

Filter Disk Number Two

-----

T A B L E O F C O N T E N T S

=====

CUSTOMER SERVICE INFORMATION .....	15
COMML/FLT .....	2
DICTATE/FLT .....	3
DOSPEED/FLT .....	4
KSMPLUS/FLT .....	5
LCOUNT/FLT .....	7
MARGIN/FLT .....	8
MAXLATE/FLT .....	9
SLOSTEP/DCT .....	13
VIDSAV/CMD .....	14

LDOS - Filter Disk Number Two - Model I & III  
Copyright - 1983 by Logical Systems, Incorporated  
All Rights Reserved

LDOS is a trademark of Logical Systems, Incorporated  
Prowriter is a trademark of C. Itoh, Inc.  
Radio Shack, TRS-80, and Scripsit are trademarks of Tandy Corp.

COMM1/FLT

This is a communications filter designed for the LDOS RS232x series of drivers. It provides testing for modem carrier, delay between characters, and linefeeds and nulls after carriage returns. The proper syntax is :

```
=====
| FILTER devspec using COMM1/FLT (parm,parm)
|
| devspec is a device which has been previously SET
|   to one of the RS232x drivers.
|
| Parameters:
| CARRIER= ON or OFF. Setting the carrier to ON will
|   cause all input or output requests to be
|   ignored unless a MODEM is receiving a
|   carrier signal. The default is OFF.
| ADDLF=   ON or OFF. Setting this parameter to ON will
|   add a linefeed after each carriage return
|   transmitted. The default is OFF.
| NULLS=   0 to 255. This designates the number of nulls
|   (character X'00') to send after each carriage
|   return. The default is 0.
| DELAY=   0 to 65535 or X'00' to X'FFFF'. This allows
|   the establishment of a variable timing delay
|   between output characters in order to allow
|   transmission to systems incapable of full
|   speed reception. The default is 0.
| MASK=    ON or OFF. Setting this parameter to ON will
|   strip the high bit from received characters.
|   The default is OFF.
|
| Abbr: ON=ON/YES/Y, OFF=OFF/NO/N, CARRIER=C, ADDLF=A
|       NULLS=N, DELAY=D, MASK=M
| Memory Usage : 113 bytes (X'71')
|
=====
```

This routine is handy for many types of serial communications requirements. The Nulls and Addlf parameters are useful to add necessary requirements for serial printers. The carrier parameter is useful for Auto-Answer modems. The Mask parameter removes parity bits that may have been added by the sender. The mask parameter should be used for ASCII data only (X'01' through X'7F') and not for 8-bit binary data.

The filter is attached to a device which has been established using the appropriate LDOS RS232x driver for a given hardware configuration. Proper use of the driver is detailed in the LDOS manual under the SET Library Command and in the Drivers/Filters Section under RS232x, where x is the individual driver designation. The sample implementation below ASSUMES the use of a standard Radio Shack Model III.

```
SET *CL TO R5232T/DVR (BAUD=1200,WORD=7,STOP=1,PARITY=ON,EVEN)
      followed by:
      FILTER *CL using COMM1/FLT (MASK=ON)
```

The example establishes 1200 Baud, 7 bit word, 1 stop bit, even parity communications through a device arbitrarily named \*CL. The communications are further modified by the filter so that incoming parity bits are removed.

```
SET *CL RS232T
      followed by:
      FILTER *CL COMM1 (N=50)
```

This provides default settings of 300, 7, 1, and even. Additionally, 50 nulls will be sent after each carriage return (X'0D').

DICTATE/FLT

This is a keyboard filter which allows the switching of the attached cassette recorder motor on and off.

```
=====
|
| FILTER *KI using DICTATE/FLT (parm)
|
| CHAR= is an optional parameter which allows the
|        definition of which keyboard character will be
|        used as the switch. The default is CONTROL-C.
|
| Abbr: CHAR=C
| Memory Usage : 33 bytes (X'21')
|
|=====
```

The CHAR character is set to CONTROL-C (control-[A-Z] is generated by simultaneously depressing the shift key, downward pointing arrow, and the character key) which is an ASCII 3. It is a good idea not to use a normal printable character because DICTATE will function every time the character is typed. The range for the CHAR parameter is 1 through 255 decimal or hex X'01' through X'FF'.

