

71-00007

PROGRAM DESCRIPTION

Program Title Character Set LEX File Generator

Contributor Bruce Stephens

Address Hewlett-Packard Company (PCD), 1000 NE Circle Blvd.

City Corvallis State Oregon Country U.S.A.

Telephone 757-2000 Zip/Postal Code 97330

Program Description (include equations) This program creates a LEX file that contains an alternate character set of your design, and adds a keyword to activate that character set.

Necessary Accessories None

Supported Accessories N/A

Operating limits and warnings _____

_____ File name(s) _____

Size of file(s) _____ Additional RAM Requirement to run the program _____

References _____

This program has been verified only with respect to the numerical example give in Program Description. User accepts and uses this program material AT HIS OWN RISK, in reliance solely upon his own inspection of the program material and without reliance upon any representation or description concerning the program material.

NEITHER HP 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 HP 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.

71-00007

VARIABLE DEFINITIONS

NAME	DEFINITION
F\$	Name of LEX file to be created (1 to 8 characters)
V\$	VER\$ string of new LEX file (1 to 7 characters)
W\$	Name of new keyword (2 to 8 characters)
L1	LEX id # of new LEX file
L2	Token # of new keyword
L3	Character set id #
C	Length of character set (in bytes) (6 bytes per character)
T	Length of LEX file (in bytes)
P\$	Holds string of hex digits to be put into new LEX file
FNH\$	Returns a character representing the hex value of argument
FNS\$	Returns a hex string, 2 digits for each character in argument. The least significant nibble of the first byte of the argument occurs first, followed by the most significant nibble of the first byte. Successive bytes are appended after the first byte.
FNT\$	Same as FNS\$ except successive bytes are inserted in front of preceding bytes, thus reversing the order of the bytes.
Z\$	Used by FNS\$ and FNT\$ to hold value to be returned.

71-00007

SAMPLE PROBLEM

Create a character set by following the example in the HP-71 Owner's Manual on pages 133-135 (or create a character set of your own). When you are satisfied that you have the alternate character set as you like it (any number from 0 to 128 characters may be defined) run the example as shown on the following page.

When the example run has been completed the program will have created a new LEX file called TESTCH. To tell the system to look for the new LEX file, turn the machine off then back on.

Now display the VER\$ function and you should see the string TST embedded somewhere in the string:

```
>VER$  
HP71:1BBBB TST
```

To cancel the current alternate character set definition, type:

```
>CHARSET ""
```

The LEX file has added a new work (TESTCH) to the language. This keyword may be entered into a BASIC program or executed directly from the keyboard. To activate the character set, type:

```
>TESTCH
```

Now the character set is active. To display the first character in the special set type:

```
>CHR$(128)
```

The character set will remain active until the character set is redefined by another character set defining word is executed, the CHARSET statement is executed or the LEX file (TESTCHAR) is purged from system memory.

When the character set is activated, only 7 bytes of RAM is used in addition to the memory required to hold the LEX file. If the LEX file is in a ROM then only 7 bytes total is required to activate the character set.

71-00007

SAMPLE PROBLEM SOLUTION

DISPLAY CONTENTS	USER RESPONSE	COMMENTS
>	RUN CHARSET	
New LEX file name:	TESTCHAR	Any valid file name
VER\$ string:	TST	1 to 7 characters
LEX id #(decimal):	92	1 to 255
Token #(decimal):	5	1 to 255
Charset id #(decimal):	92	0 to 255

71-00007

(Continuation Page)

This program prepares a string of hex digits which it POKE's into a file. This string must be exact to prevent locking up the machine or causing a memory lost condition. For this reason, the user should not attempt to modify this program unless he/she is quite familiar with the internals of the machine and understands the cryptic detail of the program.

The user should be careful not to select a LEX id/token # that conflicts with some other application that he is likely to run. LEX id numbers in the range 92-94 have been set aside for just such use by users creating their own LEX files. The user should be aware of possible conflicts with any other locally written lex files. If the token # is also defined by another LEX file with the same LEX id, the results are unpredictable and certainly undesirable.

Hewlett-Packard has a process to allocate LEX id's and token numbers to users submitting programs to the Users' Library or burning application ROM's.

In addition to having a unique LEX id/token number, the LEX file must have a unique character set id. This number identifies which character set LEX file is active. Theoretically, up to 256 character set LEX files may be present in memory if they each have unique character set id's. It is probably a good idea to have the character set id match the LEX id if possible.

