HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS
HOOSIER USERS GROUP HOOSIER USERS GROUP
USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER
GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS GROUP HOOSIER USERS

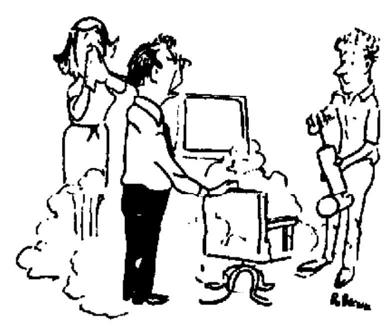
THE HUGgers HOOSIER USERS GROUP
People Helping People

July 1988 | THE HUGgers NEWLETTER | Volume 6, Number 5

TALKIN' SMART
by
JIM ELLIS
Part I

ATH,ATZ,ATAA,AT*R. You say it looks like a lot of bad printing errors? This is the first in a series of articles dealing with the commands that are used with smart modems. But, if you are a modem user you will know some of these commands and perhaps not recognize others. I will attempt to explain some of these and others in this series of articles. What is a smart modem you ask? The basic modem that some of us started out with did only two things. It either could send originate) or receive (answer) computer information using a 'phone line and had to be switched on and off by hand. How it does that is beyond the intent of this article. A smart modem, of which there are several in existance, contains ROM and sometimes RAM. The rom contains the software that permits your terminal program to TALK with the modem via the RS-232 card. The units that contain ram will let you make certain changes in the modem's program that are more to your need, within reason. For instance, it may give you the choice of how many rings before it answers the phone. That is only just one example, there are many others. One of the first, perhaps THE first smart modem was manufactured by Hayes Microcomputer Products, Inc. of Norcross, GA. This company set the standard as evidenced by the comment of so many modem ads that state 'Hayes compatible' in their attempt to sway you into buying their product. Some include the letters 'HC' as part of their model number. There are varying degrees as to how 'compatible' these modems are in their operation. Not all support every command or all parts of a command set. More on this later. Most modems that use ram contain a battery, so that some of the customizing that you do may be saved in non-volatile memory. Note, I said SOME but NOT all may be saved. (To be cont').

###############################
PASTE UP PARTY
HELP GET THE NEWSLETTER OUT
FIRST THURSDAY EACH MONTH
AT BOB STAHLHUT'S HOUSE
CALL 856-4962
###############################



"Look, when I said I wanted someone to debug the program, I meant . . ."

Reprinted from ROUGH NOTES *magazine*

****
SOUTH SIDERS MEETING
SECOND-----THURSDAY
AFTER THE MEETING
MONTHLY
CALL 786-3270
FOR LOCATION
****

Version 3.1 of Disk Labeler

Robert Neal, of Romeoville, Illinois, and the author of the original Disk Labeler program that appeared in last month's MICROpendium, writes to tell us of version 3.1. The code for this differs considerably from the version that we published, so we're including the entire listing here. Version 3.1 features are the ability to include the date that the label was printed, which is useful for determining the last time the label was updated. Also, the number of files on the disk is also printed. The user may also select which drive to catalog.

