Ira Goldklang's TRS-80 Revived Site

TRS-80 DOS - NEWDOS/80

by Ira @ 7:06 pm on March 25, 2009.

NEWDOS/80 DOS

Page Menu

Tips and Tricks

Background

NEWDOS/80 v1.0

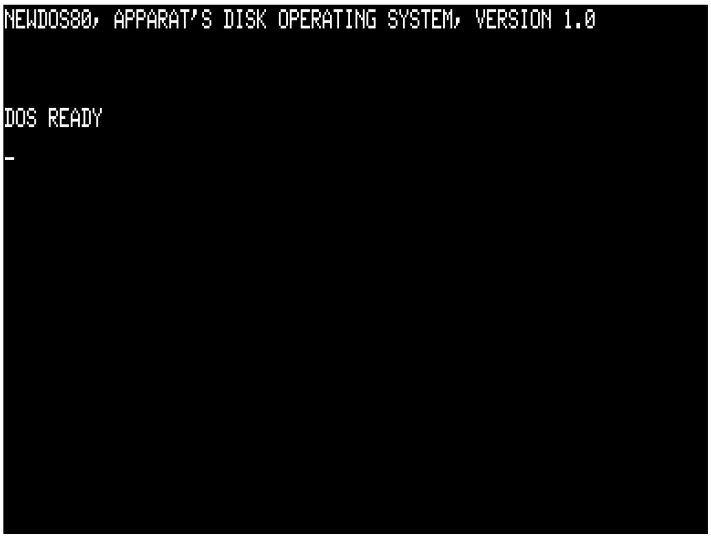
NEWDOS/80 is a Disk Operating System for the Model I and Model III TRS-80 made by Apparat, Inc.

Initially released for $149 around May 1980 as the successor to Newdos 2.1, it was another alternative to the notoriously fickle (and unreliable) TRSDOS which was shipped by Radio Shack, NEWDOS/80 offered a more versatile and user friendly experience. For those itching for the good old days of LEVEL I, NEWDOS/80 v1.0 included, as part of its software suite, a LEVEL I emulator!

Two of the most powerful NEWDOS/80 commands were SYSTEM and PDRIVE, the former allowing significant DOS customization options and the latter allowing significant disk format and disk interface options.

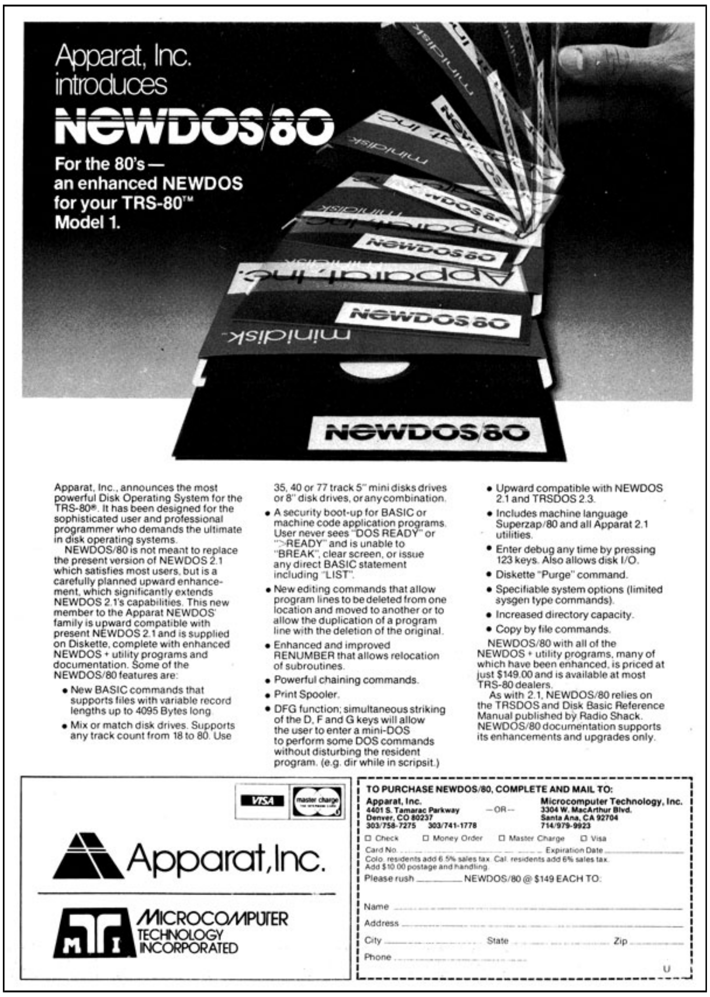
One of the best features of NEWDOS/80 was MINI-DOS. By simultaneously pressing DFG on the keyboard, you could enter a smaller version of DOS to check directories, free space, and other functions. You could do this from BASIC and any program which did not disable interrupts or take control away from the keyboard device control block (as DOS would not see the keypresses). This feature was phenomenal in preventing one from having to save all their work, exit BASIC for DOS, run their command, go back into BASIC, and try to remember where they were. The manual explains MINI-DOS best - “When, during the execution of a main program, the operator would like to interrupt the main program, execute one or more of the DOS library commands, and then resume main program execution without any changes having occurred in the main program’s state during the interruption. To execute MINI-DOS, simultaneously press the ‘DFG’ keys (but not durlng disk I/O), and execute the DOS command. Anv DOS library command can be used except APPEND, CHAIN, COPY, FORMAT, PDRIVE, AND SYSTEM. Single file copy can, however, be executed with the MDCOPY command.”

BASIC was modified as well to support renumbering (which previously required third party software) and variable and line number cross referencing using the REF command.



```
NEWDOS80, APPARAT'S DISK OPERATING SYSTEM, VERSION 1.0


DOS READY
_
```

(Bootup Screen for NEWDOS/80 v1.0)


Apparat, Inc.
introduces
NEWDOS/80
For the 80's —
an enhanced NEWDOS
for your TRS-80™
Model 1.
minidisk
Apparat, Inc.
NEWDOS/80
VISA
master charge
Apparat, Inc.
MICROCOMPUTER TECHNOLOGY INCORPORATED

Apparat, Inc., announces the most powerful Disk Operating System for the TRS-80®. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

NEWDOS/80 is not meant to replace the present version of NEWDOS 2.1 which satisfies most users, but is a carefully planned upward enhancement, which significantly extends NEWDOS 2.1's capabilities. This new member to the Apparat NEWDOS' family is upward compatible with present NEWDOS 2.1 and is supplied on Diskette, complete with enhanced NEWDOS + utility programs and documentation. Some of the NEWDOS/80 features are:

- New BASIC commands that supports files with variable record lengths up to 4095 Bytes long.
- Mix or match disk drives. Supports any track count from 18 to 80. Use 35, 40 or 77 track 5" mini disks drives or 8" disk drives, or any combination.
- A security boot-up for BASIC or machine code application programs. User never sees "DOS READY" or ">READY" and is unable to "BREAK", clear screen, or issue any direct BASIC statement including "LIST".
- New editing commands that allow program lines to be deleted from one location and moved to another or to allow the duplication of a program line with the deletion of the original.
- Enhanced and improved RENUMBER that allows relocation of subroutines.
- Powerful chaining commands.
- Print Spooler.
- DFG function; simultaneous striking of the D, F and G keys will allow the user to enter a mini-DOS to perform some DOS commands without disturbing the resident program. (e.g. dir while in scripsit.)
- Upward compatible with NEWDOS 2.1 and TRSDOS 2.3.
- Includes machine language Superzap/80 and all Apparat 2.1 utilities.
- Enter debug any time by pressing 123 keys. Also allows disk I/O.
- Diskette "Purge" command.
- Specifiable system options (limited sysgen type commands).
- Increased directory capacity.
- Copy by file commands.

NEWDOS/80 with all of the NEWDOS + utility programs, many of which have been enhanced, is priced at just $149.00 and is available at most TRS-80 dealers.

As with 2.1, NEWDOS/80 relies on the TRSDOS and Disk Basic Reference Manual published by Radio Shack. NEWDOS/80 documentation supports its enhancements and upgrades only.

TO PURCHASE NEWDOS/80, COMPLETE AND MAIL TO:

Apparat, Inc.
4401 S. Tamarac Parkway
Denver, CO 80237
303/758-7275 303/741-1778

—OR—

Microcomputer Technology, Inc.
3304 W. MacArthur Blvd.
Santa Ana, CA 92704
714/979-9923

☐ Check ☐ Money Order ☐ Master Charge ☐ Visa

Card No. __________ Expiration Date __________

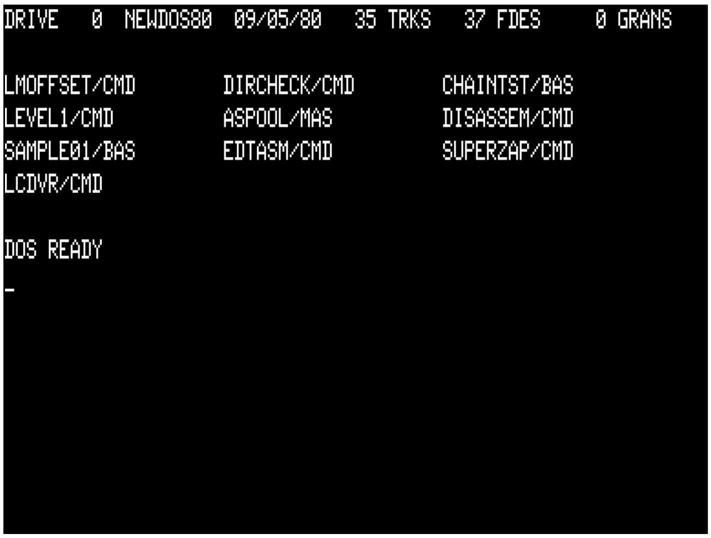
Colo. residents add 6.5% sales tax. Cal. residents add 6% sales tax.
Add $10.00 postage and handling.

