MICROPOLIS USERS GROUP

MUG Newsletter # 18 - January 1982

****

MICROPOLIS S/W FROM A SINGLE SOURCE

I have become, or am in the process of becoming, a dealer for many of the vendors who produce S/W for Micropolis. I, along with Lynn, Tam, and Brad, will be able to answer questions on the operation of the various packages, provide comparison information between similar packages, and, in general, provide a knowledgable, accessible source for Micropolis S/W. Support after the sale is certainly also included.

The details on the amount of S/W available, and methods of providing information to a potential buyer (e.g., a catalog) are still in the development stage at this point. However, if you are considering a purchase of S/W (MDOS or CP/M) created by any of the following vendors, I'd appreciate you giving us a call.

SYSTEMATION
BONJOEL
DATASMITH
GMS S/W
MONK S/W
ORGANIC SOFTWARE
COCKERHAM & ASSOC S/W
LENZ S/W
AFB MICRO CONTROLS
UNITED CONTROLS CORP

We can be reached at our new business phone, (205) 881-1697, or at our old personal number, (205) 883-2621.

....

THE BASIC TOKEN TABLE

by Burks A. Smith of DATASMITH
Box 8036, Shawnee Mission KS 66208

Commands and functions in Micropolis BASIC are stored in memory and on disk as single-byte "tokens" that are codes for the operation specified. Not only does this save time when the program is being interpreted by BASIC, but it also saves space in memory.

If you "dump" the memory image of a BASIC program, you will see that the program is stored in ASCII character codes with the exception of the BASIC commands, functions, and even information in quotes that contains imbedded command and function names. BASIC examines everything entered, and if it finds it in its table, it converts the text to a single-byte token for internal use. Each line in the program is terminated with a FD hex character, and the end of the program is designated by an FE hex character.

Knowing the internal storage format and the command token table will allow you to access BASIC program files and treat them as data for utility programs. This is exactly what DATASMITH's utility programs for removing unnecessary spaces, converting to text, and merging several files do.

Since the Basic commands are not stored as English language text like they appear when the program is LISTed, some interesting possibilities arise. One that occurred to me was that it wouldn't be much trouble to produce a foreign language version of Micropolis Basic simply by creating a token table with foreign language words in it. This would make it easier for people who don't speak English to learn BASIC, but wouldn't produce any incompatibility in the programs they write. The language that would appear when the LIST command was entered would depend only on the token table BASIC was using at the time. Too bad I don't speak any foreign languages well enough to play with this one.

A somewhat easier application is to create a version of BASIC that has a secret "LIST" command. This would prevent any unauthorized person from listing a program. To do this, all you have to do is to find the word LIST in the table (3086H in vs. 4) and change it to a different 4-letter word. LIST will then cause a syntax error and your secret word will cause a list.

Listed below is the BASIC token table as it appears in Version 4.0 BASIC. The table starts at 30A0H in memory and consists simply of the Basic keywords followed by a hex token equivalent. It is interesting to note that an undocumented operator (/=) appears in the table. Although it has a unique token, it appears to operate as "not equal." It appears that the order of the table is important, since some tokens begin with the same letters. For instance, INPUT must appear before IN to avoid converting it to IN and PUT. The table as printed here is read from left to right, top to bottom.

