HELPFUL HINTS IN USING YOUR NEWDOS/80 VERSION 2.5

We suggest using the following check-list as a guide to setting up software on your Hard Disk System:

1. Carefully read through all related documentation.

2. Make hardware installation as directed by instructions supplied with your Hard Drive Unit.

3. Boot on your NEWDOS/80 Version 2.5 original master diskette and make backup copy or copies. Refer to Chapter 1, Section 1.4 of the NEWDOS/80 Version 2.0 manual for details if unfamiliar with the procedure.

4. Designate one of your backups as a working copy an d boot (or reset) on it. Use this diskette for the remainder of this procedure.

5. Use HDFMTAPP/CMD. Refer to Section 4 in the NEWDOS/80 Version 2.5 manual (Appendix C).

6. Set up your PDRIVE definitions for the Hard Drive Volumes according to Section 3 & 6 of the NEWDOS/80 Version 2.5 manual.

7. Use DOS command FORMAT on each defined volume of the Hard Disk. This has not have very clear to some of our users. HDFMTAPP/CMD is only a "media format" utility that allows the hard drive to be used with NEWDOS/80 (or other DOS system). DOS command FORMAT, although does not prepare the media as HDFMTAPP/CMD, is important for checking the media and preparing the directory (DIR/SYS) and the boot sectors (BOOT/SYS) on the hard drive volumes. If the volume was not prepared with FORMAT, it will seem to operate correctly with COPY, OPEN, and CLOSE (ie. normal file operations), but HDBACKUP/CMD and DIRCHECK/CMD will not funtion.

8. (Optional) Move NEWDOS/80 Version 2.5 to Hard Disk Volume. Refer to Section 5 of the NEWDOS/80 Version 2.5 manual. Don't forget to move your PDRIVE definitions to appropriate new slots. Follow prodedure to create Boot Diskette.

9. Install application software and related data files.

NEWDOS/80 VERSION 2.5 --- THE NEWDOS/80 VERSION 2 HARD DISK SYSTEM

The NEWDOS/80 Version 2 modified for hard disk operations is called both the NEWDOS/80 Version 2 Hard Disk Operating System and NEWDOS/80 Version 2.5. The difference between the regular NEWDOS/80 Version 2 and Version 2.5 is the inclusion in 2.5, via patches, of code to handle the hard disk. This documentation for NEWDOS/80 Version 2.5 is considered as Appendix C to the regular NEWDOS/80 Version 2 manual and should be inserted into that manual after Appendix B.

NEWDOS/80 Version 2.5 is NOT offered as a stand alone DOS; the regular NEWDOS/80 Version 2 must be PURCHASED and REGISTERED either prior to or at the time of purchase of NEWDOS/80 Version 2.5.

As usual with NEWDOS/80, the user should study this document carefully before attempting to do anything with the NEWDOS/80 Version 2.5 Hard Disk Operating System.

The basic NEWDOS/80 Version 2.5 supports Apparat's and Tandy's hard disks for either the TRS-80 Model I or III. Special Version 2.5 system diskettes may later be made available for other types of hard disks.

You may have already been using your hard disk under LDOS or another DOS and have valued user files on the hard disk. If this is so, you are already a serious hard disk user and cannot afford to lose valued data just because you are switching DOSs. You must:

1. CAREFULLY plan your move.

2. Backup up those files from hard disk to diskette using that DOS's offload program. This is insurance in case the conversion to NEWDOS/80 fails; you can reformat the hard disk(s) for the other DOS and reload the files.

3. Move NEWDOS/80 Version 2.5's HDBACKUP/CMD program over to that DOS (see the NND parameter discussion in section 7).

4. Use HDBACKUP under that DOS to again offload your files to another set of diskettes. Remember to use the SAVE, INCLUDE and NND parameters. Also remember, for 5 million bytes of data, the HDBACKUP SAVE function will need 27 pre-formatted single sided, double density 40 track (720 sector) diskettes.

5. Initialize the hard disks for NEWDOS/80 Version 2.5 using HDFMTAPP, PDRIVE and FORMAT.

6. Use HDBACKUP under NEWDOS/80 Version 2.5 to RESTORE the files from

diskettes to the hard disks. Carefully plan this move; you may decide to use more than one RESTORE from the same backup to get the various files where you want them.

1. OVERVIEW

1. A data file may contain as many as 16 million bytes.

2. A hard disk data volume can be up to 65535 sectors and contain up to 246 user files.

3. PDRIVE allows a maximum of eight active slots with a maximum of 4 floppy data volumes or eight hard disk volumes active at one time.

4. The capacity to support hard disk drives of over 100 million bytes exists, though currently only the Apparat and Tandy hard disk drives are supported.

5. A hard disk drive is divided into one or more drive sections.

6. A hard disk section is divided into one or more data volumes. The data volume is what is defined by PDRIVE. A data volume may not span multiple hard disk drives or drive sections.

7. 48K of RAM is required. The hard disk modifications for NEWDOS/80 Version 2.5 have preempted computer main memory 0F900H - 0FFFFH. Programs that execute in that area must no longer be used.

8. Aside from the main memory limitation above, most programs that work with NEWDOS/80 Version 2 will work with NEWDOS/80 Version 2.5. However, if any program assumes certain volume sizes (i.e. 350 sectors on the Model I or 720 sectors on the Model III) or a certain location and size of the directory, that program will have to be modified. Basically, programs that use standard file I/O and observe HIMEM should be OK.

9. The hard disk system can operate either from floppy drive 0 (in which case floppies retain their old drive numbers) or from the hard disk (in which case, floppy drives 0 - 3 become drives 4 - 7 respectively). Section 5 steps the user through the shift of the system from a floppy to a hard disk volume.

10. Program HDFMTAPP/CMD is used to magnetically format Apparat and Tandy hard disks.

11. Program HDBACKUP/CMD is used to selectively save hard disk files of any size onto floppies or to selectively restore them to hard disk. This is NEWDOS/80 Version 2.5's hard disk backup facility and must be used to make backup copies of valued data files. Further, the program HDBACKUP/CMD can be transferred to another DOS (see NND parameter in section 7) so that the program can be used under that DOS to offload user files to diskette preparatory to changing the hard disk to operate under

NEWDOS/80 Version 2.5 or to onload user files from diskette to hard disk should the user wish to take the hard disk back to the other DOS.

12. Program EXTPDRIV/BAS can be used to set PDRIVE slot definitions from definitions stored in an ASCII text file. Since hard disk data volume specifications are both difficult and critical, it is recommended they be permanently built in a text file via SCRIPSIT or CHAINBLD and then activated when needed via EXTPDRIV.

13. Parameter HDS has been added to DOS command PDRIVE to define hard disk volumes.

14. Hard disk volumes defined under Model III NEWDOS/80 can be used under Model I NEWDOS/80 and vice versa. The files on these volumes are NOT useable interchangeably if they were NOT useable interchangeably when those files were on diskettes.

2. COMMENTS and RESTRICTIONS:

1. The user must be knowledgeable of NEWDOS/80 Version 2 and all subsequent information issued via the zaps prior to attempting to use the Hard Disk system. All hard disk discussion herein assumes this knowledge. This document is intended only as supplementary information to the regular NEWDOS/80 Version 2 manual and its subsequent zaps.

2. This document does NOT provide information about your hard disk. That information must be obtained from the source where you purchased or otherwise obtained your hard disk. The information NEWDOS/80 needs to know about your hard disk drive is (1) the number of recording surfaces (or number of I/O heads), (2) the number of tracks per surface (or the number of cylinders), (3) the number of 256 bytes sectors actually formatted on each track, and (4) track-to-track stepping rate code.

3. NEWDOS/80 Version 2 was not designed to operate with hard disks. All of the changes creating Version 2.5 have been done by patching the standard Version 2, with the exception that SYS0/SYS has been extended five sectors. This patching to Version 2 provides a minimum hard disk operating system and each specially purchased hard disk system diskette will operate with one and only one type of hard disk drive. If another type of hard disk drive has exactly the same interface to the computer, then it can also work with a particular hard disk system diskette (example, both Tandy's and Apparat's hard disks for the Model I and Model 3 have the same software interface, therefore either (but not both at the same time) can be used with the same Version 2.5 Hard Disk system diskette). The standard issue NEWDOS/80 Version 2.5 hard disk system supports Apparat's and Tandy's hard disks for either the TRS-80 Model I or III.

4. This system REQUIRES 48K of RAM. To implement hard disk code, main memory from F900H to FFFFH has been taken by DOS and is not available to the users. Any user programs that use this area MUST either no longer be

used or be modified to no longer use the F900H to FFFFH main memory area. HIMEM is set to 0F8FFH by DOS automatically and programs that observe HIMEM should be all right.

5. The number of active PDRIVE slots (formerly called drives) has been expanded from 4 to 8, allowing a maximum of 4 floppy volumes, 8 hard disk volumes or a combination thereof. The number of actual active PDRIVE slots is still controlled by SYSTEM option AL. If a PDRIVE active slot is to be unused, defining it as a hard disk volume and setting the PDRIVE HDS subparameter vscl to 0 will cause PDRIVE to accept the definition in an active slot and NEWDOS/80 to treat the slot as DEVICE NOT AVAILABLE whenever it attempts to use that slot (drive).

6. Two floppy drives are desirable, though only one is required. The Version 2.5 Hard Disk System comes on a standard 40 track, double density diskette on the Model III and a standard 35 track, single density diskette on the Model I. Since the capacity of the Model I diskette is too small to contain all the files for the hard disk system as well as those of the non-hard disk system, some of the files (or program modules) of the regular NEWDOS/80 are not present on the hard disk system diskette. When needed, you may copy these modules over from the regular NEWDOS/80 system diskette.

7. The user may elect to run using a floppy system diskette (a copy of the Version 2.5 Hard Disk System diskette) or he/she may move the NEWDOS/80 Version 2.5 Hard Disk System onto a hard disk volume.