Keep in mind that if the value cannot be generated from a single or combination of keys that the filter is fairly useless. See the keyboard charts in the LDOS owners manual to determine the available choices. Be sure that if a value is selected which can only be generated with the KI driver, that KI/DVR is established.

The use of the filter is sheer simplicity. Type the selected key sequence and the cassette motor starts or stops depending on the current state. DICTATE merely does the opposite of the current function.

Thus a typist could use this to type dictation into a word processor, a programmer to record notes for later use, etc.

DOSPEED/FLT

This filter is used to regulate the speed of an output device.

```
=====
| FILTER deyspec using DOSPEED/FLT |
|                                   |
| There are no parameters.         |
| Memory Usage : 58 bytes (X'3A')  |
|                                   |
=====
```

The devspec may be any LDOS device which has an output direction. DOSPEED controls the rate of output to a device. The rate is determined by the 0 to 9 keys where 0 is the fastest speed and 9 is the slowest.

For example, suppose the filter is installed over the video device \*DO. During a BASIC listing to the screen, the speed of the listing may be changed by pressing any numeric key (0-9).

Because DOSPEED continually scans the keyboard for commands, the filter pulls characters out of the type-ahead buffer. This could cause characters in an executing JCL file to be improperly used. Use of this filter with any program which utilizes or depends on this feature is not advised.

## KSMPLUS/FLT

This filter works essentially the same as KSM provided with a standard LDOS system, but is capable of extended features such as the editing of a resident KSM table, recall of date and time, and recall of the last LDOS command.

```
=====
|
| FILTER *KI to KSMPLUS/FLT using filespec (parm,parm)
|
| filespec is an existing KSM type file
|
| Parameters:
| ENTER=   is an optional parameter specifying the
|           character to be used as an embedded ENTER.
| SPACE=   is the amount of additional memory to be
|           reserved in case the resident table is to be
|           expanded. The default is 35.
|
| Abbr: SPACE=S, ENTER=E
| Memory Usage: 552 (X'228') + length of table + space
|
|=====
```

KSMPLUS basically functions exactly as the Key Stroke Multiply filter found in the LDOS owner's manual in the DRIVERS/FILTERS section. The user should be familiar with this section because the following will cover only the advanced features.

<CLRS> in the following refers to the <CLEAR> key depressed simultaneously with the <SHIFT> key.

The first difference is the SPACE parameter. If it is planned to replace certain keys while running, additional space may be reserved if the replaced strings are longer than the resident strings. For example, if it were desired to replace the <D> key string of DIR; with DEVICE; the latter requires three more spaces in memory. Additional room to accommodate this change must exist in the high memory table and, therefore, should be specified when the filter is first entered. The three extra spaces would come from the area defined by the SPACE parameter. If the default of 35 had been used and the above change were implemented, there would be 32 spaces left over after the string substitution occurred.

Similarly, if DIR; were re-substituted for DEVICE; , the three characters would rejoin the pool of available space. It can be seen that replacing phrases with a null string is a method of increasing the space pool. If there were no space available, then replacing <W> KSM of "WI" with "WISCONSIN" would not be allowed. The necessary seven characters could be found by first editing <D> "DEVICE;" to a carriage return after which the space pool would now contain seven characters.

To substitute a KSM entry depress <CLRS><E> simultaneously. In the upper left corner of the screen will appear a plus mark (+) to indicate that the KSMPLUS editor is active. To abort without any changes hit the <BREAK> key. To edit a KSM entry press any KSM eligible key (A-Z). The pressed key followed by "=" will appear. To change the current entry merely type in a new entry (don't forget the imbedded enter if necessary). Press <ENTER> when finished. If there is not enough room, the edit will abort. To abort in the middle of typing in a substitute string type <CLRS><A> and the original string will be left intact. To remove the KSM entry entirely, type the <ENTER> key as the first character of the new string (this adds the length of the entry into the space pool).

Any keyboard character (including <BREAK>) can be part of a KSM entry. Keep in mind that this "live" edit is not the same as if a KSM entry is being created with a BUILD command. This sequence "memorizes" all keystrokes with the exception of the left-arrow (to allow backspace while editing) and the <ENTER> key (used to signify that edit is over). The imbedded enter key is designated by the ENTER parameter specified when initiating KSMPLUS. That character (defaulted as ";"), is the third character which cannot be part of the replacement string (unless it represents the <ENTER> key pressed).