"I think you'll also find this version is much easier to modify in that all the printer codes are included as variables, therefore not requiring users to search through the code to find all the printer codes that need to be changed."

```
1 ! ************************ !235
2 ! ** DISKLABEL V3.1 ** !ØØØ
3 ! ** by ROBERT NEAL ** !148
4 ! ** TI Users Group ** !135
5 ! ** of Will County ** !186
6 ! **                  ** !Ø75
7 ! **  THIS PROGRAM  ** !Ø59
8 ! **    IS _FREE_    ** !213
9 ! ************************ !235
1ØØ DIM PN$(127),SZ$(127),PT$(127)!193
11Ø TYPE$(1)="D/F" :: TYPE$(2)="D/V" :: TYPE$(3)="I/F" :: TYPE$(4)="I/V" :: TYPE$(5)="PRO" !186
12Ø PL$="########## ### ### " !ØØ9
13Ø TL1$="Avail:#### Used:#### ## Files ##### ########
###" !254
14Ø TL2$="                                    #########
###" !237
15Ø OPEN #1:"PIO" !253
16Ø LF$=CHR$(27)&CHR$(65)&CHR$(6)!*** SETS LINE FEED TO 6/72 INCH !227
17Ø COND$=CHR$(15)!*** PUTS PRINTER INTO CONDENSED PRINT--17 CHAR/INCH !Ø63
18Ø SUBON$=CHR$(27)&CHR$(83)&CHR$(1)!*** TURNS SUBSCRIPT MODE ON !162
19Ø SUBOFF$=CHR$(27)&CHR$(84)!*** TURNS SUBSCRIPT MODE OFF !Ø42
2ØØ RESET$=CHR$(27)&CHR$(64)!Ø1Ø
21Ø DISPLAY AT(2,1)ERASE ALL:"           DISKLABEL":"          ==========": :"              by: Bob Neal":"           Version 3.1" !ØØ3
22Ø IF DATE$<>"" THEN 25Ø !127
23Ø DISPLAY AT(1Ø,1):"ENTER TODAYS DATE:   /" :: DISPLAY AT(11,2Ø):"-- --" !195
24Ø ACCEPT AT(1Ø,2Ø)SIZE(2):MO$ :: ACCEPT AT(1Ø,23)SIZE(2):YR$ :: DATE$=MO$&"/"&YR$ !178
25Ø DISPLAY AT(15,1)BEEP:"The Disk to be Labeled is in":"Which Drive (1-5)? Ø to QUIT" !114
26Ø CALL KEY(Ø,K,S):: IF S=Ø OR K<48 OR K>53 THEN 26Ø ELSE IF K=48 THEN PRINT #1:RESET$ :: CLOSE #1 :: STOP !Ø8Ø
28Ø K=K-48 :: DISPLAY AT(19,3):"Reading from Drive #";K !Ø42
29Ø OPEN #2:"DSK"&STR$(K)&".",INPUT ,RELATIVE,INTERNAL !133
3ØØ FOR X=1 TO CNT :: PN$(X),SZ$(X),PT$(X)="" :: NEXT X !185
31Ø CNT=Ø !156
32Ø INPUT #2:A$,J,J,K !156
33Ø A$=RPT$(" ",1Ø-LEN(A$))&A$ !169
34Ø FOR X=1 TO 127 !178
35Ø INPUT #2:PN$(X),A,SZ,S !Ø32
36Ø IF LEN(PN$(X))=Ø THEN 4ØØ !Ø14
37Ø SZ$(X)=STR$(SZ):: SZ$(X)=RPT$(" ",3-LEN(SZ$(X)))&SZ$(X)!236
38Ø A=ABS(A):: PT$(X)=TYPE$(A):: IF A=4 AND S=254 THEN PT$(X)=TYPE$(5)!Ø89
39Ø NEXT X !238
4ØØ CNT=X-1 :: CLOSE #2 !2Ø9
41Ø PRINT #1:SUBOFF$&COND$&LF$;:: PRINT #1,USING TL1$:STR$(K),STR$(J-K),CNT,DATE$,CHR$(14)&A$ !1Ø2
42Ø PRINT #1:SUBON$&RPT$("=",58):: LC=2 !178
43Ø FOR X=1 TO CNT STEP 3 !224
44Ø IF LC>9 THEN 45Ø ELSE 47Ø !Ø5Ø
45Ø PRINT #1:"":"" !139
46Ø LC=2 :: PRINT #1:SUBOFF$;:: PRINT #1,USING TL2$:CHR$(14)&A$ :: PRINT #1:SUBON$;:: PRINT #1:RPT$("=",58)!2Ø6
47Ø PRINT #1,USING PL$&PL$&PL$:PN$(X),SZ$(X),PT$(X),PN$(X+1),SZ$(X+1),PT$(X+1),PN$(X+2),SZ$(X+2),PT$(X+2):: LC=LC+1 !172
48Ø NEXT X !238
49Ø FOR X=1 TO 12-LC :: PRINT #1:"" :: NEXT X :: GOTO 21Ø !111
```

Lister revisited

Ray Kazmer, of Sylmar, California, provides the following file-listing program. It is the third to be published by MICROpendium (one was published last month), but it is the easiest to use and the

(See Page 3)

DISCLAIMER

This newsletter is brought to you through the efforts of the officers and members of the HOOSIER USERS GROUP. Every member is encouraged to submit articles.