    1. If the system is being run from floppy drive 0 (the normal drive 0 for the computer), the floppy drives 0 - 3 use PDRIVE slots 0 through 3 respectively, just as they do in the regular NEWDOS/80 Version 2. Under the floppy hard disk system, PDRIVE slots 1 - 7 may be defined as hard disk volumes and accessed by user programs as drives 1 - 7 respectively.

    2. If the system is being run from hard disk, the floppy drives 0 - 3 use PDRIVE slots 4 through 7 respectively, and the floppy drives 0 - 3 are known to the system and user programs as drives (or slots) 4 through 7 (though you may use any or all of drives (or slots) 0 through 7 for hard disk volumes).

        ***** Warning, when the system is being run from the hard disk, access to the floppy drives is slots 4 - 7 meaning that all slots between the system volume in slot 0 and the slot being used by the floppy drive MUST be valid definitions even though you are using only one hard disk volume. PDRIVE will allow and DOS will ignore a slot defined for a null hard disk volume (having the HDS subparameter vscl equal 0), thus allowing access to the floppy drives.

8. The DOS command FORMAT or the format portion of COPY do NOT actually format the hard disk; instead of formatting, the message, INITIALIZING SECTORS, is displayed and the sectors are written with a standard pattern. To actually format the Apparat or Tandy hard disk, use the HDFMTAPP/CMD program provided. To format another hard disk, you must use a program provided by the hard disk retailer (NOT provided by Apparat).

9. This hard disk upgrade does NOT support the standard LDOS TRS-80 hard disk data volumes as directory concepts slightly differ, though those volumes can be read via SUPERZAP by expert users (provided the hard disk is divided into sections properly, spgl value is 16, the ddsl1 value is 76, the ddsal value is 32, and the gpl1 value is 2 for one surface volumes, 4 for two surfaces, 6 for three surfaces and 8 for four surfaces)(changing HIT sector rel byte 1FH from 00H to 16H allows DIR to work and many other functions marginally)(you are on your own processing LDOS volumes under NEWDOS; don't call Apparat when you get into trouble). Basically, when shifting from one DOS to another, the user must off-load the hard disk files to floppies under that DOS using the NEWDOS/80's HDBACKUP program and bring them back in under the other DOS using NEWDOS/80's HDBACKUP program (the NND parameter must be used when the DOS is other than NEWDOS/80).

10. A number of user programs read and interpret the directory. If that program was reading the directory as the DIR/SYS file, observing the protected sector error code and observing EOF, there should be no problem. If the program was using the DDSL value in the data volume 1st sector to compute the directory location, the program will fail unless the data volume has spgl = 5. If the program was assuming the location and size of the directory, it will most probably fail!!!

11. A data volume must not exceed 65535 sectors. Aside from the space used by BOOT/SYS and DIR/SYS on that data volume, all the remaining space may be allocated to one file, over 16 million bytes. The sector range assigned to one data volume must NOT overlap that of any other data volume; it is the user's responsibility, through careful PDRIVE definition of the data volumes, to avoid this overlap, which can be quite disastrous.

12. A hard disk drive is logically divided into one or more data volumes via judicious use of the PDRIVE HDS parameter. Though a data volume is limited to a maximum of 65535 sectors, a hard disk drive is limited ONLY by its actual capacity AND the limitations that Sectors Per Track (SPT or spgl) must be less than 256, Tracks Per Cylinder (TPC)(or RSC (Recording Surface Count)) must be less than 256, Tracks Per Surface (TPS or tpsl) must be less than 65536, and TPS times TPC must be less than 65536.

13. A hard disk physical drive's space may be divided into drive sections. Normal NEWDOS/80 Version 2.5 operations DO NOT require this. However, if your division of the hard disk is to be such that part of the hard disk is to be used for data volumes of another DOS (such as LDOS) which assigns data volumes in units of one or more entire recording surfaces, it is necessary to sectionalize your hard disk under NEWDOS/80. This is done by setting the PDRIVE HDS subparameter sscl value to the number of recording surfaces assigned to that drive section and by setting the sfsl value to the relative number of the first recording surface assigned to that drive section. For a given drive, no two sections may share the same recording surface, and no data volume may have space assigned from more than one drive section.

14. For DOS command COPY, the =tcl parameter is not legal if the SOURCE is a hard disk data volume. For FORMAT and COPY, the =tc2, DDSL and DDGA parameters are not legal if the DESTINATION is a hard disk data volume.

For FREE and the header of DIR, to avoid ambiguity, a track count of 0 is displayed if the data volume is on a hard disk.

15. The SUPERZAP displays may look awkward as they were not designed to handle over 9999 sectors. However, they do work, excepting that TRK and SOT values are not displayed for sectors on hard disk. The DTS main menu function is not allowed for hard disk volumes.

16. Format 5 COPY (full diskette COPY) requires that SOURCE and DESTINATION have the same GPL and SPG values and, if the destination is on a hard disk, the same ddsll and ddsal value. Otherwise format 6 COPY (Copy By File) must be used.

17. Hard disk volumes defined under Model III NEWDOS/80 can be used under Model I NEWDOS/80 and vice versa if NEWDOS/80 supports the drive for the Model I and III. The files on these volumes are NOT useable interchangeably if they were NOT useable interchangeably when those files were on diskettes (such as system program and most user, non-BASIC program files). If you intend to use a hard disk with both your Model I and your Model III (though not at the same time) and intend to run the system from that hard disk, you should create two system volumes on the hard disk, one for the Model I and one for the Model III.

18. ****** Errors may occur in DIRCHECK and SUPERZAP if DFG (MINI-DOS) or 123 (DEBUG) are used during the program's execution and the target drive is not explicitly respecified after conclusion of MINI-DOS or DEBUG. After MINI-DOS or DEBUG in SUPERZAP, it is recommended that you return to the main menu or do the 'J' display function; for DIRCHECK, respond Y or N to the menu.

3. CHANGES TO PDRIVE for hard disk operation.

No existing parameters in PDRIVE have been changed (so floppies are defined exactly as before), and one parameter, the HDS parameter, has been added to accommodate the hard disks.

The TRS-80 diskette directory was originally intended for 35 or 40 track diskettes of 350 to 400 sectors. In NEWDOS/80 Versions 1 and 2, the directory was modified somewhat to allow for a maximum of 222 user files instead of 62 and allow a maximum of 1536 granules instead of 192. To get these extra granules, the granule lockout table was eliminated from the GAT sector and number of granules per lump (GPL) was expanded from the old implied value of 2 to a user specified value with a maximum value of 8. At 5 sectors per granule, this allowed for 7680 sectors (1,966,080 bytes) per data volume.

However, with hard disks, we really want the capability of allowing a volume to be up to 65535 sectors and a file to be not much less than that. In order to retain the same directory structure but increase the number of sectors for a data volume, we have changed the number of Sectors Per Granule from the old implied value of 5 to a user specified value of not more than 255. Theoretically, this should allow for 1536 * 255 = 391,680 sectors, but there is another governing restraint, that of the NEXT and EOF fields of the directory FPDE and the file's FCB. These fields allow for a maximum of 65535 sectors (if wrap around is to be avoided). Normally this restriction limits the size of a file, but actually this restriction limits a data volume's size since NEWDOS/80 has a special use of the FCB that allows sector I/O directly to a data volume, bypassing the file concept altogether. Therefore, the NEWDOS/80 version 2.5 hard disk system limits a data volume to 65535 sectors (16,776,960 bytes). Since each volume has a BOOT/SYS file and a DIR/SYS file, the maximum size of a user file is somewhat less than 65535 sectors.

Though 5, 10 or 15 million byte hard disk can be treated by NEWDOS/80 as one data volume, it is generally desirable to divide a hard disk into more than one data volume. NEWDOS/80 allows the user great flexibility in this, admittedly at a cost of complexity (as usual with NEWDOS/80's PDRIVE which many users are still uncomfortable with). A PDRIVE slot definition actually specifies a data volume, not a floppy drive or a hard disk drive or a hard disk drive section. The specifications for the drive and, optionally, drive section are simply part of the specifications of a data volume.

The definition of hard disk data volumes is more difficult and more critical than for floppy diskette data volumes. The user is solely responsible of assuring that a hard disk sector is NOT shared by two or more data volumes. As an aid to the user, the BASIC program EXTPDRIV/BAS has been provided to search an ASCII text file for a specified definition and assign the definition to a specified PDRIVE slot. Using SCRIPSIT or CHAINBLD, the user can carefully and permanently build his/her hard disk data volume definitions (actually just the HDS parameters), and later, when a particular data volume is needed in a particular PDRIVE slot, EXTPDRIV can be used to effect this assignment.

Further, NEWDOS/80 Version 2.5 does NOT maintain a table of bad hard disk sectors. If your hard disk has bad sectors, you must either operate that drive with a sufficiently reduced SPT (sectors per track) value or you must define data volumes such that the bad sectors at not included within any data volume.

Since a hard disk data volume's definition has more values than for a floppy data volume, and we want to limit each slot's definition to one line on the display, we have decided to combine all 12 values of a hard disk data volume specification into one parameter, the HDS (Hard Disk Specification) parameter. The 12 values are called sub-parameters; ALL 12 MUST be given EACH time the HDS parameter is used, and all must be in the exact order specified. The specification of the HDS parameter is:

```
HDS=(hddnl,tpsl,sfsl,sscl,sptl,tsrl,vfsl,vscl,spgl,gpll,ddsll,ddsal)
```

where:

1. hddnl means Hard Disk Drive Number and is the relative number (0 - 3) of the drive on the hard disk cable with 0 being the first drive. hddnl specifies which physical hard disk drive the data volume is on.

2. tpsl means Tracks Per Surface and is the number of tracks per recording surface (also the number of cylinders) for the hard disk drive. Each recording surface of the drive has tpsl number of tracks. tpsl is an integer from 0 to 65536. For Apparat hard disks, tpsl = 306. For Tandy 5 Meg hard disks, tpsl = 153.