Other built in features of KSMPLUS are: <CLRS><S> will print the current real time clock in the (hh:mm:ss) format. <CLRS><Z> will print the current system date in the <MM/DD/YY) format. <CLRS><X> will print the last LDOS command entered. All of these features may be used in the edit sequence.

## LCOUNT/FLT

This filter writes a line number preceding every line of text in an output device usually the \*PR. The proper syntax is as follows:

```
=====
|
| FILTER *PR using LCOUNT/FLT (parm,parm)
|
| Parameters:
|
|   START      Specifies the number at which the counter
|               starts or resets. The default is 00000.
|               The range is 0 through 65535.
|   INC        Specifies the Increment to be added to the
|               next line number. The default is one and
|               the range is 1 through 1000.
|   CLEAR      Selects a character to be intercepted by
|               the filter which resets the current count
|               back to the number specified by the START
|               parameter. The default is none and the
|               range is 1 through 255.
|   PAGE       Proper response is either OFF/NO/N or
|               ON/YES/Y. When selected, will reset the
|               counter after each top-of-form character
|               (usually X'0C') is passed.
|
|   Abbr: START=S, INC=I, CLEAR=C, PAGE=P
|   Memory Usage: 203 bytes (X'CB')
|
|=====
```

This filter will write a five digit line number right justified and padded with leading zeros in front of every line terminated by a carriage return. It will start numbering with any specified number from 0 through 65,535 and then continue numbering until 65,535 is reached and then begin again at the starting number. A Space will separate the number from the printed text.

The count may be reset by printing the character selected by the CLEAR parameter. The filter will not print the character but instead reset the counter. The character for this reason should be one which is not normally printed but is capable of being generated by the printing program or through the means of a direct printer command sequence such as the MINIDOS filter's <CLEAR><SHIFT><P> command.

The page parameter which defaults off, will cause the counter to reset every time an X'0C' character is printed (top-of-form).

Some examples of the command sequence:

```
FILTER *PR LCOUNT (START=100,CLEAR=127,INC=10)
```

This would start at line 100 increase the line number by 10 and reset if an ASCII 127 (X'7F') is received.

```
FILTER *PR LCOUNT (S=100,I=10,C=127)
```

This example performs the same functions as the previous one.

```
FILTER *PR LCOUNT (S=10,P=Y)
```

This would start the lines at 10 and reset to 10 at each top of form.

MARGIN/FLT

This filter sets the width of the left margin for the \*PR device. The proper syntax is as follows:

```
=====
|
| FILTER *PR using MARGIN/FLT (parm,parm)
|
| Parameters :
|
| WIDTH  Sets margin width from 1 to 130.
|         The default is 5.
|
| CONT   Control character which is the first
|         character of a two character code sequence.
|         The default is 27 X'1B'.
|
| Abbr: WIDTH=W, CONT=C
| Memory Usage: 102 Bytes (X'66')
|
|=====
```

The filter works in the same manner as does the Margin parameter of the PR/FLT which comes standard on the LDOS Master Diskette. However, this filter will allow a sequence of control characters to be passed to the \*PR device. The control sequence to be passed must be a two character sequence. If the control initialization character for a specific printer is other than a 27 (X'1B' ), it must be set with the CONT parameter, or it will be ignored. The control character must be less than ASCII 32 (X'20') or an error will result and the filter implementation sequence will be aborted.

The control sequence should not be one which alters the font spacing. MARGIN/FLT achieves its purpose by injecting a variable number of spaces in front of a line of print. Changing fonts from a 10 cpi format to a 5 cpi format would NOT change the amount of spaces to be printed.

The default margin setting (achieved by specifying no parameter) is five. Any margin width of 1 through 130 may be specified. Note that the number represents the WIDTH of the margin and not the first TAB position to print. (I.E. To start printing at tab 20 the margin width should be 19. A margin width of 20 would start printing at tab 21.)

The difference between this filter and the MARGIN parameter of the PR filter is that this filter passes a printer control sequence BEFORE sending the margin. Since many printers require a control sequence to be the first part of a line, the PR filter (which sends the margin first) would cause incorrect results.