KEYWORD	BYTE	KEYWORD	BYTE
DEF	82	DIM	83
ENDPAGE	85	ELSE	86
END	87	EXEC	88
FOR	89	LIST	8A
GOSUB	8B	GOTO	8C
IF	8D	INPUT	8E
LET	8F	MEMEND	90
NEXT	91	ON	92
PRINT	93	READ	95
REM	96	RESTORE	97
RETURN	98	RUN	99
SIZES	9A	STEP	9B
STOP	9C	STRING	9D
THEN	9E	TO	9F
GETSEEK	A0	PUTSEEK	A1
GET	A2	PUT	A3
OPEN	A4	CLOSE	A5
TAB	A6	FLOW	A7
NOFLOW	A8	POKE	A9
OUT	AA	DELETE	AB
RENAME	AC	ATTRS	AD
FREESPACE	AE	EOF	AF
SAVE	B0	SCRATCH	B1
RECORD	B3	PAGESIZE	B4
ASSIGN	B5	LINK	B6
CLEAR	B7	ERROR	B8
LOAD	B9	CHAIN	BA
PLOADG	BB	DISPLAY	BC
OR	BD	AND	BE
<>	BF	=>	C0
=<	C1	>=	C2
<=	C3	>	C4
<	C5	/=	C6
NOT	C7	=	C8
+	C9	-	CA
\	CB	*	CC
/	CD	^	CE
CHAR$	CF	ASC	D0
LEFT$	D1	RIGHT$	D2
LEN	D3	STR$	D4
VAL	D5	REPEAT$	D6
INDEX	D7	MID$	D9
VERIFY	DA	INT	DB
FIX	DC	SGN	DD
FRAC	DE	RND	DF
ABS	E0	SQR	E1
LN	E2	LOG	E3
EXP	E4	SIN	D5
COS	E6	TAN	E7
ATN	E8	PEEK	E9
MOD	EA	IN	EB
FMT	EC	MIN	EE
MAX	EF	NAME	F0
ATTR	F1	TRACKS	F2
FREETR	F3	SIZE	F4
RECGET	F5	RECPUT	F6
ERR$	F7	ERR	F8
SPACELEFT	F9	PGMSIZE	FA

....

THE MONTH IN REVIEW

I've been plagued with computer problems this month. Two pieces of 32K RAM decided to quit working, a keyboard went out on one of my SOLs, a voltage regulator blew on one of the VG drives, and a VG terminal has decided to randomly spread and compress its display. I also think I have problems with the CPU board on the CCS. Great fun, especially when one is as incompetent in hardware as I am.

COMMUNICATIONS

On the bright side, I have communication terminals running both at home (on a SOL) and at work (on a VG). I bought the Prentice acoustic modems (use to be Livermore) for $129 from Priority One during their recent sale. They are running under CP/M, not MDOS. Testing Bob Barnum's (Newsletter 7) MDOS modem program is on my to-do list, but a friend set these up with public domain CP/M software. I'm using MODEM7 for initial control and XMODEM for transfering files. I haven't personally made much use of either capability. The office system is used daily. though, on one of the NASA contracts. Through the NASA netwoking system, we can talk to installations throughout the U.S., and can utilize the computing power of VAX 11/780's. The terminal is a definite timesaver for the young lady who is Network Manager of this NASA system. She now performs a lot of her work without having to physically make the trip to the NASA installation.

RIGID DISK

A second bright spot is the installation of a 35 megabyte Micropolis rigid disk at the office. Again, however, it is running under CP/M, not MDOS or OSM. I tried to get OSM to work, but couldn't. Lifeboat sells a $125 expansion program that interfaces the rigid disk to the system. To configure it is simple; answer two questions on where your controller is addressed and where you want the extra code put. It does cut your memory by about 8k. To the user, it looks like you have five 6-megabyte disks specified as A, B, C, D, & E. The floppies are still on line as drives M and N. It's a nice system - considerably faster and obviously offering a great deal of file space. I've had no problems, though some people think the drive is a bit loud. I don't; It's music to my ears. It's just a "whirr", by the way. You don't hear any "clack-clack" of heads moving.

During a recent trip to the west coast, I stopped by and talked to Micropolis about getting an extension of MDOS to run on the rigid disk. I figured they had that running first in the development of OSM, which is a multi-user, multi-tracking system. They didn't develop in that way, sorry to say.