Please rush ______ NEWDOS/80 @ $149 EACH TO:

Name __________
Address __________
City __________ State __________ Zip __________
Phone __________

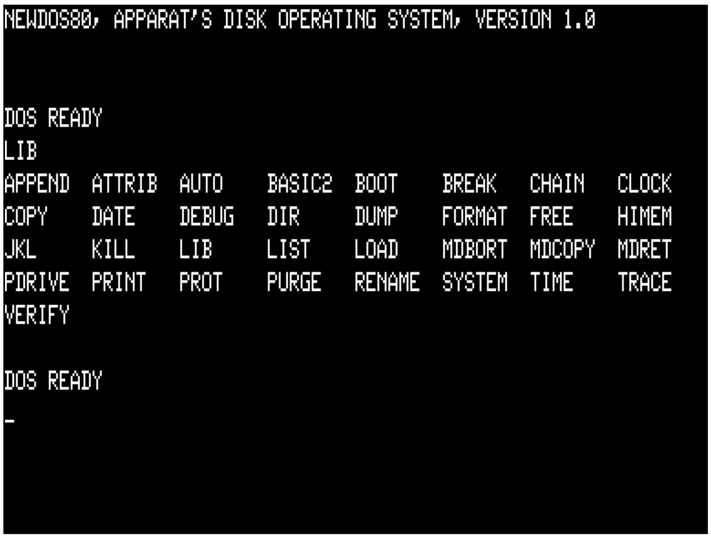
U

NEWDOS/80 v1.0 Advertisement from the May 1980 issue of 80 Micro



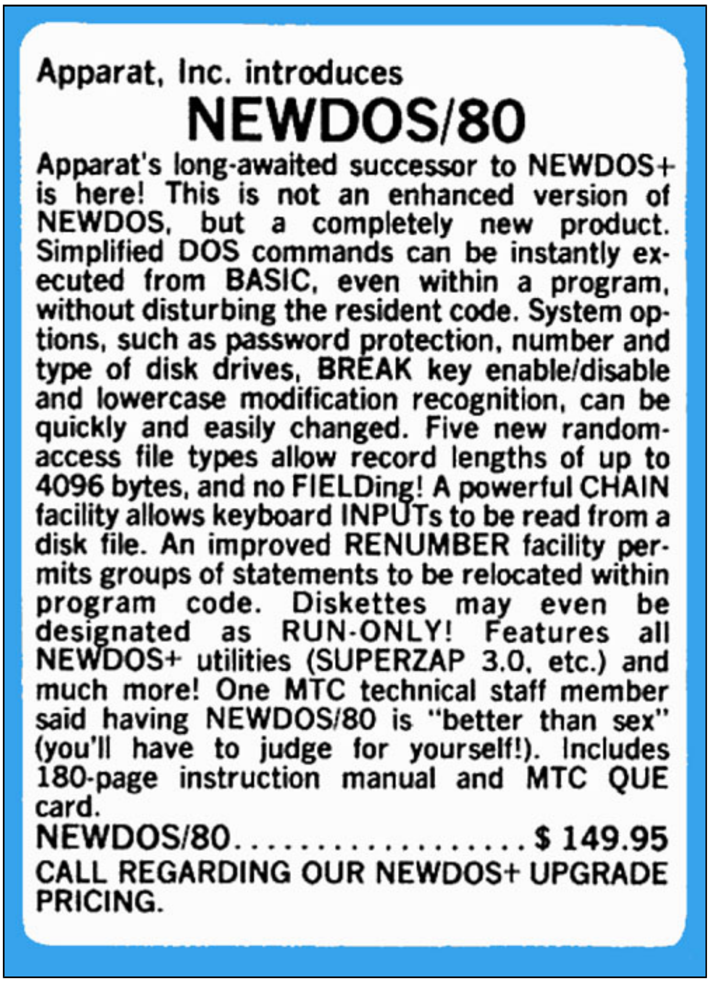
```
DRIVE  0  NEWDOS80  09/05/80   35 TRKS   37 FDES     0 GRANS

LMOFFSET/CMD       DIRCHECK/CMD       CHAINTST/BAS
LEVEL1/CMD         ASPOOL/MAS         DISASSEM/CMD
SAMPLE01/BAS       EDTASM/CMD         SUPERZAP/CMD
LCDVR/CMD

DOS READY
_
```

(Directory Screen for NEWDOS/80 v1.0)



```
NEWDOS80, APPARAT'S DISK OPERATING SYSTEM, VERSION 1.0


DOS READY
LIB
APPEND  ATTRIB  AUTO    BASIC2  BOOT    BREAK   CHAIN   CLOCK
COPY    DATE    DEBUG   DIR     DUMP    FORMAT  FREE    HIMEM
JKL     KILL    LIB     LIST    LOAD    MDBORT  MDCOPY  MDRET
PDRIVE  PRINT   PROT    PURGE   RENAME  SYSTEM  TIME    TRACE
VERIFY

DOS READY
_
```

(Library Screen for NEWDOS/80 v1.0)



Apparat, Inc. introduces

NEWDOS/80

Apparat's long-awaited successor to NEWDOS+ is here! This is not an enhanced version of NEWDOS, but a completely new product. Simplified DOS commands can be instantly executed from BASIC, even within a program, without disturbing the resident code. System options, such as password protection, number and type of disk drives, BREAK key enable/disable and lowercase modification recognition, can be quickly and easily changed. Five new random-access file types allow record lengths of up to 4096 bytes, and no FIELDing! A powerful CHAIN facility allows keyboard INPUTs to be read from a disk file. An improved RENUMBER facility permits groups of statements to be relocated within program code. Diskettes may even be designated as RUN-ONLY! Features all NEWDOS+ utilities (SUPERZAP 3.0, etc.) and much more! One MTC technical staff member said having NEWDOS/80 is "better than sex" (you'll have to judge for yourself!). Includes 180-page instruction manual and MTC QUE card.

NEWDOS/80.................. $ 149.95

CALL REGARDING OUR NEWDOS+ UPGRADE PRICING.

NEWDOS/80 v1.0 Advertisement from MTC in the August 1980 issue of 80 Micro

NEWDOS/80 v2.0

NEWDOS/80 v1.0 was wonderful when compared against TRSDOS. As other competing alternative DOS's came on the scene (such as LDOS), however, NEWDOS/80 v1.0 started to pale.

As a result, Apparat released NEWDOS/80 v2.0 in June of 1981, featuring upgrades to the operating system which leave NEWDOS/80 as one of the few true standout DOS's for the TRS-80.

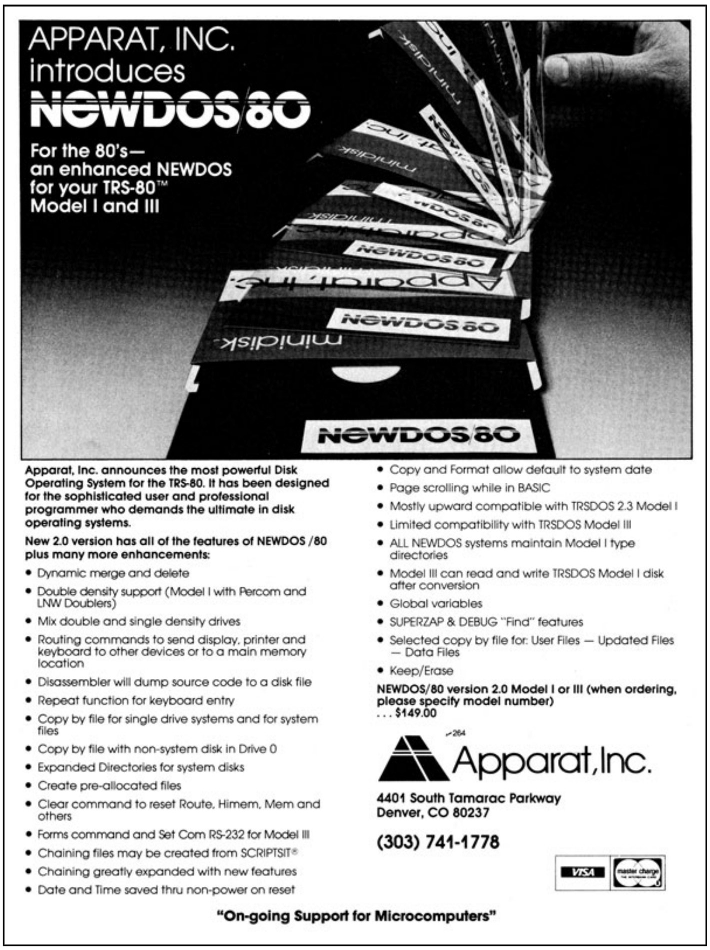
According to Apparat, the v2.0 updates were:

- Dynamic merge and delete
- Double density support (Model I with Percom and LNW Doublers)
- Mix double and single density drives
- Routing commands to send display, printer and keyboard to other devices or to a main memory location

- Disassembler will dump source code to a disk file
- Repeat function for keyboard entry
- Copy by file for single drive systems and for system files
- Copy by file with non-system disk in Drive 0
- Expanded Directories for system disks
- Create pre-allocated files
- Clear command to reset Route. Himem. Mem and others
- Forms command and Set Com RS-232 for Model III
- Chaining files may be created from SCRIPTSIT
- Chaining greatly expanded with new features
- Date and Time saved thru non-power on reset
- Copy and Format allow default to system date
- Page scrolling while in BASIC
- Mostly upward compatible with TRSDOS 2.3 Model I
- Limited compatibility with TRSDOS Model III
- All NEWDOS systems maintain Model I type directories
- Model III can read and write TRSDOS Model I disk after conversion
- Global variables
- SUPERZAP & DEBUG “Find” features
- Selected copy by file for: User Files - Updated Files - Data Files
- Keep/Erase

NEWDOS/80 v2.0 included a handful of utilities as follows:

- DIRCHECK would examine and repair a director, as well as provide extra file information.



APPARAT, INC.
introduces
NEWDOS/80

For the 80’s—
an enhanced NEWDOS
for your TRS-80™
Model I and III

NEWDOS/80
minidisk
Apparat, Inc.

Apparat, Inc. announces the most powerful Disk Operating System for the TRS-80. It has been designed for the sophisticated user and professional programmer who demands the ultimate in disk operating systems.

New 2.0 version has all of the features of NEWDOS /80 plus many more enhancements:

- Dynamic merge and delete
- Double density support (Model I with Percom and LNW Doublers)
- Mix double and single density drives
- Routing commands to send display, printer and keyboard to other devices or to a main memory location
- Disassembler will dump source code to a disk file
- Repeat function for keyboard entry
- Copy by file for single drive systems and for system files
- Copy by file with non-system disk in Drive 0
- Expanded Directories for system disks
- Create pre-allocated files
- Clear command to reset Route, Himem, Mem and others
- Forms command and Set Com RS-232 for Model III
- Chaining files may be created from SCRIPTSIT®
- Chaining greatly expanded with new features
- Date and Time saved thru non-power on reset
- Copy and Format allow default to system date
- Page scrolling while in BASIC
- Mostly upward compatible with TRSDOS 2.3 Model I
- Limited compatibility with TRSDOS Model III
- ALL NEWDOS systems maintain Model I type directories
- Model III can read and write TRSDOS Model I disk after conversion
- Global variables
- SUPERZAP & DEBUG "Find" features
- Selected copy by file for: User Files — Updated Files — Data Files
- Keep/Erase

NEWDOS/80 version 2.0 Model I or III (when ordering, please specify model number) . . . $149.00

✓264

Apparat, Inc.