If you have an article you would like to share with the other members mail it to:

John Powell
327 W. Southern Ave.
Indianapolis, IN 46225

Opinions expressed are those of the author and not necessarily those of the HOOSIER USERS GROUP.

(Continued from Page 2)

shortest. It runs in Extended BASIC. He writes:

I hate having to input types and lengths of files, which I'm trying to read. That junk is what my computer cares about, not me! It's all I can do to simply remember the name of the file. Why must we 'dim'bulbs' know that junk before we can pry into files?

File Lister 2 is a file reader I tossed together long ago in an hour. It's so user-friendly you don't even need instructions. I found several similarities in the code and the way this and Universal File Reader (March 1988, MICROpendium) are structured, and yet there are some notable differences.

Though PRESCAN_IT! is a great program, FL2 doesn't need prescanning. FL2 is so short (9 sectors) that it's up and running, waiting at the first prompt before your drive stops. The first prompt is for a drive number, 0-5. Zero is one way you tell FL2 to quit. Select drive 0, press Enter and FL2 blows itself out of memory. Drive 5 is usually used for a RAMdisk. When you input a drive number, FL2 looks for that drive and also checks to see if there's a disk in the drive. If either answer is 'no' you get a plain English error message.

Next, you're prompted for a filename, and that's all. If you must know the file type and length, just watch the screen and FL2 will tell you what they are. You then have the standard print option, Y or N, which can be entered in upper or lowercase.

FL2 also uses the spacebar to start and stop scrolling. Press once to stop, press again to restart or, if you hold the spacebar down, FL2 does a slow step scroll, one line at a time. This can be helpful if you must stop scrolling at a precise point. For example, to make a waxpaper digitizer 'screen map' of exact characters and their locations of TI-Runner screens.