### MAXLATE/FLT

This filter is used on any device to "translate" a user defined key into a group of characters from 0 to 255 elements in length. This provides an effective means of "foreign" translation between the computer and many peripherals. It is hard to overestimate the efficacy of this program. Some suggested uses are given below. The proper entry syntax is:

```
=====
FILTER devspec to MAXLATE/FLT using filespec (parm)

devspec  is any LOOS device.
filespec is any ASCII file which has been previously
          established as a translation table.

Parameters :
INPUT    declares that an input/output device should be
          only translated in the input direction.
OUTPUT   declares that an input/output device should be
          only translated in the output direction.

NOTE: The parameters are mutually exclusive and need
      not apply to unidirectional devices.

Abbr : INPUT=I, OUTPUT=O
Memory Usage: 110 bytes + length of table (X'6E')
=====
```

MAXLATE/FLT is used to filter a specified device according to translation data stored in a disk file. The translation file is created by anything capable of generating an ASCII file such as BUILD, LED, a word processor, text editor, or an applications program. This filter program greatly facilitates the translation of data to an I/O channel which provides greater versatility in communicating to peripheral equipment.

### The Translate Data File

The translation table (specified in the command line by "filespec") contains the substitutions to be used by MAXLATE. The default file extension used by the command line is "/MLT". The default extension will aid in the identification of translation table files and is recommended. The translation code is entered into the file as a free-form equivalence field. All translations take the form K=T where K is the character upon receipt of which, MAXLATE will substitute the phrase defined as T.

The key value K must be either a single character within quotes or be a two character hexadecimal value. For example, "A"= or 41= , represent the same condition. Every time the upper case character A is encountered, it will have the phrase subsequent to the equal sign substituted for it.

The string value of the phrase T may also be within quotes or a series of hex digits. Thus, if it were desired to translate every upper case [A] into a lower case [a], any of the following would produce identical results : 1) "A"="a", 2) 41="a", 3) "A"=61, or 4) 41=61. In addition, T may be nothing or a mixture of hex digits and material within quotes.

For example, suppose it were desired (on a printer with backspace capability) to place a slash [/] over a zero to render a [0] rather than a [0]. This requires a three for one substitution. The ASCII character for [0] is 30, a backspace character on some printers is 08, and a slash is 2F. The translation code can be rendered several ways, among which are: ["0"="0",08,"/"], ["0"=30,08,2F], [30=30082F] etc.

Note that commas and spaces may be included for readability but are NOT necessary. This particular method has a few restrictions. Note that every line of the translation table must end by pressing the <ENTER> key, and that quotes are used as delimiters for data. Therefore, in order to imbed carriage returns (X'0D') and quote marks (X'22'), they MUST BE entered as hex digits.

Beyond that, the line must contain no more than 255 characters to be output. There is no limit to the size of the translation file, but keep in mind that it must reside in high memory so that the file must be shorter than what RAM is available LESS the run time memory requirements. The data file may also contain comment lines designated by beginning the line with a period [.] and ending it with <ENTER>. The comment lines will not be stored in memory but will instead print to the video display as the data file loads. This allows relevant contents to be displayed. For more information see the example section.

Finally, a character can be totally eliminated by placing nothing after the equal sign. As an example, the character X'1C' when printed to a TRS-80 Model I or III video will cause the cursor to "home" to the upper left corner. Printing will resume at that location which overwrites existing characters. This can be prevented by a MAXLATE data line such as [1C=] and FILTERING \*D0 for output with MAXLATE. MAXLATE would trap the character X'1C' from being sent to video and thus the cursor reposition would not occur.

#### INPUT and OUTPUT Parameters

All LDOS devices have a direction sign as follows:

```
=> implies one-way OUTPUT (*PR)
<= implies one-way INPUT (*KI)
<=> implies two-way INPUT and OUTPUT (*DO)
```

In MAXLATE the translation occurs only one direction at a time. In one-way devices no parameter should be specified. Input or output filtering will occur based on the direction. \*KI will automatically translate a character on its way TO the system and \*PR will automatically translate a character on its way FROM the system. With a device such as \*DO which can be either, the direction in which the translation will occur MUST BE SPECIFIED. To translate characters going FROM the system, specify the OUTPUT parameter when filtering \*DO. To translate characters coming TO the system, specify the INPUT parameter. If there is a need to translate in both directions of a bi-directional device, FILTER the same device again with the opposite parameter.