4401 South Tamarac Parkway
Denver, CO 80237

(303) 741-1778

VISA master charge

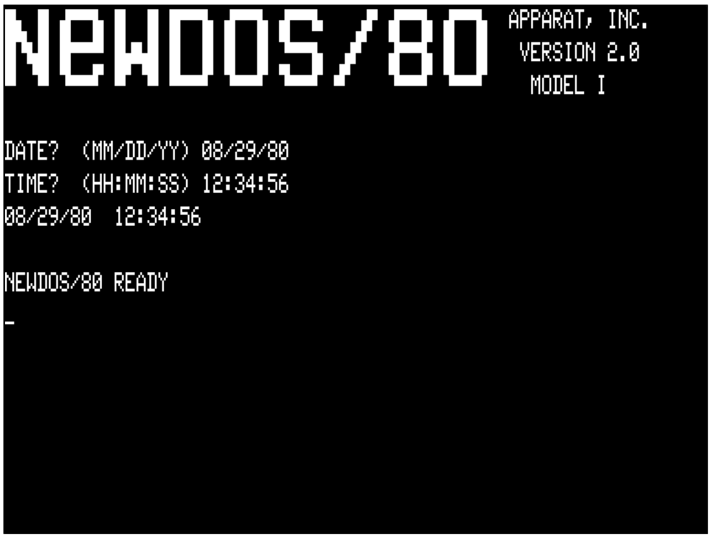
“On-going Support for Microcomputers”

(NEWDOS/80 v1.0 Advertisement from the June 1981 issue of 80 Micro)

NEWDOS/80
Apparat, Inc. Denver, Colorado
VERSION 2.0 - Model I - Reg. # 28168

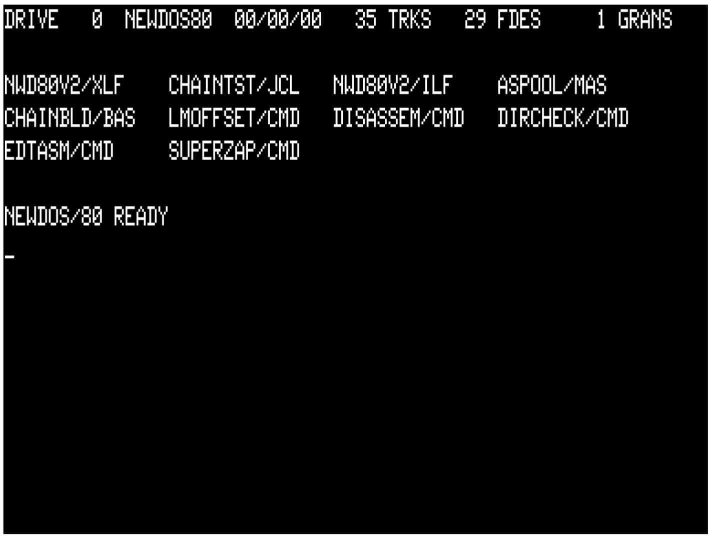
(NEWDOS/80 v2.0 Original Diskette)

- LMOFFSET (Load Module OFFSET) would allow you to tranfer a machine language (SYSTEM) program from cassette to diskette. In doing so, however, the program would be modified to be relocated to a different area of memory. Of course, functions within the program which still relied on tape I/O would no longer work, as the interrupts necessary to disk I/O would not be disabled (as is needed for tape I/O). Radio Shack offered “TAPEDISK” to do the same thing, but LMOFFSET did it much better.
- DISASSEM would disassemble a machine language (/CMD) file. One of the most important options was to save the source to diskette, which would then allow you to modify and possibly recompile.
- SUPERZAP could be used to modify disk and file sectors
- EDTASM was a disk based version of Tandy’s machine language programming environment Editor/Assembler.
- ASPOOL (written by H.S. Gentry) was a printer spooler which, when enabled, would spool print jobs to DISK instead of sending them to the printer which would free up the computer for other tasks.
- CHAINBLD was a program which would aid in creating Job Control Language (/JCL) files to allow for (limited) batch operation.



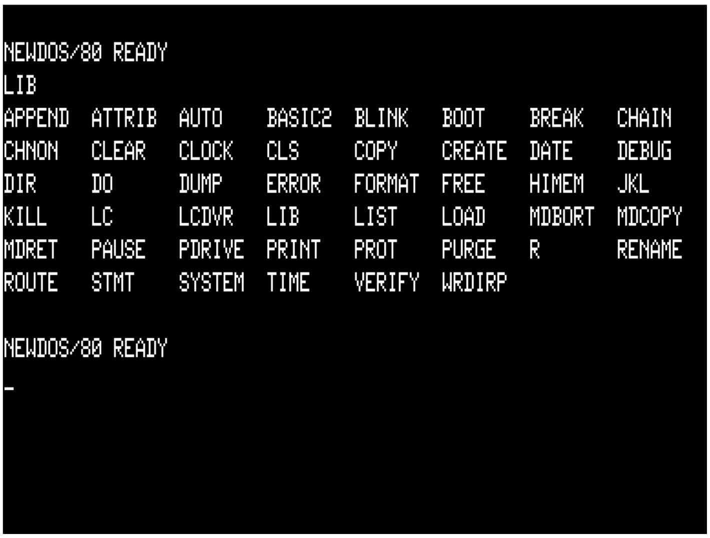
```
NEWDOS/80  APPARAT, INC.
            VERSION 2.0
             MODEL I

DATE?  (MM/DD/YY) 08/29/80
TIME?  (HH:MM:SS) 12:34:56
08/29/80  12:34:56

NEWDOS/80 READY
_
```

(Bootup Screen for NEWDOS/80 v2.0)



```
DRIVE   0  NEWDOS80  00/00/00   35 TRKS   29 FDES     1 GRANS

NWD80V2/XLF    CHAINTST/JCL   NWD80V2/ILF    ASPOOL/MAS
CHAINBLD/BAS   LMOFFSET/CMD   DISASSEM/CMD   DIRCHECK/CMD
EDTASM/CMD     SUPERZAP/CMD

NEWDOS/80 READY
_
```

(Directory Screen for NEWDOS/80 v2.0)



```
NEWDOS/80 READY
LIB
APPEND  ATTRIB  AUTO    BASIC2  BLINK   BOOT    BREAK   CHAIN
CHNON   CLEAR   CLOCK   CLS     COPY    CREATE  DATE    DEBUG
DIR     DO      DUMP    ERROR   FORMAT  FREE    HIMEM   JKL
KILL    LC      LCDVR   LIB     LIST    LOAD    MDBORT  MDCOPY
MDRET   PAUSE   PDRIVE  PRINT   PROT    PURGE   R       RENAME
ROUTE   STMT    SYSTEM  TIME    VERIFY  WRDIRP

NEWDOS/80 READY
_
```

(Library Screen for NEWDOS/80 v2.0)

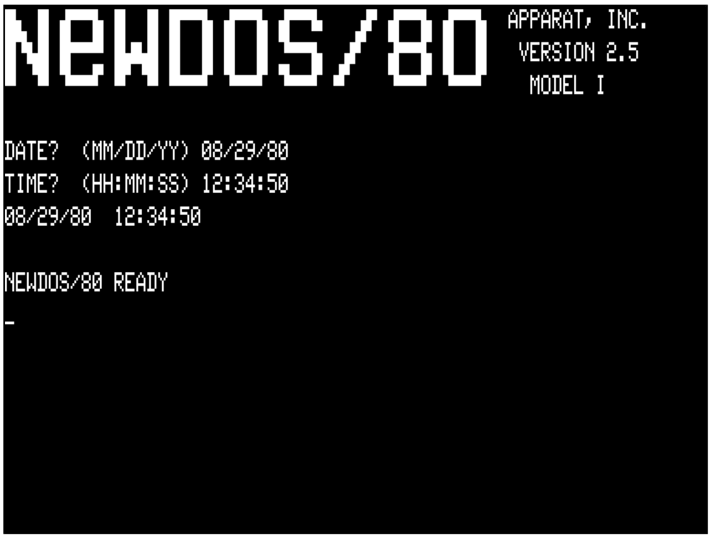
The DOS consisted of 14 overlay modules and BASIC consisted of one program and 8 overlays. Depending on your needs, you could delete unneeded overlays and free up some disk space. A breakdown of the /SYS files and their purposes is found here.

One feature of note was the updated COPY command, which could be used to copy files as well as whole disks, was extremely powerful. There were options to format the destination drive (FMT) or leave it as is (NFMT), warn if the destination drive has files (N) or just go ahead (Y), prompt for disk mounts (for those with fewer drives than needed) or not (NDMW), custom source pdrive specifications (SPDN=x) and custom destination (DPDN=x) so that you wouldn’t have to change your PDRIVE table just for the copying function, checks based on the source drive password (SPW=x) or the destination drive’s name (ODN=x) with prompting (DDND), options to keep the destination disks name (KDN) and date (KDD) or to use the source disks date (USD). While there were no wildcards available, you could instruct NEWDOS/80 v2.0 to copy by file (CBF) and then to choose only user files (files which were not system and not invisible) (USR), by a specific extension (/ext), only updated files (UPD), files in a pre-created text list (ILF=x), and files NOT in a pre-created text list (XLF=x).

Three hot-key combinations were also present: [1][2][3] involked DEBUG, [D][F][G] invokled mini-dos, and [J][K][L] printed the screen. There were third party programs which could be run which would convert screen graphics to MX-80 graphics if JKL was hit.

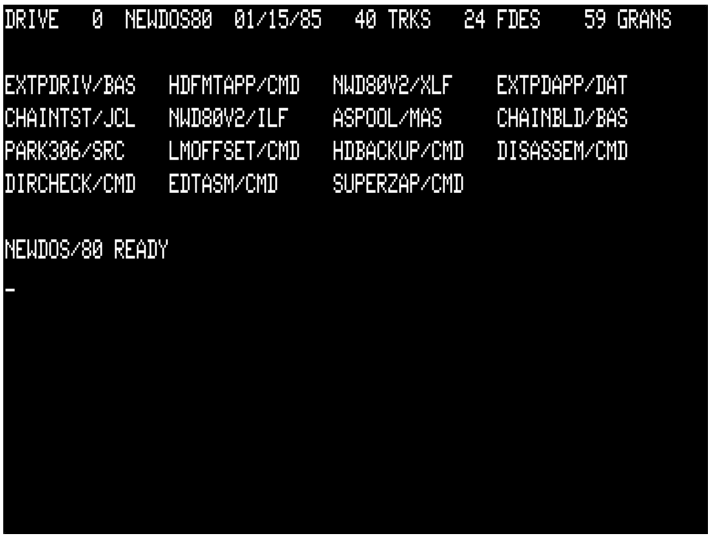
NEWDOS/80 v2.5

NEWDOS/80 v2.5 was released adding Hard Drive support to the operating system.