You also press Enter to abort a listing with FL2 but, if you make a boo-boo and must redo the same file for another look, FL2 'remembers' the last filename entered.

```
100 CALL CLEAR :: @=0 :: [=1
 :: ]=2 :: _=3 :: \=4 :: CAL
L SCREEN(15):: CALL CHAR(128
,RPT$("F",16)):: CALL COLOR(
13,7,[):: Q$=RPT$(CHR$(128),
30):: CALL HCHAR([,[,128,32)
!058
110 D$="1" :: DISPLAY AT([,8
):" FILELISTER 2 ";:: DISPLA
Y AT(],8):"BY: RAY KAZMER":;
:"FOR D/V, D/F, I/F, I/V FIL
ES":;:"ENTER: DRIVE # (1-5;
0=QUIT)" !139
120 DISPLAY AT(8,8)BEEP:"DSK
"&D$&"." :: ACCEPT AT(8,11)V
ALIDATE("012345")SIZE(-[):D$
 :: IF D$="" THEN D$="1" ::
GOTO 120 ELSE IF D$="0" THEN
 350 !051
130 J$="  DEVICE ERROR  " ::
 ON ERROR 330 :: OPEN #[:"DS
K"&D$&".",INPUT ,RELATIVE,IN
TERNAL :: INPUT #[:F$,T,S,L
!134
140 DISPLAY AT(6,[)BEEP:"
    ENTER: FILENAME":;:"
   DSK"&D$&"."&N$ :: ACCEPT
AT(8,13)SIZE(-10):N$ !255
150 FOR X=[ TO 127 :: INPUT
#[:F$,T,S,L :: IF F$=N$ THEN
 170 ELSE IF LEN(F$)=0 THEN
320 !143
160 NEXT X :: IF F$<>N$ THEN
 320 !009
170 T=ABS(T):: IF T<[ OR T>\
 THEN J$=" WRONG FILETYPE "
:: GOTO 330 !004
180 ON T GOTO 190,200,210,22
0 !230
190 OPEN #]:"DSK"&D$&"."&F$,
SEQUENTIAL,DISPLAY ,INPUT ,F
IXED L :: DISPLAY AT(11,[):"
  FILE = DISPLAY FIXED";L ::
 GOTO 230 !013
200 OPEN #]:"DSK"&D$&"."&F$,
SEQUENTIAL,DISPLAY ,INPUT ,V
ARIABLE L :: DISPLAY AT(11,[
):" FILE = DISPLAY VARIABLE"
;L :: GOTO 230 !190
210 OPEN #]:"DSK"&D$&"."&F$,
SEQUENTIAL,INTERNAL,INPUT ,F
IXED L :: DISPLAY AT(11,[):"
  FILE = INTERNAL FIXED";L :
: GOTO 230 !168
220 OPEN #]:"DSK"&D$&"."&F$,
SEQUENTIAL,INTERNAL,INPUT ,V
ARIABLE L :: DISPLAY AT(11,[
):"FILE = INTERNAL VARIABLE"
;L !129
230 J$="PRINTER OFF-LINE" ::
 ON ERROR 330 :: DISPLAY AT(
14,[)BEEP:"SPACE = START/STO
P SCROLLING":;:"    ""ENTER""
 ABORTS LISTING":;:;:" ALSO
PRINT TO PIO? (Y/N) N" !179
240 ACCEPT AT(19,27)VALIDATE
("NYny")SIZE(-[):P$ :: IF P$
="Y" OR P$="y" THEN P=[ :: O
PEN #_:"PIO",VARIABLE 80 ::
CALL CLEAR ELSE P=@ :: CALL
CLEAR !077
250 IF EOF(])THEN 300 ELSE I
F T>] THEN INPUT #]:R$ ELSE
LINPUT #]:R$ !171
260 CALL KEY(5,K,S):: IF K=3
2 THEN 270 ELSE IF K=13 THEN
 340 ELSE 290 !253
270 FOR X=[ TO 150 :: NEXT X
 !127
280 CALL KEY(5,K,S):: IF K=3
2 THEN 290 ELSE IF K=13 THEN
 340 ELSE 280 !007
290 PRINT R$ :: IF P=[ THEN
PRINT #_:R$ :: GOTO 250 ELSE
 250 !099
300 PRINT :;:Q$&"          END O
F FILE.       "&Q$&" " :: DIS
PLAY AT(24,\):"LIST ANOTHER?
 (Y/N) Y" :: ACCEPT AT(24,24
)VALIDATE("NYny")SIZE(-[)BEE
P:X$ !184
310 IF X$="Y" OR X$="y" THEN
 340 ELSE IF X$="N" OR X$="n
" THEN 350 !057
320 J$=" FILE NOT FOUND " !0
16
330 DISPLAY AT(10,[):Q&"
 "&J$&"       "&Q$ :: CALL SOUN
D(225,220,@):: CALL SOUND([,
4E4,30):: FOR X=[ TO 1E3 ::
NEXT X :: DISPLAY AT(10,1):"
":"":"":"":"":"":"":"":"":""
:"" !014
340 ON ERROR 110 :: CLOSE #[
 :: CLOSE #] :: IF P=[ THEN
CLOSE #_ :: GOTO 100 ELSE 10
0 !107
350 CALL INIT :: CALL LOAD(-
31804,0,36)!054
```

MONTHLY MEETING LOCATION
ST.ANN'S SCHOOL
2839 S. McCLURE
INDIANAPOLIS,IN
MEETINGS OPEN AT
2:00 PM

Reprint from
QB MONITOR ~ QB-99'er NEWSLETTER

DISK CONTROLLERS - from TI to MYARC

The views expressed in this article reflect the author's personal experience with TI, Corcomp, and Myarc disk controllers. Technical data has been verified wherever possible, but is not publicly documented in some instances. Please bring any errors to the attention of the author.

The disk capacity of the TI99 has increased in just a few years from less than 80K (a single one-sided 35 track drive) to almost 2.9 megabytes (four double-sided, double-density, 80 track drives). The early standalone was replaced by the PEBox system which would support three double-sided 40 track drives (540K). Corcomp introduced their four drive double-density system (1440K), followed by Myarc's similar ..tem with two double-density formats .0K and 1440K). Then in 1986, Myarc offered its 80 track upgrade which doubled capacity again. Even as capacity was increasing rapidly, the TI and Corcomp controllers differed only modestly in I/O speed. When MYARC introduced its fast DSDD controller, few reviewers did justice to its speed advantage. Early comparisons were done at the standard TI or Corcomp interlace, but the big speed gains required taking advantage of the much tighter sector interlace possible with the high-speed MYARC card. To understand how this works we need to take a look at the way a disk drive performs.