LDOS 5.1.3 appears unable to support the tpsl value of 306 used with Apparat's hard disk. However, an Apparat 10 Meg hard disk (with tpsl = 306 and RSC = 4) can be used as a 5 Meg hard disk with LDOS 5.1.3 where implied values of tpsl = 153 and RSC = 4 are used.

3. sfsl means Section First Surface and is the relative number of the first surface of the hard disk drive assigned to the drive section containing the data volume. sfsl is an integer between 0 and RSC-1. If you are not sectioning your hard disks, sfsl will always be 0.

RSC means Recording Surface Count and is the number of recording surfaces for the hard disk. Another term for the number of recording surfaces is TPC (Tracks Per Cylinder). For Apparat 5, 10 and 15 Meg hard disks, RSC is 2, 4 and 6 respectively. For Tandy 5 Meg hard disks, RSC is 4.

4. sscl means Section Surface Count and is the number of consecutive surfaces of the hard disk drive assigned to the drive section containing the data volume. sscl is an integer between 1 and RSC with the sum of sfsl and sscl not greater than RSC. If you are not sectioning your hard disks, sscl will always equal RSC.

5. sptl means Sectors Per Track and is the number of 256 byte sectors on each track of the hard disk drive which in turn is the number of sectors formatted on each track by the format program supplied with your hard disk (for Apparat and Tandy hard disks, this is the HDFMTAPP program). sptl is an integer from 1 to 255. Normally, Apparat and Tandy hard disk drives have 32 sectors per track; however, if during HDFMTAPP formatting of the hard disk, a track is found with more than one error sector, it will be necessary to format the hard disk with less than 32 sectors per track unless you intend to define data volumes to bypass the bad sectors; remember, NEWDOS/80 does NOT maintain a hard disk bad sector table.

6. tsrl means Track Stepping Rate and is a code used by DOS to send track-to-track stepping rate information to the hard disk controller when it is necessary to move the disk arm which contains the read/write heads. tsrl is an integer between 0 and 255. Apparat hard disk use tsrl = 0. Tandy 5 Meg hard disks require tsrl = 6.

7. vfsl means Volume First Sector and is the relative sector number within the drive section of the data volume's first sector (the data volume's relative sector 0). vfsl is an integer between 0 and 16,777,215 with an effective upper limit of one less than the number of sectors assigned to the drive section (if a hard disk is not sectioned, the hard disk is one in the same as its one section). If vfsl = 0, then the data volume's sector range starts with the first sector of the drive section; further, if both vfsl and sfsl are 0, the data volume's range starts with the drive's 1st sector.

8. vscl means Volume Sector Count and is the number of consecutive sectors of the drive section, beginning with sector vfsl, assigned to this data volume. vscl is an integer between 0 and 65535 but the sum of vfsl and vscl must not exceed the number of sectors assigned to the drive section (which is tpsl * sscl * sptl). If vscl is simply the asterisk (*

) character instead of an integer, PDRIVE will assign all of the drive section's remaining sectors to the data volume.

*********** IMPORTANT. PDRIVE will accept a vscl value of 0, meaning a null data volume, and it will allow the data volume definition into an active slot (provided the definition has no other errors). If vscl is 0, DOS will generate a DEVICE NOT AVAILABLE error whenever a slot is selected that contains this data volume. This is needed as a way of filling in PDRIVE slot definitions so that access can be made to slots 4 - 7 for floppy diskette operations when running the system from hard disk and not all of slots 1 to 3 are defined for valid hard disk data volumes.

The sub-parameters hddnl, tpsl, sfsl, sscl, sptl, vfsl and vscl combine to define a unique range of hard disk sectors assigned to the data volume. No sector in this range may be shared by another data volume defined by PDRIVE; it is the user's responsibility to avoid this conflict. Otherwise, the same sector can end up being used for two different purposes.

9. spgl means Sectors Per Granule and is the number of sectors in each allocation granule for this data volume. spgl is an integer between 1 and 255. If spgl = 0 is specified, PDRIVE will compute the lowest spgl above 4 that will suffice for the gpll value specified and the number of sectors assigned to the data volume (vscl).

When DOS assigns disk space to a file, it does so in minimum units called granules; so the spgl value defines the minimum number of sectors allocated to a file and also is one more than the maximum number of sectors that a file will have allocated beyond its needs. Generally, it is desirable to have a small spgl value, but the smaller the spgl value, the smaller the maximum size a data volume may be. In the regular NEWDOS/80 Version 2, a SPG value of 5 was implied and always used, except in some of the COPYs to and from special TRSDOS diskettes. If full diskette COPY (not CBF) compatibility is wanted with the floppies, spgl = 5 must be used as that is the standard in the NEWDOS/80 Version 2 floppy world.

****** Warning, when a NEWDOS/80 system volume is being COPY'ed using CBF and the destination spgl value is less than the source spgl value, DISKETTE GAT OVERFLOW error may occur. The only alternative is to copy the system from the hard disk system diskette and use a destination spgl greater than 4.

10. gpll means Granules Per Lump and is the maximum number of allocation granules for each byte in the data volume directory's Granule Allocation Table in the GAT sector (the first sector of the directory). gpll is an integer between 2 and 8. GPL = 2 is the standard for the old Model I TRSDOS 2.3, and the NEWDOS/80 Version 2 master diskettes use GPL = 2. However, any data volume, whether hard disk or floppy, with more than 1920 sectors, should use a larger GPL under the criteria that it is better to increase the GPL that the SPG. It is recommended that if GPL = 2 is not used, then use GPL = 8. Though the other values are legal, don't use them unless you are attempting compatibility with another DOS.

A lump???? For NEWDOS/80 Version 2, we wanted to eliminate the one-to-one correspondence between a byte in the GAT table and a diskette (or hard disk) track or cylinder so that granules could flow across track and cylinder boundaries. A granule's allocation state is handled by one bit in the GAT, and we wanted to use all eight bits in each GAT byte to extend the number of granules the GAT could account for. However, the old TRSDOS 2.3 standard was to use only the right two bits of each GAT byte; so we couldn't arbitrarily force all directories to start using all 8 bits. Yet, we wanted to allow use of all 8 bits; so we had to come up with a name for a byte in the GAT as distinct from anything else. Under the assumption that if a number of sectors is a granule, then a number of granules could be called a lump, we defined a lump to be simply a byte in the Granule Allocation Table in the data volume directory's first sector, and that's all it is.

11. ddsll means Default Directory Starting Lump and means the relative number of the lump whose 1st sector is the beginning of the data volume's directory. ddsll is an integer between 1 and 191, though no guarantee is given that a particular value will work. The standard ddsll value in the 35/40 track single sided, single density diskette world was and is 17, and your master NEWDOS/80 system diskette uses that value. If ddsll = 0 is specified, NEWDOS/80 will compute a ddsll value somewhere near the middle of the data volume, but not greater than 80, as it is assumed more data will exist near the beginning of the volume than at the end.

All Model I and Model III DOSs put the directory somewhere in the middle of the data volume. Since NEWDOS/80 runs with a variety of diskette and hard disk capacities, NEWDOS/80 allows the user to specify where the directory is to be put. The ddsll value is this specification. NEWDOS/80 stores the ddsll value in 3rd byte of the first sector of BOOT/SYS (also the first sector of the data volume) during data volume format (either FORMAT or COPY) so that DOS (and clever users) can find the directory. NEWDOS/80 senses it has lost the directory location when it reads a directory sector that is not protected. It then goes to the 3rd byte of the volume's 1st sector for the ddsll value and computes the directory location. The standard DDSL value for the 35 and 40 track single density diskettes was 17, but as diskettes have increased in capacity and hard disks have appeared, starting the directory at lump 17 placed it too close to the start of the data volume. For dual sided 80 track, double density diskettes with GPL=8, it was common to put the directory at lump 35.

In the diskette world, DDSL has meaning only when a diskette is formatted as NEWDOS/80, at all other times, can find the directory when it wants to. However, in the hard disk world, since we can't write directory sectors with address marks different from the other sectors, NEWDOS/80 cannot tell when it should go to the volume's first sector, get the ddsll value, and recompute the location of the directory. Therefore, the ddsll value is used by NEWDOS/80 at all times to know where a hard disk volume directory is. If you change the ddsll value at a time other then just before the hard disk volume is formatted, NEWDOS/80, without realizing it, will process non-directory data as directory data.

12. ddsal means Default Directory Sector Allocation and specifies the number of sectors to be used for the directory. ddsal is an integer from 10 to 33. This ddsal value is different than the DDGA value used by PDRIVE for floppy diskette definitions. Do not confuse the two. The change from DDGA to DDSA was necessitated by the fact that SPG for the hard disks is no longer a standard 5 sectors per granule. A ddsal value of 10, 15, 20, 25 and 30 is compatible with older configurations that used DDGA=2, 3, 4, 5 or 6 respectively. A ddsal value of 33 allows a data volume to have a maximum of 246 user files. Unless a hard disk data volume is to be small or compatibility with diskettes is to be maintained, it is recommended that ddsal = 33 be used.

The ddsal value for hard disk is more important than the DDGA value is for floppies. The DDGA value is used only at diskette format time. The ddsal value, along with the ddsll value, is the only way the DOS sector I/O routines know if a sector is part of the hard disk data volume directory or not; therefore, if the ddsal value is to be changed, it must be changed only before a hard disk data volume is formatted. The ddsll and ddsal values are the ONLY way the NEWDOS/80 sector I/O routines know that given hard disk sector is a directory sector.

EXAMPLES:

****** Remember, when parameter HDS is specified, all 12 subparameters must be supplied in the correct order.

1. PDRIVE,0,1,HDS=(0,306,0,2,32,0,0,2880,5,8,35,33)
specifies a 2880 sector data volume with 5 sectors per granule, 8 granules per lump, a 33 sector directory positioned at the start of lump 35. The first 2880 sectors of the first drive section of hard disk drive 0 will be allocated to this volume. The drive section consists of the first 2 recording surfaces of the drive, which may or may not be all that the drive has. Each recording surface has 306 tracks. Each track has 32 sectors and the drive's stepping rate code is 0. This data volume can be accessed by user programs as drive 1.