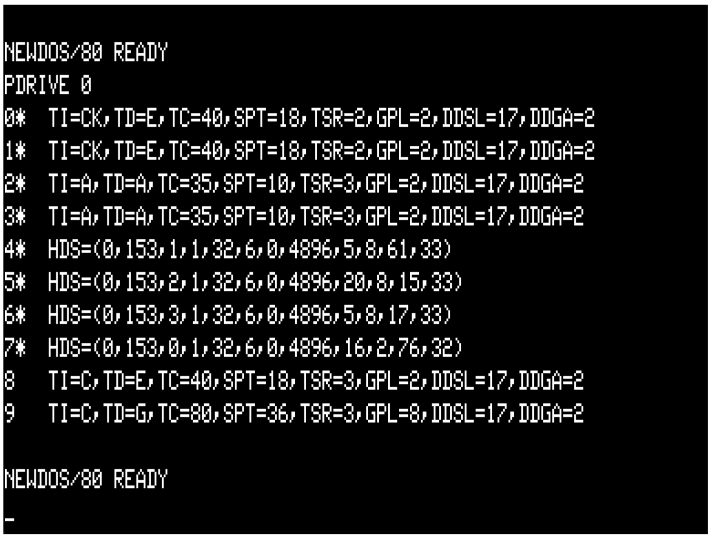
```
NEWDOS/80   APPARAT, INC.
             VERSION 2.5
              MODEL I

DATE?  (MM/DD/YY) 08/29/80
TIME?  (HH:MM:SS) 12:34:50
08/29/80  12:34:50

NEWDOS/80 READY
_
```

(Bootup Screen for NEWDOS/80 v2.5)



```
DRIVE   0  NEWDOS80  01/15/85   40 TRKS   24 FDES    59 GRANS

EXTPDRIV/BAS   HDFMTAPP/CMD   NWD80V2/XLF    EXTPDAPP/DAT
CHAINTST/JCL   NWD80V2/ILF    ASPOOL/MAS     CHAINBLD/BAS
PARK306/SRC    LMOFFSET/CMD   HDBACKUP/CMD   DISASSEM/CMD
DIRCHECK/CMD   EDTASM/CMD     SUPERZAP/CMD

NEWDOS/80 READY
_
```

(Directory Screen for NEWDOS/80 v2.5)

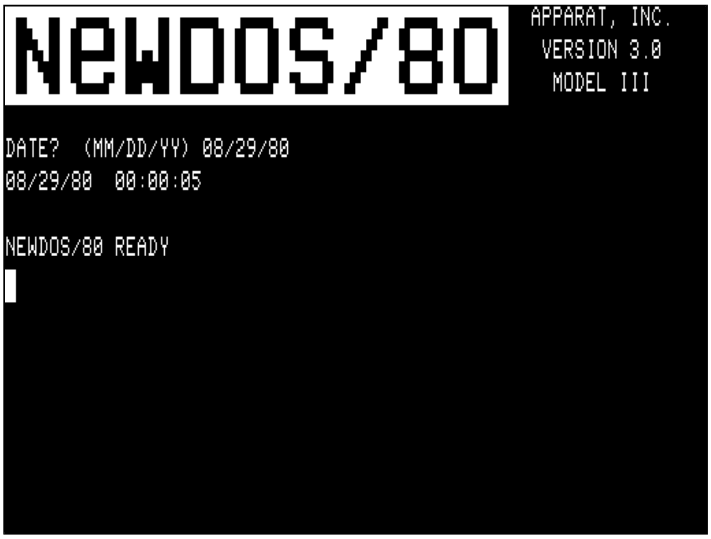


```
NEWDOS/80 READY
PDRIVE 0
0*  TI=CK,TD=E,TC=40,SPT=18,TSR=2,GPL=2,DDSL=17,DDGA=2
1*  TI=CK,TD=E,TC=40,SPT=18,TSR=2,GPL=2,DDSL=17,DDGA=2
2*  TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
3*  TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
4*  HDS=(0,153,1,1,32,6,0,4896,5,8,61,33)
5*  HDS=(0,153,2,1,32,6,0,4896,20,8,15,33)
6*  HDS=(0,153,3,1,32,6,0,4896,5,8,17,33)
7*  HDS=(0,153,0,1,32,6,0,4896,16,2,76,32)
8   TI=C,TD=E,TC=40,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
9   TI=C,TD=G,TC=80,SPT=36,TSR=3,GPL=8,DDSL=17,DDGA=2

NEWDOS/80 READY
_
```

(PDRIVE Screen for NEWDOS/80 v2.5 - Entries 4 through 7 are Hard Drives)

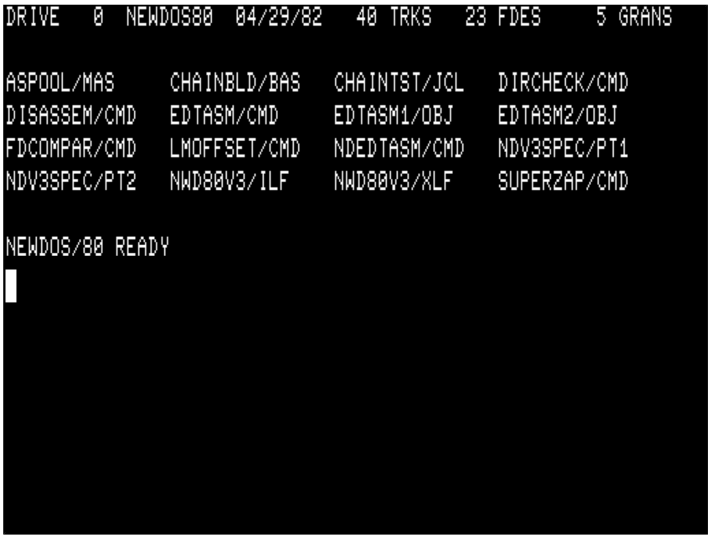
NEWDOS/80 v3.0, NEWDOS/86, and NEWDOS/90

These DOS’s have appeared in the wild, but I am unsure as to whether they are true releases by Apparat, Inc. and what they bring to the table, if anything.



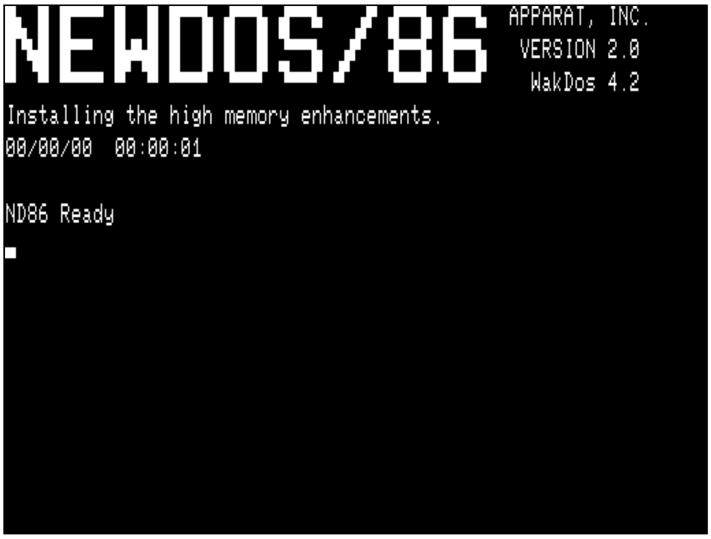
```
NEWDOS/80   APPARAT, INC.
             VERSION 3.0
              MODEL III

DATE?  (MM/DD/YY) 08/29/80
08/29/80  00:00:05

NEWDOS/80 READY
```

(NEWDOS/80 v3.0 Initial Boot Screen)



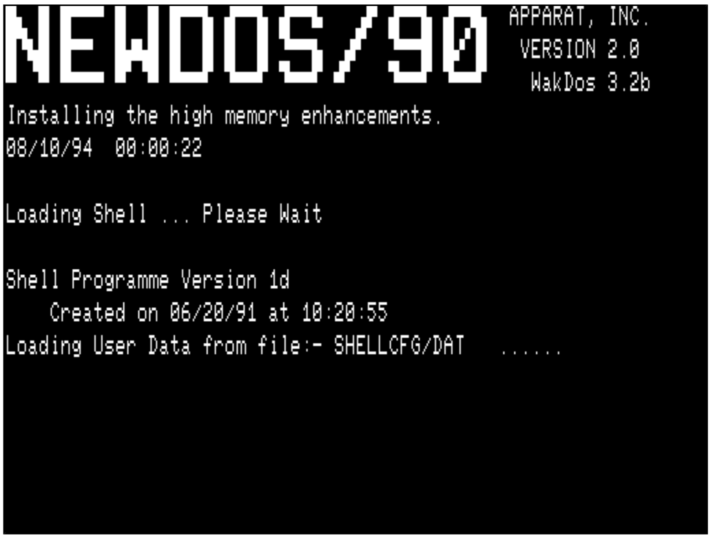
```
DRIVE   0  NEWDOS80  04/29/82   40 TRKS   23 FDES     5 GRANS

ASPOOL/MAS     CHAINBLD/BAS    CHAINTST/JCL    DIRCHECK/CMD
DISASSEM/CMD   EDTASM/CMD      EDTASM1/OBJ     EDTASM2/OBJ
FDCOMPAR/CMD   LMOFFSET/CMD    NDEDTASM/CMD    NDV3SPEC/PT1
NDV3SPEC/PT2   NWD80V3/ILF     NWD80V3/XLF     SUPERZAP/CMD

NEWDOS/80 READY
```

(NEWDOS/80 v3.0 Directory)



```
NEWDOS/86   APPARAT, INC.
             VERSION 2.0
              WakDos 4.2
Installing the high memory enhancements.
00/00/00  00:00:01

ND86 Ready
```

(NEWDOS/86 Initial Boot Screen)



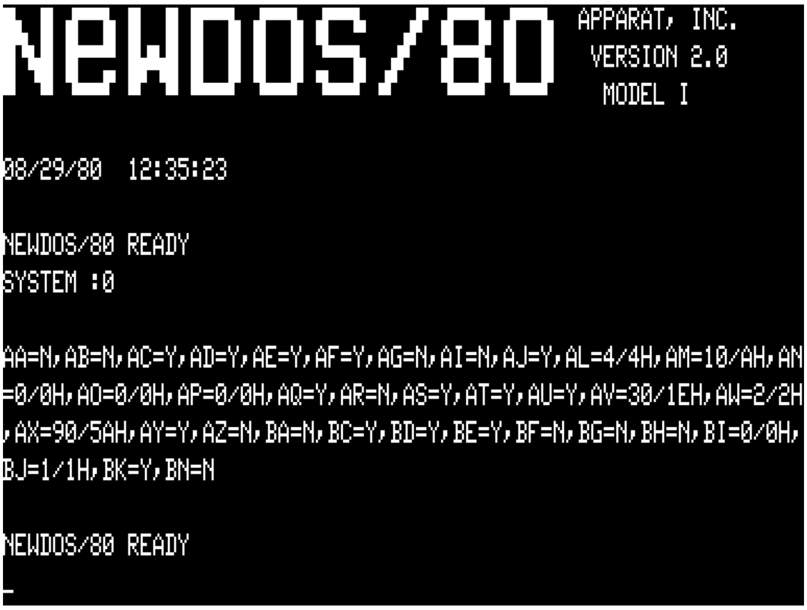
```
NEWDOS/90   APPARAT, INC.
             VERSION 2.0
              WakDos 3.2b
Installing the high memory enhancements.
08/10/94  00:00:22

Loading Shell ... Please Wait

Shell Programme Version 1d
    Created on 06/20/91 at 10:20:55
Loading User Data from file:- SHELLCFG/DAT   ......
```

(NEWDOS/90 Initial Boot Screen)

SYSTEM Command

The SYSTEM command allowed for customization as to many parts of the NEWDOS/80 v2.x operating system.