Disk Drive Fundamentals

A floppy disk drive writes information in concentric rings called "tracks" on a thin plastic disk coated with a film of ...netic particles. Each track in turn is divided into blocks of information called sectors. A blank disk has one (or more) index holes used to synchronize the process of writing to and reading from the disk. The type with many holes are called "hard sectored" since each sector has its position fixed by an index hole. The type of disks used by most computers have only one hole and are called "soft sectored". In this system the computer must write magnetic signposts on the disk to mark out each sector in a process called "formatting" or "initializing" a disk. These signposts take up a subtantial fraction of the space on a track since they include not only sector numbers but buffers (filler bytes) that allow the computer to get into synchronization to read or write sectors of data and to prevent the sector identifier from being overwritten by a drive operating at a slightly different speed from the drive that formatted the disk.

The typical 5.25 inch disk drive has a "stepper motor" capable of moving the drive's read/write head(s) in or out along a radius of the disk in steps of 1/48 of an inch (thus the terminology "48 tpi" = 48 tracks per inch). Since the inner tracks have a smaller circumference, they crowd the bits of information together. Magnetic coatings on a floppy disk are rated by their capacity in bits per inch at standard magnetic flux for the write head. This figure is usually over 5000 bpi for modern floppies, but was somewhat lower a few years ago. The circumference of the inner track of a 40 or 80 track disk is about 10 inches -- which allows about 6250 bytes to be written on the track without exceeding 5000 bpi. For comparison, the Corcomp double density format requires over 6400 bytes per track. Media limitations were the reason that some early 5.25 disk drives only used the outer 35 tracks. The 16 sector (by 256 bytes/sector) format recommended by most drive makers requires only 6250 bytes per track and includes several hundred additional "buffer" bytes to compensate for differences in drive timing.

Timing is EVERYTHING

With soft-sectored disks, the integri of the read/write processes requi critical timing. The disk rotates 300 rpm within a small margin. Th means there are about 250 thousa magnetic pulses (bits) passing benea the head each second. In single densi format, the majority of these pulses a timing or filler bits — in doub. density, many of the timing bits a suppressed in order to double the ra of data bits. In a typical sector rea the drive must bring the disk up t speed, recognize the index hole, st out to track zero (to get its bearings) determine single or double density verify its position, step in to th target track, verify the track numbe (written in the format operation) detect the sector identifier as it flie past, then immediately read the 256 dat bytes into memory. Five of these operations require accurate reading of the magnetic pulses whizzing by at over 230K bits per second.

If you do some quick arithmetic (256 bytes/sector = 2048 bits/sector into 250k bits/second)... hmmm... Why can't the drive read a 125 sector file in one second? Well first many of those bits are not data bits, they are overhead to keep things synchronized and allow for timing variation between drives. Second, some time is used moving the head from one track to the next when more than one track must be read. Third, 250K is the instantaneous read rate and the computer must take time to do other things like move the last sector out of its buffer to make room for the next one. In the standard TI protocol for reading a disk, the data is moved into VDP ram (so the drive could be used without the memory expansion) before it goes to the expansion memory. All this thrashing eats great chunks of the time available for reading data. By the time one sector is safely tucked away in the 32K card, several sec... have already passed by the drive's read head. If the sectors were written

DISK CONTROLLERS - Cont'd. Pg 2

consecutively on the disk, we would have to wait a full revolution (0.2 seconds) before the next sector would pass under the head. To avoid this inefficiency, the consecutively numbered sectors are spaced out around the disk so that they are separated by just enough time to take care of other business. The actual pattern in which the sectors are scattered is called the "interlace". The idea of the interlace is to spread the sectors out to match the timing needs of the hardware — both the time needed to stash each sector and the time needed to step from one track to the next and get the the head settled down for some serious (250K bps) reading.

Interlace and Head Step Times

Life was simple with the TI disk controller. Both the interlace and the head step time were locked into the controller's PROM (that's the programmable chip that contains the control programs for the card). The head step time is the built-in delay between step signals to allow the stepper motor to move the head one "click" in or out. The TI settings are very conservative (read "slow") to allow for slow drives. The step time is 20ms -- if you step from track zero to track 39, it takes 20x39=780ms, almost four revolutions of the drive. The TI interlace lays the sectors down on a track in the order 075318642. This allows all sectors to be read in four revolutions of the disk though the slow head step lets another revolution go by between tracks. Thus the maximum read rate is about 9 sectors per five revolutions (= one second) or 2304 bytes per second.