2. PDRIVE,0,2,HDS=(1,153,1,3,32,6,2000,10000,0,8,0,33)
specifies a data volume on hard disk drive 1 that has 153 tracks per surface and 32 sectors per track. The drive section consists of the 2nd, 3rd and 4th recording surfaces. The data volume consists of 10,000 sectors beginning with the drive section's relative sector 2,000. PDRIVE will compute the sectors per granule and and use 8 granules per lump. PDRIVE will compute the positioning of the 33 sector directory within the volume. User programs will access this data volume as drive 2.

3. PDRIVE,0,1,HDS=(0,153,0,4,32,6,0,*,0,8,0,33)
specifies a data volume that occupies all 19,584 sectors of the first four

recording surfaces of hard disk drive 0. The vscl, spgl and ddsll values are computed by PDRIVE. User programs will access this data volume as drive 1.

306 4896 8 61

4. PDRIVE,0,3,HDS=(0,153,0,1,32,6,0,0,5,2,17,33) specifies a null data volume (vscl value is 0). NOTE, all other subparameters must be valid. If a user program attempts I/O via drive 3, DEVICE NOT AVAILABLE error will occur. However, the FREE command and any other DOS functions that search the various drives will ignore drive 3.

EXAMPLES OF PDRIVE COMBINATIONS:

1. Settings to exactly overlay the standard LDOS volumes on a single Tandy 5 Meg drive where each of 4 volumes has one surface.

```
HDS=(0,153,0,1,32,6,0,4896,5,8,61,33)
HDS=(0,153,1,1,32,6,0,4896,5,8,61,33)
HDS=(0,153,2,1,32,6,0,4896,5,8,61,33)
HDS=(0,153,3,1,32,6,0,4896,5,8,61,33)
```

This divides the hard disk drive into 4 drive sections, each containing one data volume. If you assign the 4 definitions to PDRIVE slots 0 - 3 respectively, you must have moved the NEWDOS/80 system to hard disk as described in section 5. However, if you assign these definitions to slots 4 - 7 and have previous file data from LDOS operation, you can look at that data via SUPERZAP (if you are interested), and you can look at the directory starting at relative sector 2432.

2. The user has one Apparat 5 Meg drive, fundamentally wants all his/her user files accessible via drive 1 with a small amount of work space on drive 2. The user wants to run using a hard disk system volume for drive 0 and to be able to access to his two floppies via slots 4 and 5. With SYSTEM option AL = 6, the definitions for slots 0 - 5 will be as follows:

```
HDS=(0,306,0,2,32,6,0,720,5,8,17,10)
HDS=(0,306,0,2,32,6,720,16864,11,8,80,33)
HDS=(0,306,0,2,32,6,17584,2000,5,8,25,33)
HDS=(0,1,0,1,1,0,0,0,5,8,17,10)        a dummy definition
TI=A,TD=E,TC=40,SPT=18,TSR=0,GPL=2,DDSL=17,DDGA=2
TI=A,TD=E,TC=40,SPT=18,TSR=0,GPL=2,DDSL=17,DDGA=2
```

Note that the 8th subparameter (vscl) of HDS is the number of sectors assigned to the data volume (NOT the ending sector number). Slot 3 has been defined as a dummy (the vscl value = 0) to allow FREE to get to slots 4 and 5.

3. The user has two Apparat 10 Meg drives and wants the system volume on hard disk, 8 hard disk data volumes with slot 1 to contain all the space of the 2nd drive. The definitions for slots 0 - 7 could be:

```
HDS=(0,306,0,4,32,0,0,5595,5,8,69,33)
HDS=(1,306,0,4,32,0,0,39168,26,8,94,33)
HDS=(0,306,0,4,32,0,5595,5595,5,8,69,33)
HDS=(0,306,0,4,32,0,11190,5595,5,8,69,33)
HDS=(0,306,0,4,32,0,16785,5595,5,8,69,33)
```

```
HDS=(0,306,0,4,32,0,22380,5595,5,8,69,33)
HDS=(0,306,0,4,32,0,27975,5595,5,8,69,33)
HDS=(0,306,0,4,32,0,33570,5595,5,8,69,33)
```

4. FORMATTING YOUR HARD DISKS.

Hard disks must be formatted before they can be used with NEWDOS/80 Version 2.5 or any other DOS. Some hard disk manufacturers format their hard disks before shipping the drive and have internal coding to bypass error sectors automatically, and if this is the case, you may bypass this section on hard disk formatting.

NEWDOS/80 Version 2.5 does not maintain an error sector table and assumes the consecutive sectors that it can read from a hard disk are error free. Bad (error) sectors must be hidden from NEWDOS/80. One way to do this is to reduce the number of data sectors per track, allowing HDFMTAPP to write a dummy sector over the bad spot on the track. Another way is to later define (via PDRIVE) the data volumes such that the bad sectors are not part of any data volume.

NEWDOS/80 DOS commands FORMAT or COPY with format do not actually format a hard disk. The actual formatting must be done either by a stand alone program or by a program that operates under NEWDOS/80 but does all of its own I/O to the hard disk. NEWDOS/80 Version 2.5 provides the program HDFMTAPP to format Apparat's and Tandy's hard disks for the Model I or III. The format program for other types of hard disk drives must be supplied to the user by that hard disk drive retailer.

Formatting a hard disk destroys all information on that hard disk. If you must re-format a hard disk, be sure to extract as much valued information from that hard disk (you may use program HDBACKUP) as you can before re-formatting.

Though we recommend that you format the hard disk drive before use with NEWDOS/80 so that you will be made aware of all the error sectors, a previous format done for another DOS (such as done during the LDOS 5.1.3 hard disk initialization) can suffice if there were no error sectors or you know where they are for bypassing in your definition of data volumes using PDRIVE, and if you know the parameters needed for PDRIVE's HDS parameter. If you elect to do this, then bypass the rest of this section (on HDFMTAPP). An example where you might want to do this is where you wish to share the hard disk between one or more existing LDOS volumes and one or more NEWDOS/80 volumes, thus allowing both LDOS and NEWDOS/80 to use the hard disk (though not both at the same time and not the same data volumes).

To format an Apparat or Tandy Model I or III hard disk, assure the hard disk drive is properly connected to the computer and power is on; then execute the DOS command HDFMTAPP, proceeding as follows:

1. Reply the relative hard disk drive number. This is the same number as hddnl in the PDRIVE HDS parameter.

2. Reply the relative number of the first surface to be formatted. When

formatting an entire hard disk drive, this value is 0.

3. Reply the number of recording surfaces to be formatted. When formatting an entire hard disk drive, the value is the number of recording surfaces the hard disk has (the RSC or TPC values discussed earlier). For Apparat 5, 10 and 15 Meg hard disks, this value is 2, 4 and 6 respectively. For Tandy 5 Meg drives, this value is 4.

4. Reply the number of tracks per surface (TPS or tpsl) for this drive. This is the same as the number of cylinders the drive has. For Apparat hard disks, this value is 306. For Tandy 5 Meg hard disk drives, this value is 153.

5. Reply the relative number of the first cylinder (the first track on a surface) to be formatted. When formatting an entire hard disk drive, this value is 0.

6. Reply the number of cylinders (number of tracks on each surface) to be formatted. When formatting an entire hard disk drive drive, this value is the same as given in #4 above.

7. Reply the track stepping rate code. Use a value of 15 here as we are not too concerned with a slow stepping rate during formatting.

8. Reply your intended data sectors per track. The normal value here is 32. The tracks supposedly have a capacity for 33 sectors per track, but test have shown that many parity errors occur. Specifying 32 sectors per track does allow for one error sector per track to be automatically specially encoded so that NEWDOS/80 will never see it.

9. Reply the sector interleave count. We recommend a value of 21 if there are to be 32 sectors per track. This value allows time for the DOS I/O routine, the transfer of the bytes on the cable to/from the drive's buffer, the actual read/write of the sector by the drive, and 1 to 2 milliseconds for the user program to invoke the I/O for the next sequential sector. This value of 21 is also optimal for the HDBACKUP program, which is too slow as it is. Values 19 and 20 will work, but allow much less time for the user program to turn the I/O around. Values 22 to 30 allow the user more turn around time but slowly decrease the number of I/Os per second that can be done. Values 0 - 18 allow too little time for the above functions and cause the hard disk to wait till the next revolution (16.7 ms) for the next sector.

10. Reply N if you wish to restart the specifications again at step 1 above. Reply Y if the program is to start the format.

11. Once started, the formatting will proceed, blinking an asterisk in the display upper right corner to indicate progress. If a track cannot be formatted with the required number of sectors, an error will be displayed giving the cylinder, head and number of error sectors above and beyond the number implicitly allowed in step 8 above. A track that has some error sectors and some good data sectors will have the good sectors numbered from track relative sector 0 consecutively on up with the higher numbered sectors for that track simply not there.

12. During HDFMTAPP execution, holding down the up-arrow key causes the program to terminate and the right-arrow key causes the program to pause. After right-arrow, pressing ENTER causes the program to continue. This pause/cancel function is useable only through the keyboard matrix, not via remote terminals.

13. When the format is complete, the number of tracks with too many errors will be displayed. If there are any such tracks, you SHOULD reformat the hard disk using a lesser sectors per track value. Mark the resulting sectors per track value sptl on a label on the hard disk to remind you of what sptl value MUST be used in all PDRIVE definitions for data volumes on that drive. HOWEVER, when only a small number of consecutive tracks have all the error sectors, you may decide to leave the error sectors alone and define your volumes (via PDRIVE) in such a way as to assure that the error tracks at not assigned to any volume (i.e., ending one volume on the last sector of the first good track preceeding the bad track range and starting another volume on the first sector of the first good track following the bad track range). If the error tracks are assigned to a volume, NEWDOS/80 will give SECTOR NOT FOUND error whenever I/O is attempted to the a bad, non-existent sector. NEWDOS/80 does NOT maintain any bad track or bad sector tables.