There is a way we could get such a piece of software. Steve Zook, the developer of MDOS, and a prime contributor to OSM, is now working as an independent consultant. He certainly has the knowledge required to write a MDOS "lookalike" for the rigid disk. As Steve explained to me, it really can't be an extension; it has to be a totally new piece of software. It has something to do with the file, track and directory sizes being greatly expanded from MDOS. So, what I'd like to know is: How many members have a rigid disk? Who would be interested, now or later, in the availability of such an operating system? Would you pay $200 for it?

MUG CONTROL OF MDOS & MpBASIC

I also discussed the possibility of the MUG having the source code to the current Version 4 system. I proposed that we could serve the user better than Micropolis does. Not better than they could, but better than they do. Micropolis doesn't feel that there is a need for such a move.

They believe that there isn't any need for support of MDOS. All the bugs are out. Not very many new systems are being sold. They get very few requests for support, therefore there isn't any need for the MUG to have the code.

Their basic premise is correct. The system is solid, with few, if any bugs. From a manufacturer's viewpoint, support isn't needed.

From a USER'S viewpoint, lots of support is needed. Micropolis' responsibility ended with configuration to hardware and purging of software bugs. If it runs, their job is done. How to use it? Micropolis doesn't feel that is their problem. Fine - but it is the MUG's problem. The Micropolis documentation is useful to an experienced programmer, but not to a first time user. In addition, expansion of utilities, development of new utilities or application programs, require knowledge of how the system develops and manipulates internal tables. Possible development of a Version 5, or a rigid disk MDOS depends on the lawful use of Version 4.

Micropolis doesn't object to our finding and documenting of this information. For the time being, then, we'll all keep trying to document what we know about the system. I'll keep trying to get the source.

....

MICROPOLIS PREVENTIVE MAINTENANCE

The following is a table of preventive maintenance activities which should be performed on Micropolis drives. Parts and materials can be ordered from Tim Matthews of Micropolis, at 21329 Nordhoff St., Chatsworth CA 91311, phone (213) 709-3300.

Maintenance	Frequency	Time Required Hours
Replace Head Load Pad(s)	2000 hrs of disk access	0.1
Clean Heads(s)	2000 hrs of disk access	0.1
Replace Drive Motor(s)	5000 hrs of motor operation	0.5
Lubricate Lead Screw(s)	2000 hrs of disk access	0.1
Lubricate Latch Mechanism	Every 2 years or when latch becomes irratic	0.1

CLEANING AND LUBRICATION

Cleaning the head

CAUTION

ROUGH OR ABRASIVE CLOTH SHOULD NOT BE USED TO CLEAN THE MAGNETIC RECORDING HEAD. USE ONLY ISOPROPYL ALCOHOL OR DUPONT FREON TF. USE OF OTHER CLEANING SOLVENTS, SUCH AS CARBON TETRACHLORIDE, MAY DAMAGE THE HEAD LAMINATION ADHESIVE.

Clean the magnetic head, using a lint-free cloth or cotton swab moistened with isopropyl alcohol or DuPont Freon TF. Wipe the head carefully to remove all accumulated oxide and dirt. Dry the head using a lint-free cloth.

NOTE

The magnetic head must be cleaned
after head load replacement.

Lubrication of the Stepper Motor Lead Screw

Prior to lubrication, the stepper motor shaft (lead screw) should be cleaned. Wipe the shaft with a lint-free cloth lightly moistened with isopropyl alcohol. Lubricate the lead screw and portion of the carriage that rides on the platen by applying a liberal portion of Saunders Magnalube Micropolis P/N 732-0001-6 to these surfaces.

CAUTION

DO NOT CONTAMINATE MAGNETIC RECORDING HEAD OR HEAD PAD WITH LUBRICANT. DAMAGE TO THE RECORDING SURFACE CAN BE CAUSED BY LUBRICANT DEPOSITED ON THE MAGNETIC HEAD OR LOAD PAD BEING TRANSFERRED TO THE DISK.

Lubrication of the Latch Mechanism

To ensure smooth positive action of the door latch mechanism, apply a heavy coat of Saunders Magnalube Micropolis P/N 732-0001-6 to the entire latch mechanism.