For details about how the LEX file implements the character set, see the HP-71 IDS Volume I.

71-00007

SYSTEM MODIFICATIONS

GENERAL FEATURES

Alternate character set _____
ASSIGN # _____
ENDLINE _____
EXACT _____
Files _____

FLAGS

BEEP ON/OFF _____
Beep volume _____
Math Exceptions _____
OPTION BASE/ROUND/ANGLE _____
Other system or user flags (include flag number) _____

STARTUP

Variables _____

Other _____

DISPLAY

CONTRAST _____
DELAY _____
FIX/SCI/ENG/STD _____
WIDTH _____
WINDOW _____

KEYBOARD

LC _____
Re-defined keys _____
USER mode _____

HPIL

ASSIGN IO _____
DISPLAY IS _____
PRINTER IS _____
PWIDTH _____
STANDBY _____

NOTES This program does not modify any general features, flags, start up, display, keyboard or HPIL parameters.

71-00007

7 of

```
10 ! CHARSET - Written by Bruce Stephens
20 ! Creates a LEX file that contains the current character set and adds
30 ! a keyword
40 ! that enables the character set.
40 DIM F#[8],V#[7],W#[8]
50 DESTROY ALL
60 INPUT "New LEX file name: ";F#
70 INPUT "VER$ string: ";V#
80 INPUT "Name of new keyword: ";W# @ W#=UPRC$(W#) @ IF LEN(W#)<2 THEN 80
90 INPUT "Lex id #(decimal): ";L1
100 INPUT "Token #(decimal): ";L2
110 INPUT "Buffer id #(decimal): ";L3
120 C=LEN(CHARSET$)/6
130 T=121+LEN(V#&W#)+6*C
140 DIM P#[T*2],Z#[C*12+16]
150 P#=FNS$(CHR$(L1))&FNS$(CHR$(L2)&CHR$(L2))
160 P#=P#&"00000f71000000"
170 P#=P#&FNS$(CHR$(20+2*LEN(W#)))
180 P#=P#&"000000"&FNS$(CHR$(144+2*LEN(W#&V#)))
190 P#=P#&"000D"&FNH$(2*LEN(W#)-1)&FNT$(W#)&FNS$(CHR$(L2))
200 P#=P#&"1ff969d031bf961"&FNS$(CHR$(40+LEN(V#)*2))&"0012b1351121C"
210 P#=P#&FNH$(LEN(V#)*2+1)&"1378b6ce13510b3"
220 P#=P#&FNH$(LEN(V#)*2+1)&FNS$(V#)
230 P#=P#&"0215d"&FNH$(LEN(V#)*2+1)
240 P#=P#&"0032bf38f1c8115e23010e0290a2217414b31"
250 P#=P#&FNS$(CHR$(L3))
260 P#=P#&"966311c47e501c330b15d000038d30350"
270 P#=P#&"9ffff2ffffd2307d532bfb8fd79114908d84a807e1017431"
280 P#=P#&FNS$(CHR$(L3))&"14d1cb30115d050e071450375ff"
290 P#=P#&FNH$(C*12)&FNH$(C*12 DIV 16)&FNH$(C*12 DIV 256)
300 P#=P#&FNT$(CHARSET$)
310 CREATE TEXT F#,(LEN(P#)+1) DIV 2
320 A=HTD(ADDR$(F#))
330 POKE DTH$(A+37),P#
340 POKE DTH$(A+16),"802e00"
350 STOP
360 DEF FNH$(N)=DTH$(N)[5,5]
370 DEF FNS$(S$)
380 Z$=""
390 FOR Z=1 TO LEN(S$)
400 Z#=DTH$(NUM(S#[Z,Z]))[5,5]&DTH$(NUM(S#[Z,Z]) DIV 16)[5,5]&Z#
410 NEXT Z
420 FNS#=Z#
430 END DEF
440 DEF FNT#[1536](S$)
450 Z$=""
460 FOR Z=1 TO LEN(S$)
470 Z#=Z#&DTH$(NUM(S#[Z,Z]))[5,5]&DTH$(NUM(S#[Z,Z]) DIV 16)[5,5]
480 NEXT Z
490 FNT#=Z#
500 END DEF
```

! Create a file of proper size

*! Poke hex code into file
! Change file type to LEX*