From the manual: USE: *SYSTEM,pwdn1,options*. PURPOSE: Change and/or display the system options in the control sector of the system diskette mounted on drive dn1. These options determine the system configuration at subsequent re-boots when this diskette is the system diskette.



```
NEWDOS/80   APPARAT, INC.
             VERSION 2.0
              MODEL I

08/29/80  12:35:23

NEWDOS/80 READY
SYSTEM :0

AA=N,AB=N,AC=Y,AD=Y,AE=Y,AF=Y,AG=N,AI=N,AJ=Y,AL=4/4H,AM=10/AH,AN
=0/0H,AO=0/0H,AP=0/0H,AQ=Y,AR=N,AS=Y,AT=Y,AU=Y,AV=30/1EH,AW=2/2H
,AX=90/5AH,AY=Y,AZ=N,BA=N,BC=Y,BD=Y,BE=Y,BF=N,BG=N,BH=N,BI=0/0H,
BJ=1/1H,BK=Y,BN=N

NEWDOS/80 READY
_
```

The SYSTEM commands were as follows:

AA=yn	Passwords enabled?
AB=yn	Run only mode enabled?

AC=yn	Keyboard debounce enabled? (Model I only);
AD=yn	JKL for screen print enabled?
AE=yn	123 for debug enabled?
AF=yn	DFG for MINI-DOS enabled?
AI=yn	Lower case mode installed? (Model I only);
AJ=yn	DOS’s keyboard intercept routine enabled? If N, see manual for side effects, especially Chaining;
AL=al	al = the number (1 to 4) of physical drives in the system;
AM=am	am = disk I/O tries (i.e. 10) before error declared;
AN=an	The DIR command default drive number is an};
AO=ao	Drive ao is first and lowest drive used when creating a new file where a drive is not specified;
AP=ap	Default HIMEM address;
AQ=yn	CLEAR key enabled?
AR=yn	COPY formats 5 and 6 allowed without diskette password checking?
AS=yn	BASIC input text strings forced to upper case? (Model I only);
AT=yn	Chaining in byte mode instead of record?
AU=yn	Use clock driven repeat key function?
AV=av	If AU=Y delay av 25ms intervals before key’s first repeat;
AW=aw	Number (i.e. 3) of write with verify disk I/O tries before declaring error;
AX=ax	ax is the highest ASCII code for the printer;
AY=yn	Ask for date and time at DOS unrecognizable reboots?
AZ=yn	Ask for date and time at DOS recognizable boots?
BA=yn	Set ROUTE,DO,NL state?
BB=yn	Clock interrupts 50 times per second instead of 60? (Model III only);
BC=yn	Operator allowed to pause or cancel chaining?
BD=yn	Hold down ENTER during reboot overrides AUTO command?
BE=yn	DOS command R enabled?
BF=yn	Set LCDVR,yn state (Model I only);
BG=yn	Set LC,yn state;
BH=yn	Set BLINK,yn state;
BI=bi	Set cursor character = bi (i.e. 176);
BJ=bj	bj is integer CPU speed multiple (i.e. 1, 2, 3) to help DOS cope with a faster CPU;
BK=yn	WRDIRP command and DIRCHECK functions W and C enabled?
BN=yn	Write directory sectors readable by Model III NEWDOS/80 instead of by Model I TRSDOS (Model I only);

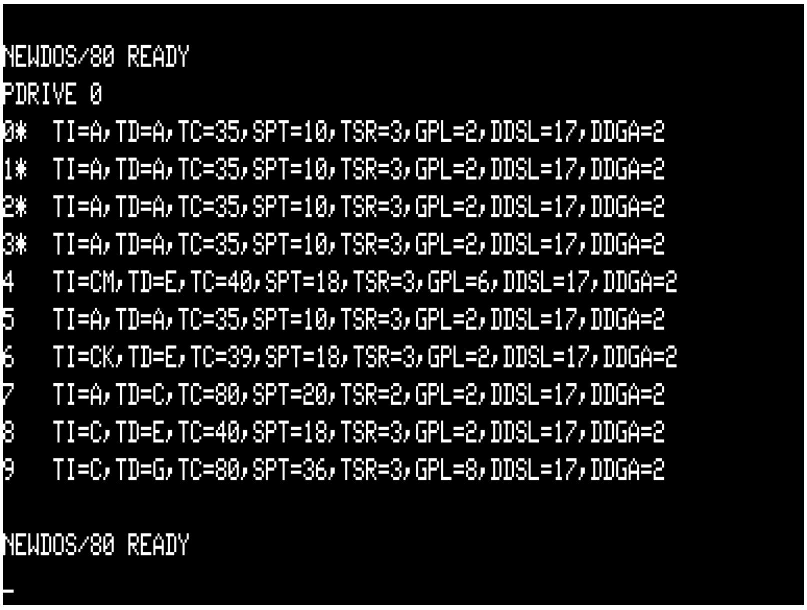
PDRIVE Command

The PDRIVE command allowed for the setting, on a disk drive by disk drive basis, the type of interface, drive type, and diskette characteristcs, making NEWDOS/80 infinitely versatile when it came to reading alien formats. Of course, you needed to know the details of those formats, although there were 3rd party applications which would attempt to automatically figure that out. The PDRIVE table was written to disk, and would require a reboot after each change unless the suffix

“,A” was added, in which case no reboot was required. There was also a patch to NEWDOS/80 to suffix with “,B” which would make the PDRIVE table changes in RAM but not on the disk. One major glitch in the use of PDRIVE was that, under certain circumstances, if you had mis-entered parameters so that they conflicted, the DOS would no longer boot.

From the manual: USE: *PDRIVE,pwdn1,dn2,options*. PURPOSE: Update drive dn2’s data and display drive dn1 diskette’s PDRIVE table.

The PDRIVE settings were as follows:



```
NEWDOS/80 READY
PDRIVE 0
0*  TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
1*  TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
2*  TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
3*  TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
4   TI=CM,TD=E,TC=40,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA=2
5   TI=A,TD=A,TC=35,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
6   TI=CK,TD=E,TC=39,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
7   TI=A,TD=C,TC=80,SPT=20,TSR=2,GPL=2,DDSL=17,DDGA=2
8   TI=C,TD=E,TC=40,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
9   TI=C,TD=G,TC=80,SPT=36,TSR=3,GPL=8,DDSL=17,DDGA=2

NEWDOS/80 READY
_
```

(PDRIVE from NEWDOS/80 v2.0)

TI=type1		Type of Interface, consisting of one or more of the following (see manual for proper combinations):
	A -	Standard TRS-80 disk interface;
	B -	Model I only OMIKRON mapper;
	C -	Model I only PERCOM double interface;
	D -	Model III only Apparat disk controller;
	E -	Model I only LNW type disk interface;
	H -	Head settle delay is to be done on drive change. Required for 8-inch drives;
	I -	Sector 1 is lowest numbered sector on each track;
	J -	Track 1 is the lowest numbered track on the diskette;
	K -	Track 0 is formatted in opposite density from the rest of the tracks and is usable only during boot;
	L -	Two step pulses between tracks are used;
	M -	Model III TRSDOS or Model I TRSDOS 2.3B or higher diskettes to be mounted.
TD=type		Type of Drive where type values are:
	A -	5 inch, single density, single sided;
	B -	8 inch, single density, single sided;
	C -	5 inch, single density, double sided;
	D -	8 inch, single density, double sided;
	E -	5 inch, double density, single sided;
	F -	8 inch, double density, single sided;
	G -	5 inch, double density, double sided;
	H -	8 inch, double density, double sided.
TC=tc1		tc1 is the number of tracks on diskettes mounted on the drive;
SPT=sc1		sc1 is the number of sectors per track;
TSR=rc1		rc1 is the track stepping rate code. 0=5ms, 1=10ms, 2=20ms, 3=40ms;
GPL=gc2		gc2 is the number of granules (2 to 8) per lump;
DDSL=ln1		ln1 is the number of the lump whose first sector is the directory’s first sector. Used only during formatting;
DDGA=gc1		gc1 is the number of granules (2 to 6) assigned to the directory. Used only during formatting;
A		If the display shows no errors, activate the PDRIVE attributes as the current attributes in

main memory. Subsequent I/O will use these attributes. If A not specified, subsequent I/O (except that via SPDN or DDPN [see COPY]) will not use any new attributes until re-boot.

Booting NEWDOS/80 v2.0 on a Model 4P - Chris Anderson

Note! This procedure is NOT necessary for a Model 4! It is only of use for Model 4P owners!

You who own Model 4P's probably know by now that you are missing something that a Model 4 owner finds very handy - THE MODEL 3 ROM! Without that ROM, you cannot directly boot several other popular operating systems. To boot them, you must first boot using a disk that contains a file called MODELx/III. Most systems came with MODELA/III. Then you must reboot using the operating system of your choice. MODELA/III is a copy of the Model 3 ROM, and is loaded down in low memory, making it appear to the machine exactly as it would if there really WAS a ROM down there as there is on a Mod 3. By doing this, operating systems other than LDOS and CPM have the ROM routines available as needed for operation.

SO… HOW to get a 4P to boot NEWDOS directly without first having to boot LDOS, load MODELA/III, and boot again? Easy.. fool the system (what else?) The following procedure creates a pseudo-directory on a NEWDOS disk. This directory will be properly located on track 20 so that it can be found by the bootstrap routine in the 4P. Immediately following the pseudo-directory will be a copy of MODELA/III. Both will be saved together as a single file. This file will also be entered into the REAL NEWDOS directory (so we don't accidently overwrite it with other files) and just so that you know what it is, it'll be called MODELA/III.

So what happens? When LDOS tries to boot the disk, it finds an entry for MODELA/III in the area of the disk where LDOS looks for the directory. This dummy directory points a little further down into the file where MODELA/III actually exists! Confused? What can I say. Trust me on this one folks; it works.