14. If all tracks have been formatted with the required number of sectors, the hard disk is now ready for use by NEWDOS/80.

It is possible, due to the extensive specifications, for the HDFMTAPP program to format just one track on the hard disk. This may be of interest to a few users when a track has apparently gone bad and an attempt is to be made to reformat just that one track.

5. MOVING NEWDOS/80 VERSION 2.5 TO THE HARD DISK.

Usually, you want to have slots 0 to 3 as hard disk volumes and still have access to your two floppy disk drives. For this, it is necessary to operate using the NEWDOS/80 Version 2.5 system volume, which must be volume 0, from the hard disk. This section steps you through setting up NEWDOS/80 Version 2.5 to run from the hard disk. The hard disk is assumed previously formatted.

1. Be sure you know how to use the DOS command PDRIVE, especially with the Hard Disk Specification parameter HDS.

2. Mount a copy of the NEWDOS/80 Version 2.5 hard disk system diskette in floppy drive 0. This will be known as the system diskette as different from the hard disk system volume which will be on the hard disk.

3. Choose one of the system diskette's PDRIVE active slots whose number is greater than one. For this example slot 2 will be used (the SYSTEM option AL must be at least 3). If you choose a different slot number, then use that number in place of 2 in the following discussion.

4. Using PDRIVE,0,2,A,HDS=----- define floppy system diskette PDRIVE

slot 2 with the specifications wanted for the hard disk system volume.

5. Execute the DOS command:

```
COPY,0,2,,FMT,CBF,USD
```

and respond to the requests for SOURCE and DESTINATION diskettes (even though the destination is on a hard disk). GAT OVERFLOW error may occur if the spgl value for the destination is less that that of the source; in which case you must increase the destination spgl value.

6. Execute PDRIVE,2 to see the hard disk system volume's specifications for the 10 slots defined on that volume. Note that the definition for slot 2 has been duplicated in slot 0. This was done as a normal part of the COPY done above. Don't confuse the specifications of PDRIVE,2 which refers to system control data on drive 2, the intended hard disk system volume, with those of PDRIVE,0 which refers to system control data on drive 0, the floppy system diskette.

7. Using PDRIVE,2,----- define the PDRIVE specifications as you intend for that volume to be used as the system volume (drive 0). Since PDRIVE,2,2 has been duplicated as PDRIVE,2,0 in anticipation of that hard disk volume becoming the system volume, you MUST now redefine the PDRIVE,2,2 slot for another volume or by setting its vscl value to 0, causing slot 2 to be undefined. The specifications for PDRIVE,2 slots 0 - 3 must be for hard disks volumes only. Definitions for the floppies must be in slots 4 - 7 which correspond to the old drives 0 - 3 respectively. If one or more of the slots 4 - 7 are not used for floppies, then they may be used for hard disk volumes, thus allowing a maximum of 8 hard disk volumes to be active at any one time. Do not go on to the next step until all PDRIVE,2 slots have been defined as you will want them to be in the system operating from the hard disk, though it is not necessary to change any of them except slot 2 and you should not change slot 0. Remember, you cannot use PDRIVE parameter A when doing PDRIVE,2 definitions as that volume is not the current system volume.

8. Using SYSTEM,2,AL=xxx, specify the number of PDRIVE,2 slots to be active. xxx must be between 1 and 8, and must be at least 5 if any floppies are to be used.

9. The hard disk system volume now has the correct specifications, but we need a hard disk boot diskette (also known in this section as the boot diskette) to enable RESET (also known as BOOT), which must start on floppy drive 0, to switch to the hard disk system volume. This diskette must contain at least BOOT/SYS, DIR/SYS and SYS0/SYS, and must have its PDRIVE slot 0 defined exactly as for the hard disk system volume. So we proceed to do this.

10. If the system diskette's PDRIVE,0,1 specification is not identical to that for PDRIVE,0,0, then make them so by executing the command:

```
PDRIVE,0,1=0,A
```

11. Assign an otherwise unused diskette as the hard disk boot diskette and label it as such. Mount the boot diskette in floppy drive 1.

12. At this point, the system diskette is in floppy drive 0, the hard disk boot diskette in floppy drive 1, and the hard disk system volume is

on the hard disk. Execute the DOS command:

```
FORMAT,1,,,,Y
```

13. When done, execute to DOS command:

```
COPY,SYS0/SYS:2,:1
```

to move a copy of SYS0/SYS, the resident DOS, from the hard disk system volume to the hard disk boot diskette. Since it is the first file placed on the boot diskette, aside from BOOT/SYS and DIR/SYS, it will automatically be placed in the proper place for RESET.

14. When done, execute:

```
PDRIVE,1,0=2
```

to move the proper hard disk system volume specification to the boot diskette's PDRIVE slot 0.

15. At this point, you may want to change the PDRIVE,0,2 and PDRIVE,0,1 definitions back to what they were before steps 3 and 10 above. This step is optional.

16. Remove the system diskette from drive 0. Move the hard disk boot diskette from drive 1 to drive 0 and press RESET. Computer execution will read the boot sector and then the resident DOS, SYS0/SYS, from the boot diskette in floppy drive 0 and then shift to the hard disk. You may now take the hard disk boot diskette out of drive 0 or leave it in, in which case it may be accessed via the PDRIVE slot 4 (used for floppy drive 0 when the hard disk system is in use) if PDRIVE,0,4 is defined for a floppy. The diskette can be accessed by user programs as drive 4.

You may use the hard disk boot diskette as a normal data diskette by copying data files on to it. Remember though, it is the hard disk system's boot diskette and its SYS0/SYS is the resident DOS that is loaded into main memory at RESET time and remains there until the next RESET.

> ********* WARNING. A backup up of a hard disk boot diskette will not transfer its booting-up-the-hard-disk capability unless the backup is done using format 5 COPY with the BDU option.

The hard disk system volume is drive (slot) 0 when operating the system from the hard disk. The hard disk system volume does NOT have to be positioned at the beginning of a hard disk drive; in steps 4 and 5 above, you are allowed to place the hard disk system volume where you wish on the hard disk.

******** The file SYS0/SYS on the hard disk boot diskette MUST remain exactly identical to the SYS0/SYS on the hard disk system volume. If you alter one, you MUST alter the other. This is necessary because the hard disk system thinks its own SYS0/SYS is in the resident DOS area (4000H - 4CFFH and 0F900H - 0FFFFH) at all times when actually it is the SYS0/SYS from the hard disk boot diskette.

If you only have one floppy drive, then the following changes must be made to the above proceedure:

1. Step 10 above is excluded.

2. In step 11, do not mount the boot diskette into drive 1.

3. In step 12, change the command to be FORMAT,0,,,,Y and perform diskette mounts as requested where the SYSTEM diskette is the system diskette and the DESTINATION diskette is the boot diskette.

4. In step 13, change the command to be COPY,$SYS0/SYS:2,:0 . Perform the diskette mounts as requested where the SYSTEM diskette is the system diskette, SOURCE diskette is the hard disk system volume and the DESTINATION diskette is the boot diskette.

5. Replace step 14 with the following action. Enter SUPERZAP and at the menu, reply CDS. Remove the system diskette from floppy drive 0, and mount the boot diskette in floppy drive 0. Reply Y. Reply 2,2 for the source drive and relative sector. Reply 0,2 as the destination drive and relative sector. Reply 1 as the sector count. Press ENTER to return to menu. Remount the system diskette in floppy drive 0. Reply EXIT to exit SUPERZAP and return to DOS READY.

6. DEFINING PDRIVE SLOTS FROM A VOLUME DEFINITION FILE.

The definition of hard disk volumes via PDRIVE is more difficult and more critical than for floppy disk volumes. Therefore, it is recommended that the user carefully plan out his/her allocation of hard disk space amongst the various volumes and store the definitions (the HDS parameter part) into an ASCII text file (called a data volume definition file) created and updated by using either CHAINBLD or SCRIPSIT or both. Do this very, very, very carefully as you can create havouc amongst your data if two or more data volumes share the same hard disk sectors. Under NEWDOS/80 Version 2.5, you have great flexibility in assignment of hard disk space to data volumes, but with this flexibility comes complexity of definition.

Each record within the data volume definition file must start with a unique but arbitrarily assigned identification integer. Following the integer must be a comma followed by the intended PDRIVE definition excluding the initial part of the PDRIVE command (the PDRIVE,dn1,dn2, portion) and the ,A (for activation) as these parts of the PDRIVE command will be supplied by the EXTPDRIV/BAS program.

Since each definition record within the data volume definition file starts with an integer, you may imbed comments within the file as you like provided the comment record does not start with an integer.

It is strongly recommended that you keep copies of the data volume definition file on floppy diskettes in case that file on your hard disk becomes unusable. Remember, this is your master copy of the hard disk space layout!

Assuming that you have carefully constructed your data volume definition file, you may assign one or more of these definitions to the various PDRIVE slots when needed by running the BASIC program EXTPDRIV/BAS.

1. The program will ask for the filespec of your volume definition file and then open it.

2. The program will ask for the identification integer of the definition to be used. Respond with an EXACT copy of the integer that starts that definition's record in the file. The program will then search the file for the record.

3. When found, the program will ask for the two numbers needed for the PDRIVE,dn1,dn2,--- function. Respond with the two numbers separated by a comma. The first number, dn1, (usually 0) specifies which data volume contains the system control information which will be changed by the PDRIVE command. The second number, dn2, specifies which PDRIVE slot definition is to be changed.

4. The program will then ask if slot definitions are to be activated within the resident DOS (i.e., the ,A PDRIVE parameter). Reply Y if so; N if not.

5. The program will then build the appropriate PDRIVE command and execute the command via DOS-CALL. You will see the PDRIVE results displayed.