#### Examples and Suggestions

The strength of MAXLATE is in the ability to form a common control language between devices which are similar. This is, of course, over and above its inherent translate ability. If an installation regularly uses two LDOS compatible word processors, a translation file (hereinafter referred to as a Maxtran) could be developed which made the edit calls of one identical to the second. Two maxtrans could be created to alter both word processors into yet a third but common control set.

In an office which contained a Model II and I TRS-80™, it was necessary to transmit a heavily controlled document (boldface, underlines etc.) from the Model I under LDOS to the Mod II. A maxtran was built which converted the various LSCRIPT print control codes to the SCRIPSIT(tm) 2.0 equivalents. A device called \*CL was then SET to the RS232 driver and then filtered with MAXLATE. The source file was then transmitted to the Model II and MAXLATE intercepted the incorrect codes and substituted the correct codes and thus saved hours of editing.

If a printer is capable of characters which are normally not generated by a single keystroke, a maxtran could be created to add them. One common irritation in the microcomputer world is that the latest generation of dot-matrix printers contain many functions such as different character sets, boldface, underline and many print effects. There is a problem in implementing these features because they are machine specific to a point where the same manufacturer does not provide common methods to effect them! MAXLATE provides a solution because a machine specific maxtran could be implemented to FILTER the output to produce IDENTICAL effects between machines.

An example is provided which would, more or less, produce the same output from a word processor on a Radio Shack™ Line Printer VIII and a C. Itoh Prowriter™ 2.

<u>Print Effect</u>	<u>Prowriter 2</u>	<u>Line Printer VIII</u>
Normal Characters	1B : 4E	1B : 13
Enlarged Characters	1B : 45	10 : 14
Reverse Linefeed	1B : 72	1B : 0A
Boldface Start	1B : 21	N/A
Boldface Stop	1B : 22	N/A
Underline Start	1B : 58	0F
Underline Stop	1B : 59	0E

Notice the differences between codes as well as the fact that the LP VIII is incapable of Boldface. The following chart will establish an arbitrary common command set between the printers.

Normal chars	CTRL P
Condensed	CTRL X
Reverse LF	CTRL R
Bold Start	CTRL B
Bold Stop	CTRL N
Underline Start	CTRL U
Underline Stop	CTRL I

The ASCII values of the above list are 10, 18, 12, 02, 0E, 14 and 09 respectively. The maxtran for the Prowriter 2 would be:

```
.Prowriter 2 Print Codes
10=1B 4E
18=1B 45
12=1B 72
02=1B 21
0E=1B 22
14=1B 58
09=1B 59
```

This is saved under the filename PRO2/MLT and is used whenever the Prowriter is attached to the computer. The maxtran for the LP VIII would be:

```
.Line Printer VIII Print Codes
10=1B 13
18=1B 14
12=1B 0A
02=
0E=
14=0F
09=0E
```

Note that the comment lines in the maxtrans will print to the video and not become part of the file. The LP VIII codes for boldface do not exist but if a mutually compatible text is to be generated, it is necessary to throw the calling codes away in the event that they could cause an unpredicted action. It now remains only to generate text using the arbitrary code sequence and (barring boldface) the two very different printers would produce two very similar documents WITHOUT HAVING TO EDIT THE TEXT; something perfectly impossible without MAXLATE.

## SLOSTEP/DCT

This module is a high memory disk driver for certain disk drives. Its use is necessary to insure accurate read/write operations but only on some brands of drives. Normally, the drives involved have two read/write heads as opposed to "flippy" type drives that do not access either side without physically inverting the diskette. However, any type of drive which has difficulty formatting might be aided by this driver.

Application of SLOSTEP/DCT is required to alleviate excessive difficulty in formatting the described drive types. If when formatting, cylinder lockout occurs from a certain point through the end of the diskette (or in similar consecutive blocks) then this driver might relieve the problem.