Boot the system as follows using LDOS6.2 This technique is the norm for booting foreign systems on the 4P.

a) Install any LDOS6.2 disk that contains the program MODELA/III, assuming you have one so prepared. If so, you will be prompted to install the disk containing the foreign system (NEWDOS, in this case) and you may skip the next step. If not, use your MODELA/III disk to boot, following step b) instead.

b) Insert your MODELA/III and press the reset button. IMMEDIATELY press and hold down the F3 and P keys. You will then receive the prompt to switch disks.

c) Install a VIRGIN copy of NEWDOS80 V2 in Drive 0. It should contain the NEWDOS program called SUPERZAP/CMD and ALL /SYS files should be present.

d) Set the PDRIVE for Drive 1 as follows:

```
PDRIVE 0,1,DDSL=29,A
```

e) Format and copy the disk on 0 to 1 as follows, using whatever you like for the current date:

```
COPY 0,1,11/25/87,CBF,/SYS,FMT
```

f) Remove the virgin disk and install the newly created NEWDOS disk into Drive 0. Put the virgin disk into Drive 1 so you'll have use of the SUPERZAP program.

g) Create a file entry called MODELA/III as follows:

```
CREATE MODELA/III:0
```

h) Use SUPERZAP to modify the directory entry for MODELA/III on the disk in Drive 1 as noted below. If you have used a virgin disk, you will find the directory entry for MODELA/III at disk sector 299.

```
FROM: 1000 0000 004D 4F44 454C 4120 2049 4949
      9642 9642 0000 FFFF FFFF FFFF FFFF FFFF
TO:   1000 0000 004D 4F44 454C 4120 2049 4949
      9642 9642 0000 3600 FFFF FFFF FFFF FFFF
```

The only actual change is the 4th pair on the second line of the entry from FFFF to 3600.

i) Remove the virgin disk from Drive 1 and install the disk that came with your Model 4P system called MODELA/III. (Yeah, I know it's an LDOS disk. Be cool..)

j) Check to make sure your PDRIVE setting is as noted below on Drive 0, entry for Drive 4 (it will be if you're using a virgin disk like I told you to!)

```
TI=AM,TD=E,TC=40,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA=2
```

k) If not, fix it!

l) Copy the program MODELA/III to the NEWDOS disk with the following command:

```
COPY MODELA/III:1 :0 SPDN=4
```

m) Remove the MODELA/III disk from Drive 1 and reinstall the disk containing SUPERZAP.

n) Using SUPERZAP, Copy Disk Sectors (CDS) on Drive 0. You want to copy 10 sectors from 290 to 522. This makes a duplicate directory up in an unused region.

o) Using SUPERZAP, modify the entry for MODELA/III in the "fake" directory as follows. You should find the entry in drive relative sector 531.

```
FROM: 1020 0000 004D 4F44 454C 4120 2049 4949
      9642 9642 3900 360B FFFF FFFF FFFF FFFF
TO:   1000 0000 004D 4F44 454C 4120 2049 4949
      9642 9642 3900 1E09 FFFF FFFF FFFF FFFF
```

Note that the only actual changes were from 1020 to 1000 in the top line, and from 360B to 1E09 in the second line.

p) Using SUPERZAP, modify the entry for MODELA/III in the REAL directory as follows. The entry will be at drive relative sector 299 like it was the first time you diddled with it:

```
FROM: 1020 0000 004D 4F44 454C 4120 2049 4949
      9642 9642 3900 360B FFFF FFFF FFFF FFFF
TO:   1000 0000 004D 4F44 454C 4120 2049 4949
      9642 9642 4D00 340F FFFF FFFF FFFF FFFF
```

Note that the changes here are from 1020 to 1000 in the first line, and from 3900 to 4D00 and from 360B to 340F in the second line.

q) Since you have a virgin copy of NEWDOS in Drive 1 (right?) you should have DIRCHECK/CMD on it as it came from Apparat. Execute DIRCHECK, specifying Drive 0 as the target drive. Everything being the way it ought to be, you should get the following error messages at the end of the program:

```
34,0 ***** GRANULE FREE, BUT ASSIGNED TO FILE(S)...
34,1 *****  "
35,0 *****  "   (Each should be shown as assigned
35,1 *****  "    to FILE(S) 67 MODELA/III.)
36,0 *****  "
```

r) Using SUPERZAP, go to the GAT sector (drive relative sector 290) and correct the errors caught by running DIRCHECK:

```
Change relative byte 34 from FC to FF
Change relative byte 35 from FC to FF
Change relative byte 36 from FE to FF
```

ALL DONE! You now have a bootable NEWDOS disk. When you remember what life is like without the PDRIVE command, you'll believe it was worth it!

Another Method by Russ McElroy, CIS 72235,615

You will need two SS 40 track disk drives, NEWDOS80 (and SUPERZAP), and a TRSDOS 6.x system disk (with MODELA/III). What we will do is create a new system file 'SYS22/SYS' which will consist of the MODELA/III file preceeded by a mock TRSDOS 1.3 FPDE positioned so the 4P will find it on bootup. The 4P will use this mock directory entry to locate the ROM image and will then proceed to load it, thereby booting without the use of a TRSDOS 1.3 diskette!

One caveat: I tried using the MODELA/III file supplied with LSDOS 6.3, but found it would not work correctly. Using the original version I got with my 4P solved the problem. If, upon booting your zapped NEWDOS80 system, you find yourself with no repeat keys (ie. hold down a key and see if it repeats), it is likely you'll need to use an older version of MODELA/III.

1. Boot up a NEWDOS80 system and set up the PDRIVE table to copy the system: PDRIVE 0,1=0. Press reset to load the new pdrive info.
2. Make a backup of the NEWDOS80 system onto a disk in drive 1 using:

   ```
   COPY,0,1,,CBF,FMT,/SYS,NDMW
   ```

   This copies only /SYS files onto the new system disk.
3. Put the mock TRSDOS 1.3 FPDE on the new system disk using SUPERZAP.
   At DRS 308, starting at byte 00 enter the following bytes:

   ```
   100A 5400 004D 4F44 454C 4120 2049 4949
   9642 9642 3900 1133 FFFF FFFF FFFF FFFF
   ```

4. Update the GAT to reflect the space to be occupied by the SYS22/SYS file.
   At DRS 170, starting at byte 1EH enter the following bytes:

   ```
   FEFF FFFF FFFF FF
   ```

   This lets the system know that this space is occupied.
5. Add the HIT entry for the SYS22/SYS file.
   At DRS 171, byte 60H enter the following byte

   ```
   25
   ```

   This is the hash code for 'SYS22/SYS' in the proper position so the system can locate the FPDE.
6. Use the CDS function of SUPERZAP to copy the 57 sectors of the MODELA/III file from a TRSDOS 6.x system disk to the new system disk.
   To do this, mount the TRSDOS disk in drive 0 and copy 57 sectors from DRS 660 of the TRSDOS disk to DRS 312 of the new NEWDOS80 system disk in drive 1.

That's all there is to it!

Another Method

The MODELA/III file must be loaded from a diskette in the TRSDOS 1.x, LDOS 5 or LS-DOS 6.x format. The 4P Boot ROM doesn't understand any of the other directory/filesystem layouts, and by default the NEWDOS disk format is quite different.*

The normal process for booting an "alien" format diskette is:

```
1.      Insert the diskette with MODELA/III on it and press reset.
2.      If that diskette is LS-DOS 6, press the [3] or [F3] key within three
        seconds.
3.      Press the [P] key to indicate that you want to swap diskettes.
        You must also do this within the three seconds after pressing RESET.
        (You can press the F3, 3 and P keys in any order or all at once.)
4.      The system prompts to change diskettes.  Insert NEWDOS, MultiDOS
        or whatever non-TRSDOS/non-LDOS format OS this is.
5.      Press [ENTER] to boot the other operating system when the new diskette
        is in place.
```

Once loaded, if you press RESET or otherwise reboot, the MODELA/III image will still be there and you can boot directly onto the NEWDOS or other Model III OS without having to swap diskettes again or pressing any other keys. If you cycle power, you will have to repeat the first procedure.

Optional keys related to Model III-mode booting are [L], which means says to load the Model III image even if it doesn't appear to be needed, and [N], which says to NOT load the ROM-image even if it does appear to be needed.

* Implied in that is if you make a NEWDOS diskette with a directory cylinder pointer in the boot sector that points to a cylinder that "looks" like a LDOS/LS-DOS diskette directory track, and it contains a LDOS/LS-DOS/TRSDOS-style directory entry that points to a cylinder/sector position containing the MODELA/III file

data, *then* you could load from a non-TRSDOS/LDOS/LSDOS OS, and someone has done this, but you have to make a very strange looking disk layout to make it occur and it wastes a lot of space. This exercise is left to those with lots of spare time.

Another Method … For the more adventuresome

Build a diskette with a boot sector that doesn't look like a Model III-mode boot sector. This means the ROM won't try to load MODELA/III at all, assuming it's a Model 4-mode diskette. Then when the boot sector is loaded and control is transferred to it, it can do anything it wants to. You could write a boot sector that knows where to load the MODELA/III file from on this alien disk format and load it yourself before loading the rest of the OS, or you could concatenate the BOOT/SYS or SYS0/SYS modules of the operating system onto the contents of a MODELA/III file, which loaded everything as though it was one big file. Typically the MODEL%/III contents needs to be loaded first, so that later stages of the boot process will have the old Model III ROM functions available for use, such as to print those boot-up screens.

In theory, you could do the latter (build a blob file containing both MODELA/III and BOOT/SYS contents) and let the boot ROM do the work (ie load MODELA/III contents and BOOT/SYS that have been combined in one file), but it would be much easier to simply not let the boot ROM try to load a MODEL%/III file and take over things from there. You can't make the 4P-ROM know about any new disk formats, so the best choice is to not rely on it at all for that function.

To decide if the MODELA/III file is needed, the 4P ROM scans sector 1 on the floppy diskette looking for the CD XX 00 (CALL 00XXH) instruction sequences and other clues. If it finds any, it knows the boot sector is expecting the Model III ROM image to be there, so the 4P-ROM loads the ROM image first before turning control over to the boot sector. Frequently, the only reference to the "A" ROM address range in the boot sector would be to display a message like "DISK ERROR", but that one hit was enough to make the determination automatic. I recall that during development back in 1983, Frank Durda IV tested several dozen alien diskette formats, including all the Kim Watt stuff, looking for signatures like this.

Another solution, by Tony Domigan and Northern Bytes, Vol.5, #7, Page 14

MODELA/III can be copied to LDOS 5.x and TRSDOS 1.3B diskettes so that they can Cold Start on the Model 4P. If, however, you wish to use NEWDOS/80 version 2.0 the only option is to load MODELA/Ill with 'pause' and then enter your NEWDOS disk - two operations not very convenient. Furthermore, the Model 4P hardware seems to check the boot sector (sector 1) in a way which excludes any DOS other than TRSDOS or LDOS.

The method I have used relocates the NEWDOS directory to lump 36 i.e. track 20; the same track on which the LDOS directory resides,

The FPDE for MODELA/III is created and the directory entry modified so that when the file is copied from the TRSDOS MODELA/Ill disk it will be copied to a true track. I have used track 30 as it is not allocated in a standard NEWDOS diskette,

The ROM file is then modified on the NEWDOS diskette so that the bootstrap loader will select sector 0 to boot the diskette.

Using Superzap the MODELA/III FPDE is again modified to read Track lEH or 30 decimal and the total granules changed to 09H. The normal boot sector (sector 1) is modified with 3 bytes located by the hardware in checking for a TRSDOS or LDOS disk.

The diskette is now configured such that the hardware assumes it is an LDOS diskette with the ROM starting on track 30 decimal, Once loaded the ROM bootstrap boots NEWDOS from sector 0.