6. The program will then ask if there is another definition from the same

file to be applied. If you reply Y, the program returns to step 2 above. If you reply N, the program ends.

EXAMPLES of data volume definition file records:

1. 103,HDS=(0,153,0,4,32,6,0,2880,5,8,35,33)

2. 91,HDS=(1,153,0,4,32,6,1000,2000,0,8,0,33)

3. 44,TI=A,TD=E,TC=40,SPT=18,TSR=0,GPL=2,DDSL=17,DDGA=2

7. BACKING UP HARD DISKS TO DISKETTE:

Copies of user data stored on hard disk must be kept elsewhere in case the hard disk crashes, a program malfunctions or a user goofs. Users MUST, from time to time, make backup copies of valued data, the frequency of backup depending upon how often the data changes and how valuable the data is.

NEWDOS/80 Version 2.5 provides the HDBACKUP (hard disk back up) function as a way of saving files from the hard disk(s) to floppy diskettes, and a way of restoring one, some or all of those files back onto the hard disk(s).

HDBACKUP saves by file rather than by full volume contents. It uses this considerably slower technique because over 50% of the restores that are eventually done involve only a selected set of files and not a full media or data volume. Restores to a hard disk don't have to be the result of a hard disk failure but more frequently are due to user mistakes or user program malfunction logically damaging or destroying certain files, and the restore should allow only the damaged files and their interrelated files to be restored, leaving unchanged all other files on the hard disk(s) involved. Unfortunately, saving by file requires more administrative consideration than does saving by entire volume contents; so we hope the greater flexibility will be worth it.

For purposes of HDBACKUP discussion, a backup is the content of the one or more diskettes used to contain the files copied from data volumes during the execution of the HDBACKUP program's SAVE function. These diskettes must be preformatted and, after being used by SAVE, cannot be read/written using standard DOS functions; however, they can be read/written using SUPERZAP disk (not file) mode.

In this discussion of the HDBACKUP function, a data volume refers to one of the active hard disk data volumes defined via PDRIVE.

HDBACKUP/CMD is the program that (1) creates a backup containing specified files from the various defined data volumes (as defined by PDRIVE) of your system, (2) lists which files are contained within a backup and (3) restores specified files from a backup to the various defined data volumes of your system. HDBACKUP is the method under NEWDOS/80 Version 2.5 of backing up your files from hard disk or diskette and, if necessary, restoring one, some or all of those files back to the hard disk or diskette. Under the SAVE paramater, HDBACKUP creates a backup that spans one or more diskettes. Under the LIST parameter, HDBACKUP lists the filespecs of and errors associated with the files contained in the specified backup. Via the RESTORE parameter, HDBACKUP copies specified files from the backup to specified data volumes of your system.

The HDBACKUP SAVE function saves a file's contents, not its attributes. Except for the file name, name extension, data volume number and, if NND not specified, the logical record length, no other attributes of the file are saved such as passwords, protection level, etc. SYSTEM files are not SAVEd. The user is responsible for backing up system files to regular diskettes using the COPY command; normally it is sufficient to simply maintain copies of your original NEWDOS/80 Version 2.5 Hard Disk System diskette and your regular NEWDOS/80 Version 2 System diskette. If the NND parameter is specified, system files included in the INCLUDE list are copied, but are no longer marked as system files.

Provided the NND parameter is specified, the HDBACKUP function is designed to attempt to run with TRSDOS-like DOSs other than NEWDOS/80 Version 2.5. Via the NND parameter, you must inform the HDBACKUP/CMD program of certain values for that DOS.

The HDBACKUP program requires passwords be disabled, as standard file OPENs are done without passwords in the filespecs. If passwords cannot be disabled in the current system, the passwords must be taken off the files being backed up. SYSTEM option AA=N disables passwords in NEWDOS/80.

The HDBACKUP program requires, unless the NND parameter is specified, that all volume directories be named DIR/SYS.

Usually after the user has responded to a request, HDBACKUP displays an * to indicate that it is no longer waiting for an operator response.

HDBACKUP blinks an * in the upper right corner of the display screen to let you know that is is preceeding in an orderly fashion. The speed of the blinks will vary due to the different functions.

The RESTORE function of HDBACKUP takes a very long time to initialize (in one test of 3444 files, it took 30 minutes). This extra initialization (1) performs KILLs if RENEW specified, (2) creates all new files, (3) CLOSEs the files to store the new EOF and release any excess disk space on the data volume, (4) if NND not specified, writes the last sector of each file to allocate any needed disk space and (5) if NND not specified, updates the logical record length in the directory.

The HDBACKUP/CMD program expects the diskettes used for the backup to already be formatted. The program will write over the entire diskette; after SAVE, the diskette will not have a directory. The program will not tolerate a bad sector when writing to the backup diskettes. If a sector is bad, you have three options: (1) retry the write, (2) cancel the entire SAVE function, or (3) restart the SAVE function at the beginning of the current backup diskette. If you choose option 3, you will be asked for the current backup volume again; you should then (and not before) mount a different previously formatted diskette (remember to label it properly) and place the other diskette in your bad diskette collection.

The HDBACKUP command sequence is:

```
HDBACKUP
fcl
PRINT
NND=(filespecl,r/n,spgl,gpll,spvl)
BSN=listl
TITLE=titlel
DATE=datel
TIME=timel
SVL=list2
RVL=list3
SLOW
SKIP
RENEW
MAXERRS=ecl
TEST
INCLUDE
EXCLUDE
*END
```

HDBACKUP invokes the HDBACKUP/CMD program. HDBACKUP must be the only parameter on the first command line (the command line used by DOS to invoke the program). This program then displays the cursor and waits for the user to input subsequent command lines. Command parameters are processed until the *END parameter is encountered. There must not be extraneous characters within a command line. A command line may contain multiple parameters separated by commas, but a parameter must be fully contained within a command line. A commmand line is limited to 79 characters in NEWDOS/80 and 63 characters for most other DOSs.

The user will generally build the command lines and the file specifications for INCLUDE or EXCLUDE into a CHAIN (aka DO) file as it is strongly recommended that HDBACKUP commands be constructed very carefully. Though CHAINBLD will work, it is recommended you build your chain file via a word processor, storing the resulting file as an ASCII file.

> **** Warning, be sure that the chain file has no extraneous characters after the end-of-line character for the *END statement; otherwise subsequent responses needed for the HDBACKUP execution will receive bad data.

The TEST parameter was included to allow the user a 'dry' run to test the workability of the command parameters. If you don't know what your are doing, gain some familiarly by using the TEST parameter before doing a live run. Remember, you can't test a RESTORE until you have a backup to test with.

> *** Warning, SAVE with TEST does write backup control information on the backup's 1st diskette; be sure that diskette is intended for a backup.

fcl fcl must be the first parameter after HDBACKUP. fcl specifies the function to be performed which is one of the following:

1. SAVE A new backup is created having title, date and time as specified by the TITLE, DATE and TIME parameters. The files specified, either explicitly or implicitly, are copied from the specified data volumes to as many backup diskettes as necessary. Parameters BSN, SVL and *END are required. Optional paramters are PRINT, NND, TITLE, DATE, TIME, SLOW, SKIP, MAXERRS, TEST, INCLUDE and EXCLUDE. If one of TITLE, DATE or TIME is not specified, the HDBACKUP program will ask for that parameter. If NND is specified, INCLUDE must be specified.

2. LIST This function lists the files contained within the specified backup and includes their associated error sector numbers. Required parameters are BSN and *END. Optional paramters allowed are PRINT, NND, TITLE, DATE and TIME. The listing starts with the backup's name, date, time, file count and error count. Then for each file in the backup's table of contents, the following are listed:

1. The filespec for the file.

2. If the file has been deleted from the backup table of contents, '***** DELETED *****' is displayed and steps 3 - 6 are bypassed.

3. The file's EOF value in xxx/yyy format where xxx is the relative sector within the file and yyy the relative byte within the sector.

4. The file's logical record length, 1 - 256.

   If NND specified during the SAVE that made this backup, the record length may or may not be correct if the file's record length prior to the SAVE was not 256. This occurs under NND as HDBACKUP does not get the record length from the directory but records whatever record length appears in the FCB after OPEN. Normal NEWDOS/80 operations do not use the file's record length from the directory, but many users want it correct anyway. If a file's logical record length was changed during the SAVE and RESTORE, the user may correct it by using the LRL parameter of ATTRIB (see regular NEWDOS/80 Version 2 ZAP 007 (Model I) or ZAP 004 (Model III).

5. The location within the backup of the file's header sector, expressed as a backup volume number and a relative sector within that volume. This is of interest only to those viewing/updating the backup via SUPERZAP. Volumes (diskettes) of a backup are numbered consecutively from 1, not 0.

6. If the file has any error sectors, they are listed each in the decimal format:

   sssss/ee/vvv/rrrrr

   where:

   1. sssss is the sector's relative number within the file.

   2. ee is the DOS error code.

3. vvv is the number of the backup volume containing the error sector

4. rrrrr is the sector's relative number within the backup volume.

3. RESTORE The files specified, either implicity or explicitly, are copied from the backup to the specified data volumes. Required parameters are BSN, RVL and *END. Optional parameters allowed are PRINT, NND, TITLE, DATE, TIME, SLOW, SKIP, TEST, RENEW, INCLUDE and EXCLUDE.

PRINT This parameter informs the HDBACKUP program that display information is to be sent to the printer as well as the display. If PRINT is not specified, only the display will be used. If PRINT is specified, the program will display WAITING ON PRINTER, and then, if the printer is not ready, the program will hang.

NND=(filespecl,r/n,spgl,gpll,spvl) This option specifies that the Disk Operating System (the DOS) is not NEWDOS/80 Version 2.5, though it can be. If NND is specified, the following hold:

1. SLOW is implied.

2. For SAVE, INCLUDE is required.

3. NND must be specified immediately after fcl and before BSN.

4. For RESTORE, the pre-allocation of needed file space during initialization is not done; an out-of-space error will not be detected until the file is actually restored.

