three underlines. 'P' stands for 'parameter.' P1 is passed in D register; other parameters on the hardware stack. Results are in D.

Routine:	Function, Parameters, Examples:	Comments:
___mul $b060	Multiplies P2 by P1 (P2=3, P1=6, result=18) 16-bit signed product	P2=multiplier P1=multiplicand
___div $b066	Divides P2 by P1 (P2=10, P1=2, result=5) 16-bit signed quotient	P2=Dividend P1=Divisor
___mod $b069	Returns the remainder after an integer division (P2=100, P1=80, result=20)	P2=operand P1=modulus
___neg $b063	Returns 16-bit two's complement of positive number (the negative); or positive value, negative operand.	P1=operand
_rshift $b06c	Shifts 16-bit operand (P2) right by the number of bits in P1. If P1 is negative, shifts left. Effect with positive P1 is to divide by powers of 2: 2,4,8,16, etc. Examples: Operand=16, rshift 1, value=8. Least significant bit shifted out; 16385 thus becomes 8192 on an _rshift of +1. An _rshift is logical; the sign bit is treated as part of the value shifted: 32768 (which sets N flag and sets bit 16 (MSB), becomes 16384 on _rshift of +1.	
_lshift $b06f	Shifts 16-bit operand (P2) left by the number of bits in P1. If P1 is negative, shifts right. For every shift left, multiplies by power of 2, but most significant bit, if set, is shifted out.	

Example in Assembly Language:

```
ldd  #3      (P2) ;Answer returned in
pshs d            ;D register is 1,
ldd  #2      (P1) ;the quotient of an
jsr  ___div       ;integer division.
[result in D register]
```

Example in Language:

(Use sys call format for language)

```
sys xxxx, P1, P2, etc.
```

The 'xxxx' is the address of the routine as found in watlib.exp, and as listed above.

kyputb_
$dd82

```
loop
  jsr kyputb_
until ne
```

Routine gets a character from keyboard and does not print it to screen. Character returns in B reg. The loop at left pauses a program until any key is hit.

```
loop
  jsr   kyputb_
  cmpb  #'y
  quif  eq
  cmpb  #'n
  quif  eq
endloop
stb  answer
```

Loop at left will cycle until it receives either a 'y' for 'yes' or a 'n' for 'no,' at which point the reply is stored in label 'answer'. This is a useful routine for limiting user input to specific replies.

kyputb_ will accept NUL or 0 as input, so it must be limited to a non-null response.