When Corcomp designed its double density disk controller, allowances were made for the increased speed of later drives by permitting the step rate to be set with DIP switches for each drive. The step rates available are 30, 20, 12, and (the faster values quoted in the CC manual are referenced to the wrong clock speed). They also provided a choice of interlace options, though only a couple of them are practical. The default interlaces are labeled "7" for single density and "10" for double density. The single density interlace is the same as TI's, but with a faster step setting the head be can moved without losing a revolution and thus reads 20% faster than the TI controller. The double density interlace allows 18 sectors to be read in five revolutions, but it doesn't leave enough margin to stash the last sector and step the head in time to catch the zero sector of the next track (that's why the sector number "hangs" for 0.2 seconds each 18 sectors while verifying a formatted disk — you are seeing the extra revolution needed to acquire the first sector of the next track). Thus the maximum read rate is 18/1.2 or 15 sectors per second, about 67% faster than the TI controller. Users of the CC controller have probably noticed that it loads its own MANAGER program faster than this. In this case a special loader bypasses VDP and loads directly to CPU RAM — this faster handling of the data allows the steppor motor to be activated sooner and saves one revolution per track (so the 98 sector file can be read in about 5.5 seconds). This provided a foretaste of the speed that MYARC would achieve with its double density controller.

The MYARC controller bypasses VDP RAM to load directly to CPU RAM. This technique coupled with a buffer RAM chip on the controller card provided a quantum jump in disk I/O speed. The MYARC card reads the TI single density interlace at 11.25 sectors/second (the same as Corcomp) and reads the CC 18 sector/track interlace at 18 sectors/second (the same speed Corcomp reads its MANAGER program), but this is only the beginning. Since the hardware empties its sector buffer faster, consecutive sectors can be placed closer together allowing a track to be read in fewer revolutions, i.e., it supports a faster interlace. With fast drives, the 9 sector/track single density format can be read at interlace "2". (NOTE: In the MYARC terminology, the interlace number represents the number of disk revolutions required to read a track. This works out to 22.5 sectors/secor compared to 9 for the TI and 11.25 fc the CC controller. The MYARC 16 sectc format can be read at interlace "3" 26.67 sectors/second -- 3 times as fas as the TI controller and almost twice a fast as Corcomp double density. Th Corcomp 18 sector format can be read a interlace "3" or "4", but the data rat is the same in either case, 22. sectors/second. Interlace "4" is smoot but requires a very quick head step. interlace "3" reads the track in revolutions but forces an extra revolution for the step from track tc track because sectors 17 and 0 are adjacent on the disk. Though both interlaces have the same data rate, interlace "3" is safer if you are uncertain about the speed of your stepper motor.

In order to read and write both double density formats, the MYARC system must insert an additional step in some I/O operations -- sector zero must be read to determine whether a double density disk has 16 or 18 sectors per track. This datum is needed to convert the the logical sector numbers used by the TI operating system into track and sector-within-track addresses for the floppy disk controller chip. The TI and Corcomp controllers do not need this step because they do not use the full potential of the TI disk I/O protocol. Once this step, accessing sector zero, is added to the various disk operations, it opens the system up for using more than two formats -- including 80 track formats.

"Where do you keep the computer [illegible]?"

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222
Forwarding and Address Correction Requested

TIME DATED MATERIAL
JULY 17 1988

88/12 7725S
Steven Sims
212 South Hancock
Indianapolis, IN 46222

APPLICATION FOR MEMBERSHIP

Below you will find an application for membership to the Hoosier Users Group. Active membership entitles you to the Newsletter, up and download on the HUGbbs, attendance and voting rights at regular club meetings, access to the HUGger Library of Programs, special club activities and special guest speakers for one year. Subscribing members will receive the **NEWSLETTER** only.

Make check or money order payable to **Hoosier Users Group.** Send completed application to:

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222

(Cut on dotted line)

Check One:

Active Member

New: $20 ______
Renewal: 15 ______

Subscribing Member

New: $10 ______
Renewal: 7.50 ______

Amount Enclosed: $ ______

______ D ______
S ______

Name: ______________________ Today's Date: ____________

Address: ______________________ Apt. # ______

City: ______________________ State: ______ Zip: ____________

Phone: (______) ______-__________

Interests/Comments: ______________________________

____