1. Construct a new system diskette with a directory on lump 36 - e.g.

```
PDRIVE,0,1,DDSL=36,A
COPY,0,1,00/00/00,CBF,FMT,NDP=NEWDOS4P,DPDN=1
```

2. Boot the new system diskette,
3. Create a file called MODELA/III - e.g.

```
CREATE MODELA/III:0
```

4. Using Superzap, page through the directory, from relative sector 360, till the FPDE for MODELA/Ill is

located, Modify Bytes 21-24 in FPDE i.e.

```
0000 FFFF to 3900 36DB
```

5. From ‘NEWDOS READY” copy MODELA/Ill from the TRSDOS 1.3B boot disk to the new NEWDOS disk - e.g.
   ```
   COPY MODELA/III:1 :0 SPDN=4
   ```
6. Using Superzap locate the MODELA/III FPDE and modify bytes 20-24 i.e.
   ```
   Find 3900 360B Change to 3900 1E09
   ```
7. Using Superzap ‘DFS’ modify MODELA/Ill at FRS 53 Relative Byte 39 i.e.
   ```
   Find 3E01 D3F2 Change to 3E00 D3F2
   ```
8. Using Superzap modify the first sector (360) of the directory (GAT). Change relative bytes 36H to 3BH inclusive to FF (i.e. lump(s) allocated),
9. Modify Disk Relative Sector 1 at relative byte 0: Do a ‘ZTFF’ to zero the sector and modify:
   ```
   Relative Byte 02 ==> 14
   Relative Byte 14 ==> 28
   Relative Byte SA ==> CD
   ```
10. Now the easy part, RESET the 4P holding the ‘P’ and ‘L’ keys, If the ROM fails to load successfully then recheck the FPDE entry and the Sector 1 bytes. If the ROM loads successfully but NEWDOS fails, either partly or completely, then recheck the patch to MODELA/Ill (or perhaps you have accidentaliy written to the wrong part of the system!).

NEWDOS/80 System Files and Purposes - Art McAninch

/sys0	Essential
/sys1	Essential - Interrogates DOS commands
/sys2	Necessary - Creates files, opens FCBs, allocates file space, allocates FDEs, encodes passwords and loads user programs. Executor for RENAME and LOAD.
/sys3	Necessary. Closes FCBs, kills files, inserts/deletes entries. Executor for BLINK, BREAK, CLOCK, DEBUG, JKL, LCDVR, LC, VERIFY, and PURGE.
/sys4	Displays DOS ERROR messages
/sys5	DEBUG
/sys6	Executor for FORMAT, COPY, and APPEND.
/sys7	Executor for TIME, DATE, AUTO, ATTRIB, PROT, DUMP, HIMEM, and the first part of PURGE, SYSTEM, and PDRIVE.
/sys8	Executor for DIR and FREE.
/sys9	Executor for BASIC2, BOOT, CHAIN, CHNON, MDCOPY, PAUSE, and STMT.
/sys10	Executes BASIC statements GET and PUT.
/sys11	Executes BASIC direct statement RENUM.
/sys12	Executes BASIC direct command REF.
/sys13	Executes BASIC ERROR messages and the first part of RENUM.
/sys14	Executor for CLEAR, CREATE, ERROR, LIST, PRINT, and ROUTE.
/sys15	Executor for FORMS and SETCOM.
/sys16	Executor for most of PDRIVE.
/sys17	Executor for WRDIRP and most of SYSTEM.
/sys18	BASIC direct statement executor.
/sys19	Executor for BASIC statements LOAD, RUN, MERGE, SAVE, and CMD”F”.

/sys20	Executor for a number of BASIC statements. Must be present if BASIC is active.
/sys21	Executor for CMD”O”.

NEWDOS/80 v2.0 Other-DOS PDRIVE Settings

- Phill Walsh - March 17, 1985

I have recently had to ‘tidy up’ a friends Model III discs which brought me to a documentation study of the PDRIVE functions of NEWDOS80 as I found myself faced with the problem of compatibility between TRSDOS, DOSPLUS, LDOS and (my favourite) NEWDOS80 V.2. This document examines SOME of the methods NEWDOS80 V.2 uses to overcome these problems with its library command ‘PDRIVE’.

The reason we can’t read DOSPLUS and LDOS double density discs is that they have strayed from the norm. The ‘norm’ in the computer industry appears to be whatever the programmer/designer decides that it will be at the time of their current project. It would take approximately 5K of program to allow NEWDOS80 to read their type of double density directory. Whilst either one will read between a double or single density Model I or Model III of their own type, they do not have the ability to read one of the others. Therefore if you have some DOSPLUS or LDOS files and wish them on NEWDOS80, copy them to single density discs and then set up the PDRIVE table as example #1 in the following table. This will allow NEWDOS80 to read and copy them onto a NEWDOS80 system or formatted data disc. One thing that you may not be aware of at first is that you are not limited to two letters in the ‘TI’ spec of PDRIVE. The manual states:

“TI = etc., Where TI consists of one or more alphabetic letter flags chosen from, etc.”

This means one or MORE alphabetic letter flags can be used in the ‘TI’ spec. Even though all the examples given therein consist of one or two - you can have three if so required. Example 6 in the manual displays:

PDRIVE,dn1,dn2,TI=AK,TD=E,TC=39,SPT=18,TSR=3,GPL=2,DDSL=17, DDGA=2

This specification is for the Model III for 13.3cm disc, 40 tracks, double density, single-sided disc that has track zero formatted in single density. Now by just altering the TI=AK to TI=AL, you have the same specification except that you can read that same 40 track disc in an 80 track drive.

Some examples of PDRIVE tables:

Example #1.

```
PDRIVE,0,9,TI=A,TD=A,TC=40,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
```

This spec allows you to read LDOS single density discs as is, and DOSPLUS single density discs (after WRDIRP) that are 40 tracks.

Example #2.

```
PDRIVE,0,8,TI=AM,TD=E,TC=40,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA=2
```

This spec will allow you to copy any known file from a Model III TRSDOS disc. You still will not be able to read the directory, as the TRSDOS directory is not compatible with any other marketed system. You can read the TRSDOS directory in one of two manners:

Boot up the TRSDOS disc and call up the directory, or better still, call up ‘SUPERZAP’ go into ‘DD’ that contains the TRSDOS disc and (display drive relative sector 306), you can even MOD sectors. This will display the directory and other sectors whilst in NEWDOS80 using the above PDRIVE setup.

Example #3.

```
PDRIVE,0,7,TI=AK,TD=E,TC=39,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
```

This spec tells your Model III that you are looking at a Model I NEWDOS80 disc that is double density with track zero in single density.

Example #4.

```
PDRIVE,0,6,TI=AL,TD=E,TC=40,SPT=18,TRS=3,GPL=2,DDSL=17,DDGA=2
```

This is the standard Model III setup except that you can read the 40 track disc in an 80 track drive.

Example #5.

```
PDRIVE,0,5,TI=AL,TD=A,TC=40,SPT=10,TSR=3,GPL=2,DDSL=17,DDGA=2
```

This spec is the same as example #1 (LDOS read) except that you can read it in an 80 track drive.

Example #6.

```
PDRIVE,0,4,TI=ALM,TD=E,TC=40,SPT=18,TSR=3,GPL=6,DDSL=17,DDGA= 2
```

This one is the same as example #2 (TRSDOS III) except that the disk is read in an 80 track drive.

Example #7.

```
PDRIVE,0,3,TI=A,TD=E,TC=80,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
```

This is the standard setup for an 80 track drive on the Model III.

Example #8.

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

This is the 40 track standard setup.

One small pointer. If you have two drives on your system, set up the 'SYSTEM' options command of 'AL=2•. This will give you use of the other seven PDRIVE tables as a storage area for the tables that you are likely to be using at different times. If you have three drives, set 'AL=3•. When you need a different drive option, you only type "PDRIVE,0 1=7• and presto, you are ready. By resetting the 'AL' function, you not only gain the extra storage space, but your system need not go looking for a file on a drive that doesn't exist. This actually increases your disk I/O time - sure not by much, but every nanosecond counts.

Setting HIMEM from NEWDOS/80 v2.0 BASIC

Regardless of what the documentation says, high memory cannot be set using a **CMD "HIMEM,(decimal address)"** command. The string storage area is still going to the top of 48k, even though you may have set HIMEM at 34000.

The only way to set high memory from BASIC is to place a poke at the beginning of the program - before any string or variable definitions are made. To calculate the poke values needed, find the beginning address of the routine. Divide this number by 256. The most significant (MSB) byte of the address will be the integer value of your answer. Multiply the decimal portion of the answer by 256 to find the least significant byte (LSB). On a Model I or III:

```
POKE 16561,LSB : POKE 16562,MSB: CLEAR 50
```

This will reset all pointers, and set memory to address calculated. The CLEAR 50 can be what ever value you need for your program. It is not necessary - and wasteful to clear more memory than needed.

Note - String garbage collection will be done at fewer intervals when the maximum amount of memory has been cleared.

Model III Self-Booting Disk using NEWDOS/80

- by Leonard Yates (with help from Gary Bryce)

Back in the August 1984 issue of SYDTRUG NEWS, Gary Bryce presented an article on creating self-booting disks using NEWDOS/80. The Model I mods worked well but,as he said at the end of the item, had not been verified on the Model III. Well, to get a Model III data disk to self-boot, you'll have to proceed as follow (I'll make this complete so you don't have to refer to the original article):

1. FORMAT a data disk and copy the /CMD file to be made self-booting to it. If it's the only file on the disk (apart from BOOT/SYS and DIR/SYS), chances are it will have only one extent (as revealed by DIR dn1 A). If

it has more than one extent, it’s probably more than 32 granules long (Super Utility 3.2 is the longest program I’ve been able to self-boot).

2. Using the DFS option of SUPERZAP, note the DRS of relative sector 0 of the /CMD file on the data disk. Next, using the DD option of SUPERZAP, and responding with the drive number and DRS noted above, record the TRS and TRK (in HEX) of the /CMD file on the data disk.
3. Using the DFS option again, zap FRS 1 of BOOT/SYS (of the data disk) as follows:

Byte 05 - TRS of /CMD file (will be 05H if it’s the only USR file on the disk, and hence won’t need to be changed).
Byte 06 - TRK of /CMD file (will be 00H…etc.).
Byte 3E - change from C8H to C9H.

Note: If the /CMD file is the only USR file on the data disk, TRS will be 05H and TRK will be 00H. BOOT/SYS is normally set up to load SYS0/SYS which always occupies from sector 05 of track 0; for a nonsystem (data) disk, the first USR file will occupy this space. Note also that on NEWDOS/80 system and data disks for the Model III, sector 1 of BOOT/SYS is a duplicate of sector 0 (required for booting on the Model III) These changes are made to sector 1 only. Further, if your /CMD program is the only USR file on the data disk, the only zap needed is to byte 3E of FRS 1 of BOOT/SYS.

4. Mount the self-booting data disk on drive 0 and press RESET. The disk will BOOT and the /CMD file will load and execute.