To use SLOSTEP/DCT at "LDOS Ready" type:

```
SYSTEM (DRIVE=n,DRIVER)
      or
SYSTEM (DRIVE=n,DRIVER="SLOSTEP")
```

Where n is the target drive number (0-7). Note that the extension "/DCT" is assumed. If the first method is used the prompt "Enter DCT driver <BREAK=default> :" will appear. Answer this prompt by typing "SLOSTEP".

The message "Stepping delay for double headed drives" acknowledges that the driver is active. Apply the driver to as many double headed drives as desired.

To avoid the drudgery of repeated implementation, utilize the "SYSGEN" parameter of the "SYSTEM" library command as documented in the Library Section of the LDOS manual.

Memory Usage: 90 bytes (X'5A')

VIDSAV/CMD

Filters the video and keyboard to reside a screen in memory which can be saved to, restored from, or swapped with the current video display.

```

=====
VIDSAV
VIDSAV (KEY=ON/OFF)

The parameter KEY is ON or OFF. Default is on.

Abbr: KEY=K
Memory Usage : 1154 bytes (X'482')
=====

```

VIDSAV establishes a buffer in high memory which is equal in size to the contents of the 64 by 16 video display. The current screen may be written to the buffer. A screen in the buffer can replace the current screen, or the current screen may be exchanged for the one in the buffer.

Commands for VIDSAV may come directly from the keyboard or through normal screen output in an applications program. It is possible to disable keyboard control of VIDSAV so that an existing application is not affected by user entries.

To use the filter type VIDSAV at LDOS Ready. The keyboard parameter (KEY=) should be switched off at this point if necessary.

Commands are as follows:

Job	Keyboard	Application
-----	-----	-----
save screen	<CTL><B>	ASCII 2
get screen	<CTL><C>	ASCII 3
swap screen	<CTL><D>	ASCII 4

<CTL> refers to the SHIFT and DOWN-ARROW keys depressed simultaneously with B, C, or D. An example of a BASIC line to save the screen would be

```
PRINT CHR$(2);
```

Note that it is necessary to suppress the carriage return through use of the semi-colon. Otherwise the screen would scroll. This filter does not work in the expanded video mode.

The LDOS Filter Disk #2 will provide capabilities not found when operating under the Standard LDOS system. For the most part, it is designed to utilize the device independent architecture of the LDOS system.

In describing each module, the documentation conforms to the syntax used in the LDOS manual. Most modules are described by a syntax block and an explanation of the function performed by the filter.

Several of the filters will have more than one parameter which will cause the filter to perform a different function. Many of the filters are directed to certain devices but may be applied to any device meeting the requirements.

Note that more than one device may be filtered with the same filter at the same time. Each time a device is filtered a new memory allocation will be used to contain the filter program. The first filter implemented will be the last one executed. If the need arises to re-filter a device with different parameters, a reset of the device may be necessary, particularly if conflicting conditions have been established.

To obtain assistance with any module included with this package, call or write to the following:

Logical Systems, Incorporated  
8970 North 55th Street  
P.O. Box 23956  
Milwaukee, WI 53223  
(414) 355-5454

NOTICE OF WARRANTY

This product is sold on an "as is" basis. Logical Systems, Incorporated makes no expressed or implied warranty of any kind with regard to the software or documentation.

All products sold and manufactured by Logical Systems, Incorporated, hereinafter referred to as LSI, grant the purchaser certain customer support privileges. This support shall be limited to the privilege of having the master diskette updated as often as desired for the current update fee. This is limited to updates within the current Series of the program. LSI will also provide a lifetime warranty on the physical diskette media of the original serialized master diskette. If the diskette media physically fails to retain the original program, replacement media will be provided at no charge. This does not include media that has been damaged in shipment from the user to LSI, or media that has been damaged by the user or their equipment. To receive this support, the user MUST fill out and return a specific registration card pertaining to the product, within 30 days of purchase. Should a user find a valid error in the program and clearly define it in writing to LSI, every effort will be made to correct the error. All support shall only apply to registered owners.

Logical Systems, Incorporated and its associates assume no liability whatsoever, with regard to the reliability and/or fitness of their products. All data entrusted to these programs and the computer that they are operating on are the sole responsibility of the user. Under no circumstances will LSI or its associates be held liable for the loss of TIME, DATA, PROGRAMS or for any consequential damages incurred by the user.