5. file logical record lengths recorded in table of contents during SAVE or in the data volume directory during RESTORE may be wrong if they were not 256.

HDBACKUP is designed to run with NEWDOS/80 Version 2.5, but users initially may have their hard disk data under a different operating system, thus creating a dilemma, as NEWDOS/80 cannot process directories for other DOSs. Recognizing this as potentially a serious problem, an attempt (via the NND parameter) has been made to allow HDBACKUP to run under another DOS using faked extents in the FCB used for backup diskette I/O. This attempt will not work with a DOS that determines a diskette's characteristics from the diskette itself (as HDBACKUP writes over the entire backup diskette) or which automatically changes a drive's specification when an error is encountered. So far, the only successful tests have been (1) with Tandy's Model III Hard Disk Operating System (LDOS 5.1.3) using single sided, double density, 40 track drives as the backup drives specified in the BSN parameter with NND=(TEMPFILE:0,N,6,3,720), and (2) with Tandy's Model I Hard Disk Operating System (LDOS 5.1.3) using single sided, single density, 35 track drives as the backup drives specified in the BSN parameter with NND=(TEMPFILE:0,N,5,2,350). Apparat does not plan to test under the other DOSs or other configurations, and Apparat reserves the right to withdraw the NND parameter and all support for it at any time and without notice.

If using HDBACKUP with the NND parameter does not work with your other DOS, the user will have to find some other way of offloading the files from hard disk under the other DOS and reloading them under NEWDOS/80 Version 2.5.

****** Warning!!! Before using HDBACKUP to offload files under a DOS that is not NEWDOS/80 and then reloading the files to hard disk under NEWDOS/80, the user should offload the valued files to diskettes using the other DOS's normal backup proceedures. This provides the user with a second backup source should the conversion to NEWDOS/80 fail.

When using the NND option, certain extra information MUST be provided to the HDBACKUP program. If you don't know what these values are, call the distributor for that DOS; don't call Apparat.

> filespecl is the filespec of new or existing file that HDBACKUP can write one sector to in order to determine a correct FCB to be used for backup diskette I/O. HDBACKUP will write garbage into that one sector and will not CLOSE the file. The file filespecl must be for a file within a volume that is already mounted when HDBACKUP begins execution; further, for some DOSs, it may be necessary that the file be on a diskette with the same spgl, gpll and spvl characteristics specified in this NND parameter. The diskette can be mounted on a drive specified in BSN below as HDBACKUP will conclude its use of file filespecl before it asks for the first backup diskette.
>
> r/n is one character, either R or N. R is specified if the EOF field of FCBs (the File Control Block in main memory, not the directory FDEs) for this DOS are in Relative Byte Address format (such as all NEWDOS versions and Model III TRSDOS 1.3). N is specified if the EOF field of the FCBs for this DOS are in Next Record Address format (such as LDOS (regular and hard disk), Model I TRSDOS 2.3 and Model III TRSDOS 1.1)
>
>> ********* The choice of R or N is critical. Choosing the wrong value will cause every file not ending on a sector boundary to be assigned the wrong EOF in the backup, thus making the file one sector too long or too short. Further, reportable errors may occur.
>>
>> Once again, the NEWDOS author apologizes for having brought Relative Byte Addressing to the TRS-80 world (the FCBs, not the directories) with the NEWDOS release in March, 1979, thus causing the confusion between RBAs and NRAs (Next Record Addressing). NRA was the standard at that time and has remained the LDOS standard (TRSDOS on the Model III changed to RBAs in July, 1982). NEWDOS shifted to and remains with RBAs because that method is the more reliable method for arbitrary random disk I/O.
>
> spgl is the number of sectors per granule for this DOS for the backup diskettes that will be mounted on floppy drive(s) specified in BSN below. (LDOS Hard Disk System uses spgl = 5 for single density 5 inch diskettes and spgl = 6 for double density).
>
> gpll is the number of granules per lump for this DOS for the backup diskettes that will be mounted on the floppy drive(s) specified in BSN below. This is also known as granules per cylinder and is the number of bits per byte used in the GAT sector to account for granule allocation. LDOS Hard Disk System uses gpll = 2 for single sided single density 5 inch

diskettes, gpl1 = 4 for double sided single density, gpl1 = 3 for single sided double density diskettes and 6 for double sided double density.

spvl is the number of sectors per backup diskette. This is the total number of sectors on a diskette (720 for single sided, double density 40 track 5 inch diskettes, 350 for single sided, single density 35 track 5 inch diskettes, 1440 for double sided, double density 40 track 5 inch diskettes). Whatever the number, the DOS must be capable of doing I/O for that number of sectors per diskette.

HDBACKUP/CMD may be moved to another DOS via the following steps:

1. Under NEWDOS/80 Version 2.5, execute LMOFFSET. Respond D. Respond HDBACKUP/CMD:0. Respond new load address = 7000. Respond N to request appendage. Record the new start, end and entry address values displayed (will be used in step 4 below). Respond <ENTER> to indicate load point not being changed again. Respond N to keep DOS enabled. Respond D. Respond XXX/CMD:0 to write the modified module back to disk. Respond N. Respond N again. You should now be back at DOS READY.

2. Execute the DOS command LOAD,XXX/CMD:0. This loads the load-offsetted HDBACKUP program created in step 1 into main memory from where it will be written to the other DOS's diskette in step 4 below.

3. Load the other DOS diskette into drive 0 and press RESET to bring up that DOS. Be sure that this DOS does not clear user memory upon coming up.

4. Use the DUMP command for that DOS to store onto that DOS's disk the HDBACKUP/CMD program loaded into main memory in step 2. The DUMP command will need the start, end and entry addresses recorded in step 1. See that DOS's manual for explanation of the DOS command DUMP. For LDOS, this command will be:

```
DUMP HDBACKUP/CMD:0 (START=X'start',END=X'end',TRA=X'entry')
```

where start, end and entry are the hexidecimal addresses recorded in step 1 above.

5. If that DOS's DUMP does not allow the filespec HDBACKUP/CMD:0, use what it will allow and then change the file's name via RENAME.

6. The HDBACKUP program is now ready for execution on that DOS.

BSN=list1 The Backup Slot Number specifies either one or two slot numbers (if two, list1 must be enclosed in parenthesis) of the slots (PDRIVE active volumes) to be used for reading/writing the backup diskettes. These slots must be defined in PDRIVE as floppy disk drives. None of the backup slot numbers may be included in the volume numbers listed in the SVL or RVL parameters. If two slot numbers are specified, they must have the same PDRIVE definition. If only one slot is specified, all backup diskettes will be mounted as needed using that one drive. If two slot numbers are specified, the backup's volume 1 is left mounted on the first drive throughout the HDBACKUP function and the second drive is used for the other volumes. Since backup volume 1 is frequently referred to or updated during the SAVE or RESTORE, assigning two slots (drives) greatly reduces operator actions. If you only have two drives, run the system from the hard disk so that floppy drive 0 is free to be used as a backup drive.

TITLE=title1 title1 is the 0 to 48 printable character title of the backup. For SAVE, this title is assigned to the backup; if not specified in the command lines, the program will ask for it. For LIST and RESTORE, an error will be displayed if TITLE is not specified or title1 does not match that of the backup; the user may elect to use the backup anyway. Where TITLE is specified in a command line, it must be the last parameter of that line as title1, even if over 48 characters, is considered to be the rest of the line; the excess characters are ignored. During SAVE, when a backup diskette is first asked for, the program will reject the diskette if it has been used for a previous backup with the same title, date and time (as it may really be an earlier volume of this backup).

DATE=date1 date1 is the backup's date in mm/dd/yy format. For SAVE, this date is assigned to the backup; if DATE is not specified, the operator will be asked for it. For LIST and RESTORE, an error will be displayed if DATE is not specified or date1 does not match the backup's date, but the user may elect to use the backup anyway.

TIME=time1 time1 is the backup's time in hh:mm:ss format. For SAVE, this time is assigned to the backup; if TIME is not specified, the operator will be asked for it. For LIST and RESTORE, an error will be displayed if TIME is not specified or time1 does not match the backup's time, but the user may elect to use the backup anyway.

SVL=list2 This Save Volume List parameter is required for and used only if the function is SAVE. list2 specifies the volume(s) whose files are to be copied to the backup during SAVE. If list2 has more than one subparameter, list2 must be enclosed in parenthesis. list2 consists of one or more subparameters, separated by commas, of the type:

> vn1 specifies the number of an active slot whose data volume files, as restricted by INCLUDE or EXCLUDE, are to be copied to the backup. vn1 may have integer values 0 to xxx, where xxx is one less than the SYSTEM AL parameter. vn1 must not equal a slot number specified in the BSN parameter.

RVL=list3 This Restore Volume List parameter is required for and used only if the function is RESTORE. This parameter specifies (1) volume numbers whose files in the backup, as restricted by INCLUDE or EXCLUDE, are to be restored and (2) optionally, the data volume to receive the files of another volume.

If list3 has more than one subparameter, **list3 must** be enclosed in parenthesis. list3 consists of one or more subparameters, separated by commas, of the type:

> vn2=vn1 The files contained in the backup for volume vn1, as restricted by INCLUDE or EXCLUDE, are copied to volume vn2. Volume numbers in the INCLUDE or EXCLUDE list refer to vn1, not vn2. If vn2 and vn1 are the same volume number, the vn2=vn1 subparameter may be written as just vn1. vn2 may have integer values 0 to xxx where xxx is one less than the SYSTEM AL parameter. vn2 must not equal a slot number specified in the BSN parameter.

SLOW This option can only be used with NEWDOS/80 and specifies that the HDBACKUP program is NOT to use its faster diskette I/O when reading/writing the backup (not the data volumes) diskettes. SLOW is implied by NND. Normally, NEWDOS/80 Version 2.5 uses a faster mode of backup diskette I/O in the hope of increasing the speed of SAVE and RESTORE by 20-40%. SLOW should be specified only if the fast I/O appears to actually run slower than normal diskette I/O. You can study this by timing the time to read or write a backup diskette, preferably a volume other than backup volume 1.