Reading a Model 4 TRSDOS 6.0 or a Model III LDOS disk on NEWDOS/80

To read a Model 4 TRSDOS 6.0 or a Model III LDOS disk directly on NEWDOS/80 V.2, use this PDRIVE setting:

```
TI=A, TD=E, TC=40, SPT=18, TSR=3, GPL=6, DSL=12, DGA=2
```

Trading Files Between LDOS6.2 and NEWDOS/80 v2.0 - Chris Anderson

One of the things I quickly discovered as I began working with my new Model 4P was that there was going to be a need to move data back and forth between my old NEWDOS disks and LDOS6 disks. The following procedure is very straightforward, and will work without hassle for those important data files.

Note: The following procedure requires at least two drives!

NEWDOS80 V2 to LDOS6.2

1) Boot under LDOS6.2.
2) Format a disk SSSD, 35 tracks. You MUST use single side, single density and 35 tracks, regardless of the type of drive you are actually using!
3) Boot NEWDOS80 V2. (Note, if you are a 4P owner, and are having problems booting NEWDOS, see the info file on this topic)
4) Set the PDRIVE for the drive where the LDOS disk resides to TI=A,TD=A,TC=35,SPT=10,GPL=2,DDSL=17,DDGA=2.
5) Use the normal NEWDOS Copy command to copy the file from your NEWDOS80 disk to the LDOS disk.
6) Reboot the system using LDOS6.2.
7) Use the RESET command or TSK/CMD if you have several copied files) to CLOSE each of the copied files. NEWDOS is not familiar with the open/closed status, and will leave each file open when copying it over to LDOS.

That’s all there is to the job in that direction. Going the other way is just as easy.

LDOS6.2 to NEWDOS80 V2

Perform the steps above in the following order:

1) Boot in 6.2
2) Format as noted above. Then, copy files as desired to the newly formatted disk while still in LDOS.
3) Boot NEWDOS.
4) Set PDRIVE as noted above.
5) Use normal NEWDOS copy command to move files from the LDOS disk to the NEWDOS disk.

Provided you have a Model 4 (and not a 4P) you may also find that it is fairly fast to use Multidos for this sort of thing.

Modifying the AMPERSAND Command in NEWDOS/80 BASIC - Gil Spencer

I never think in octal. It’s hard enough to work in binary, hex, and decimal. It always seemed to me that the default for the ‘&’ function should be hex, NOT octal. I finally dug out the source code (from Apparat’s Disk BASIC) which I found in SYS20/SYS. My rewrite fits within the required extra space. Although a quantity of bytes are changed, this is because the code is “re-arranged” more than because it is “re-written”.

First, here is the Disk BASIC (&) routine found in Apparat’s NEWDOS/80 version 2.0 - specifically SYS20/SYS, addresses 54C5H-5503H. If you are using SUPERZAP, address 54C5H is found at FRS 2, byte D1H and address 5503H is at FRS 3, byte 13H. Note that the four bytes at FRS 3, bytes 06H- 09H (which are 01 00 FA 54) are loader codes and must NOT be changed.

```
54C5          00100         ORG     54C5H
54C5 D7       00110         RST     10H
54C6 4F       00120         LD      C,A
54C7 110000   00130         LD      DE,0000H
54CA 79       00140 Q54CAH  LD      A,C
54CB FE48     00150         CP      48H
54CD 2022     00160         JR      NZ,Q54F1H
54CF D7       00170         RST     10H
54D0 EB       00180         EX      DE,HL
54D1 D630     00190         SUB     30H
54D3 FE0A     00200         CP      0AH
54D5 3808     00210         JR      C,Q54DFH
54D7 D611     00220         SUB     11H
54D9 FE06     00230         CP      06H
54DB 3022     00240         JR      NC,Q54FFH
54DD C60A     00250         ADD     A,0AH
54DF 29       00260 Q54DFH  ADD     HL,HL
54E0 3807     00270         JR      C,Q54E9H
54E2 29       00280 Q54E2H  ADD     HL,HL
54E3 3804     00290         JR      C,Q54E9H
54E5 29       00300         ADD     HL,HL
54E6 3801     00310         JR      C,Q54E9H
54E8 29       00320         ADD     HL,HL
54E9 DAB207   00330 Q54E9H  JP      C,07B2H
54EC 85       00340         ADD     A,L
54ED 6F       00350         LD      L,A
54EE EB       00360         EX      DE,HL
54EF 18D9     00370         JR      Q54CAH
54F1 0E4F     00380 Q54F1H  LD      C,4FH
54F3 B9       00390         CP      C
54F4 2801     00400         JR      Z,Q54F7H
54F6 2B       00410         DEC     HL
54F7 D7       00420 Q54F7H  RST     10H
54F8 EB       00430         EX      DE,HL
54F9 D630     00440         SUB     30H
54FB FE08     00450         CP      08H
```

```
54FD 38E3      00460         JR      C,Q54E2H
54FF CD9A0A    00470 Q54FFH  CALL    0A9AH
5502 EB        00480         EX      DE,HL
5503 C9        00490         RET
```

This is the REWRITE of the Disk BASIC ampersand (&) routine. Now the octal argument MUST be specified by ‘&O’. Hex argument MAY be specified by ‘&H’. No suffix (i.e. ‘&’) now defaults to hex rather than octal.

```
54C5           00100         ORG     54C5H
54C5 D7        00110         RST     10H
54C6 4F        00120         LD      C,A
54C7 110000    00130         LD      DE,0000H
54CA 79        00140 Q54CAH  LD      A,C
54CB FE4F      00150         CP      4FH
54CD 2828      00160         JR      Z,Q54F7H
54CF 0E48      00170         LD      C,48H
54D1 B9        00180         CP      C
54D2 2801      00190         JR      Z,Q54D5H
54D4 2B        00200         DEC     HL
54D5 D7        00210 Q54D5H  RST     10H
54D6 EB        00220         EX      DE,HL
54D7 D630      00230         SUB     30H
54D9 FE0A      00240         CP      0AH
54DB 3808      00250         JR      C,Q54E5H
54DD D611      00260         SUB     11H
54DF FE06      00270         CP      06H
54E1 301C      00280         JR      NC,Q54FFH
54E3 C60A      00290         ADD     A,0AH
54E5 29        00300 Q54E5H  ADD     HL,HL
54E6 3807      00310         JR      C,Q54EFH
54E8 29        00320 Q54E8H  ADD     HL,HL
54E9 3804      00330         JR      C,Q54EFH
54EB 29        00340         ADD     HL,HL
54EC 3801      00350         JR      C,Q54EFH
54EE 29        00360         ADD     HL,HL
54EF DAB207    00370 Q54EFH  JP      C,07B2H
54F2 85        00380         ADD     A,L
54F3 6F        00390         LD      L,A
54F4 EB        00400         EX      DE,HL
54F5 18D3      00410         JR      Q54CAH
54F7 D7        00420 Q54F7H  RST     10H
54F8 EB        00430         EX      DE,HL
54F9 D630      00440         SUB     30H
54FB FE08      00450         CP      08H
54FD 38E9      00460         JR      C,Q54E8H
54FF CD9A0A    00470 Q54FFH  CALL    0A9AH
5502 EB        00480         EX      DE,HL
5503 C9        00490         RET
```

[link] [TB]

Comments are closed.

[Ira Goldklang’s TRS-80 Revived Site is proudly powered by **WordPress**.]

Site Navigation

General

- Site News
- Main Page

- TRS-80 Model Details
- TRS-80 DOS Details
- Zaps/Mods/Pokes/Tricks
- Adventures
- TRS-80 Archive Search
- Using an Emulator
- Request Real DOS Disks
- Click to Donate

Emulation/Conversion

- Emulators
- ORCH-80
- Media Conversion Utils
- Tape Conversion Steps
- Disk Conversion Steps
- DMK Format
- Click to Donate

Reference

- Level II BASIC Ref
- NEWDOS/80 Ref
- Z-80 Quick Guide
- 80 Microcomputing
- TRS-80 Wiki
- TRS-80's on EBAY
- Click to Donate

Covers

- **Books & Manuals**
- Books
- Comic Books
- Hardware Manuals
- **Commercial/Product**
- Radio Shack Catalogs
- Vendor Catalogs
- Product Flyers
- Company Newsletters
- **Newsletters**
- User's Groups
- **Magazines**
- 80 Microcomputing
- 80 Notebook
- 80 U.S. Journal
- Baron's Microcomputing
- CLOAD
- Computer User
- Creative Computing

- H&E Computronics
- Microcomputer News
- Micro-80
- Softside
- The 8ighty
- T.A.S.
- Trace
- TRSTimes
- Other Magazines
- Click to Donate

Communication

- Send Me Email
- Email and Links
- Guestbook
- Clubs/Forums/Boards
- Personal Stories
- Copyrights
- Click to Donate

Site Visitors

Site Visitors

- **Translator**

By N2H

- Search for: Search

- **Categories**

  - Books (7)
  - Catalogs (5)
  - Companies (19)
    - Acorn (1)
    - Alpha Products (1)
    - Big Five (2)
    - Breeze/QSD (1)
    - Broderbund (1)
    - Med Systems (1)
    - Racet Computes (1)
    - Radio Shack (11)
    - Strategic Simulations Inc. (1)
  - Computers (8)
    - CoCo 3 (2)
    - CP-500 (1)
    - HT1080Z (3)
    - MC-10 (1)
    - Model 100 (1)
    - Model 4 (4)
    - Model 4P (1)
    - Model I (4)

    - Model III (4)
- Contributors (41)
    - David Cooper (17)
    - Horacio (6)
    - Knut Roll-Lund (4)
    - Matthew Reed (5)
    - Mendel Cooper (1)
    - Miguel Dutra (1)
    - Phil Ereaut (5)
    - Tim Mann (1)
    - Vernon B Hester (4)
- Downloads (22)
    - Articles (2)
    - Cassette Utilities (3)
    - CP/M (1)
    - DOSes (2)
    - DSK and DMK Utilities (6)
    - Emulators (7)
    - Windows Utilities (8)
- For Sale or Trade (10)
- Hardware (6)
    - Line Filter (1)
    - NewClock/80 (1)
    - ORCHESTRA 80/85/90 (1)
    - Printers (2)
    - RS-232 (1)
    - Telephone Interface II (1)
- Magazine (64)
    - 80 Microcomputing (8)
    - Computer User (4)
    - Micro-80 (1)
    - PROG-80 (1)
    - Radio Shack Computer Center Answers (5)
    - Softside (30)
    - The Alternate Source (15)
    - TRSTimes (1)
- Manuals (33)
    - Hardware Manual (17)
    - Software Manual (19)
- Site Updates (39)
- Software (13)
    - Astrocal (2)
    - Euclid Geometry Tutor (1)
    - Fasteroids (1)
    - Galactic Revolution (1)
    - Infinite Basic (1)
    - Keys of Acheron (1)
    - Maces and Magic (1)
    - MicroDOS (1)
    - MultiDOS (1)
    - Renum (1)
    - Sargon II (2)
    - SFinks (1)
    - Sky Warrior (1)
    - Starfighter (1)
    - Super Step Z80 Processor Model (1)
    - SuperUtility (2)

- Weerd (1)
- Zeus (2)