SKIP During HDBACKUP processing when an error is encountered and the operator would normally have a 'SKIP' option allowing processing to continue, if the SKIP command parameter was specified, the SKIP option will automatically be assumed. Normally, this option will not be specified; however, there are times when a SAVE or RESTORE must accomplish what it can despite errors. For example, if part of a hard disk has gone bad and the disk is to be sent to the repair shop where it may or may not retain its current data, it may be important to assure that whatever data can be retrieved, is retrieved with the problem of restructuring bad files addressed later.

RENEW This option is used only with RESTORE. During HDBACKUP initialization after the files to be restored have been determined, a KILL is issued to the destination volume for each file that is to be restored. If the file did not exist on the destination volume, the KILL does nothing. Normal RESTORE initialization will then recreate the files on the destination volumes. The purpose of RENEW is to reallocate file space in, hopefully, less fragmented units (which can increase the efficiency of programs using these files); RENEW should only be used when all, or almost all, files of a data volume are being restored.

MAXERRS=ec1 ec1 is the number of errors the backup is to provide for in its error table. The default value is 640 with 6400 the maximum ec1 value allowed. MAXERRS is used only in the SAVE function.

TEST This option allows initialization processing to occur, including backup control information writes. When the initialization is done, HDBACKUP terminates with 'TEST COMPLETED' error. TEST allows the user to test the command parameters, including the INCLUDE or EXCLUDE lists.

> *** Warning, TEST with SAVE writes control information to the backup's first diskette; this is necessary for a good test.

INCLUDE and EXCLUDE INCLUDE and EXCLUDE are mutually exclusive keywords. Each must terminate the current command input line. Subsequent command lines until but not including the *END command line compose a file list with each line specifying either a volume number preceeded by a colon (i.e., :3) or the filespec, without passwords, of a file to be INCLUDEd or EXCLUDEd. The number of volume numbers or filespecs allowed in a file list is limited by computer main memory constraints but is over 1500.

If the command line consist solely of a volume number, then all files for that volume are INCLUDED or EXCLUDED.

All volume numbers in the INCLUDE or EXCLUDE list must refer to a vnl volume number specified in the appropriate SVL or RVL parameter.

INCLUDE and EXCLUDE are optional keywords (except for SAVE with NND). If neither is specified, HDBACKUP will assume inclusion of all the files for the vnl volumes specified in the SVL or RVL parameter.

INCLUDE Only the files specified in the file list are included in the SAVE or RESTORE. If a file in the list does not exist, an error comment will be listed, and the operator given the option of bypassing the file.

EXCLUDE The files specified in the file list are excluded from the SAVE or RESTORE; all other files of the vnl volumes specified in the SVL or RVL parameter are copied. If a file in the list does not exist on the specified data volume (SAVE) or the backup (RESTORE), no indication is given to the operator.

*END This required parameter ends the HDBACKUP command specification.

INTERNAL STRUCTURE OF THE BACKUP:

For those users interested, this section will show the structure of a backup. Some users may find this description helpful in repairing a backup using SUPERZAP.

Each volume (diskette) of a backup has a volume header sector as the diskette's first sector. The header sector for volume 1 is the most important and is used by RESTORE and LIST to access backup control information. The headers for the other volumes contain roughly the same information, and are used during RESTORE to verify that you have mounted the correct volume and by SAVE to verify that you don't mount as a new volume for this SAVE a volume that has already been used in the SAVE. The user must remember that a file's sectors can span many backup volumes and must allow for the volume header records when computing where a particular sector of a particular file is within the backup. The contents of the backup volume header sector are:

1. 48 byte backup title.
2. 8 byte backup data in mm/dd/yy format.
3. 8 byte backup time in hh:mm:ss format.
4. 2 byte count of sectors for the table of contents.
5. 1 byte count of sectors for the error table.
6. 2 byte count of files in the table of contents.
7. 2 byte count of number errors allowed during SAVE.
8. 2 byte count of sectors per backup volume.

9. 2 byte value = this diskette's volume number.
******** valid only for volume 1:
    10. 1 byte of control bits:
        bit 7 = 1, the SAVE is complete.
        bits 6 - 0, undefined and reserved, must be 0.
    11. 2 byte count of errors in error table.
    12. 2 byte count of volumes for this backup.
    13. 3 byte backup total sector count.
14. remainder of sector's bytes are 00H.

2. On backup volume 1, the table of contents sectors immediately follow the volume header sector. Each sector contains eight 32 byte file entries of the form:

1. 8 byte file name, padded on right with blanks.
2. 3 byte file name extension, padded on right with blanks.
3. 1 byte data volume number.
4. 3 byte file EOF in RBA format.
5. 1 byte logical record length (0=256). Not necessarily valid if NND specified during SAVE.
6. 3 byte relative sector number within the backup of the file's header sector.
7. 2 byte relative entry number of this entry within the table of contents.
8. 1 byte control bits:
    bit 7 = 1, this table of contents entry is used.
    bit 6 = 1, this file is active. Bit 7-6 = 10, file has been deleted from the backup. Actually some of it may still be there, but LIST and RESTORE ignore it.
    bits 5 - 0, undefined and reserved, must be 0.

9. The remainder of the 32 byte entry are bytes 00H.

The error table sectors immediate follow the table of contents. Each sector has 64 entries of the form:

1. 1 byte containing the DOS error code plus 40H. If the byte is 00H, the error has either been corrected by the user or he/she wants it ignored.

2. 3 byte relative sector number within the backup of the file sector in error. If the error is corrected or to be ignored, this value must be set to 0.

The remainder of the backup is file data with each file's sectors preceeded by a file header record. If a file's EOF is zero, then only the file's header record will appear. The user must remember that where a file's sectors flow onto the next backup volume, the first sector on that volume with be the volume's header sector, not a file sector. The format of a file header is:

1. The first 22 bytes are an exact copy of the first 22 bytes of the table of contents entry for this file, but with none of the changes to that entry after it was initially created. During RESTORE, these 22 bytes of the file header must match the 22 bytes from the table of contents.

2. Each of the remaining bytes of the file header sector contains the ones complement of its relative location in the sector. This makes it easier to recognize a file header should it be necessary to search for it.

HDBACKUP EXAMPLES:

1.
```
    HDBACKUP
    SAVE,BSN=(4,5),SVL=(0,1,2,3),*END
```
is a copy of user files from hard disk data volumes 0, 1, 2 and 3 to a backup whose diskettes will be mounted on the floppy drives associated with slots 4 and 5 (assumed defined for floppy drives 0 and 1 respectively), with backup volume 1 remaining on slot 4's drive and the other backup volumes requested on slot 5's drive as needed. The user must have on hand enough pre-formatted diskettes for the needs of the backup. Since slots 4 and 5 are the access to floppy drives 0 and 1, we know that the hard disk system is being run from the hard disk.

2.
```
    HDBACKUP
    RESTORE,BSN=(4,5),RVL=(0,1,2,6=3),*END
```
This is a copy of user files from a backup to data volumes 0, 1, 2 and 6. The backup diskette volumes will be mounted on the drives for slots 4 and 5 as described in the above example. All files in the backup are copied, but the files that originally came from volume 3 are actually written to volume 6 instead.

3.
```
    HDBACKUP
    SAVE,BSN=1,SVL=(2,3,4,5,6,7)
    EXCLUDE
    XXX/DAT:4
    YYY/DAT:6
    *END
```
This is a copy of all user files from hard disk data volumes 2, 3, 4, 5, 6 and 7 to backup diskettes which will all be mounted as needed on the floppy drive 1. File XXX/DAT of volume 4 and file YYY/DAT of volume 6 will not be copied to the backup. Since BSN=1 was used for the backup floppy drive, we know the system is being run from a system diskette in floppy drive 0.

4.
```
    HDBACKUP
    LIST,BSN=1,PRINT,*END
```
The contents of the backup's table of contents is listed on both the display and the printer.

5.
```
    HDBACKUP
    SAVE,BSN=(4,5),SVL=(1,2),INCLUDE
    ACCTPYBL/DAT:1
    ACCTRVBL/DAT:1
    PAYROLL/DAT:2
    INVENTRY/DAT:2
    *END
```
A backup is made consisting only of the 4 files specified in the INCLUDE list. In this installation, the burden of making backups of valued files has been

placed on the individual users, in this case, accounting.

6.
```
     HDBACKUP
     SAVE,NND=(TEMPFILE:0,N,6,3,720),BSN=(4,5)
     SVL=(0,1,2,3),INCLUDE
     FILE001:0
     FILE002:0
     FILE003:1
     and so on through
     FILE999:4
     *END
```
In this example, the HDBACKUP/CMD program has been previously moved to the LDOS Hard Disk Operating System (in the manner described at the NND discussion). The HDBACKUP runs under LDOS 5.1.3 and dumps the specified files from volumes 0, 1, 2 and 3 to a backup whose diskettes have all been preformatted as single sided, double density, 5 inch 40 track (with spg = 6, gpl = 3 and spv = 720). After the hard disk has been reinitialized for NEWDOS/80 Version 2.5, the HDBACKUP program under NEWDOS/80 (without the NND parameter) can be used to RESTORE the files from the backup to the hard disk.

When NND is specified for a SAVE, such as above, an INCLUDE list must be used to inform the HDBACKUP program of which files to copy to the backup, as the HDBACKUP program does not read the non-NEWDOS/80 directories.

This example could be used for single sided, single density 35 track backup diskettes under LDOS 5.1.3 on the TRS-80 Model I by using replacing the NND parameter with NND=(TEMPFILE:0,N,5,2,350). If 40 track diskettes are used, replace 350 with 400.

The TEMPFILE:0 filespec used in this example is just our choice of a filespec for this example; you are free to use any filespec you wish so long as it conforms to the specifications given for the NND parameter.