....

NEW LIBRARY DISK SOFTWARE

Lots of new, good software has come to the MUG. The following text explains the function of some of it.

THE GAME OF LIFE

by David McGraw

I thought that some of the members might find my version of LIFE interesting. I have done just about everything possible to make it useful for a group with radically different hardware configurations.

If the terminal for which this program is being configured has curser controls of different value, it will be quite easy to modify the program. If the terminal doesn't have such controls, it will be necessary to write a different routine for entering the initial universe patterns.

This program is based on the article "LIFE CAN BE EASY" published in the April 1979 BYTE by RANDY SODERSTROM. The sixteen bit math and easy loader suggested in the article are implemented in my version as well as adjustable universe boundries and population-generation printout. Hopefully, the comments I've included in the source files will clear any ambiguity about the code. Never-the-less, if there is still any confusion feel free to let me know and I will try to clear it up.

One of my difficulties in preparing this program for the Users Group stems from the fact that I don't know what sort of configuration is best for the user's terminal. My terminal is a Compucolor II which I use in the CRT mode. This means that I use a plethora of control codes for various terminal commands that few other terminals would interpret in the same manner. Furthermore, since this is the only terminal with which I am familiar, it would be most helpful if the MUG would publish some general guidelines for programmers like myself who are preparing programs for the Users Group.

LIFE is available on MUG Library Disk 11.

....

RESTORE SCRATCHED FILES

by Carl J. Singer

RESTORE is a utility program designed to retrieve a scratched file. To invoke the program from the MDOS executive. type RESTORE followed by the scratched filename enclosed in double quotes.

```
[unit:]RESTORE "<[unit:]filename>"
```

RESTORE exits to the MDOS operating system when it is done or when it encounters an error condition.

If a file has been inadvertently (or otherwise) scratched, RESTORE will retrieve it with 100% success, provided there has been no intervening save of any kind. (Intervening scratches do not matter.) In the unlikely event of another restorable scratched file with a name identical (except for the first letter) to the name of the desired file - and preceding it in the directory - you will get back the wrong file with the searched-for name. Rename it, scratch it, and reinvoke RESTORE.

If there has been an intervening save, the possibilities abound - most of them bad. RESTORE may get your file back O.K., or it may tell you the file has been restored because all of the old tracks have been retrieved in order. However, an intervening save may have used some of your file's tracks, and subsequently have been itself scratched, returning your file's original tracks (completely altered) to free status.

If RESTORE does not get back the scratched file, it will exit with one of two messages:

(1) Scratched filename not in directory
(2) Scratched file overwritten

In the first case, the scratched filename has been replaced in the directory. One or more of its tracks MAY have been overwritten. In the second case, the old filename is still there, but at least one of its tracks has been taken.

As a consolation prize in both of these situations, there will be printed on the console a list of the only possible tracks where your old file, or its remnants, may reside. You can then use my routine GET-TRAX to bring these tracks into memory for examination and possible consolidation following the lines suggested in GET-TRAX.D.

The running time of RESTORE is rather long where a multitrack file is being retrieved (up to 23 seconds on my 2 mHz 8080) because of the large number of disk references required to assure that the correct tracks in the correct order are being picked up. Single track restorations run in about 1/3 of the time.

Consecutive uses of RESTORE may make use of the APP command, saving the time needed to reload RESTORE.

```
APP "<[unit:]filename>"
```

RESTORE is available on MUG Library Disk 09.

....

GET-TRAX

by Carl J. Singer

This program retrieves any series of consecutive tracks (e.g., a file) from the disk without recourse to the directory.

The drive number, starting track number, and the number of tracks are entered and the file is dumped into memory starting at 3000H. If "full sector" is specified, each sector starts at a 512-byte boundary and is 266 bytes long.

If "data only" is specified, each sector is 256 bytes long and the data file is dumped to memory without gaps.

GET-TRAX also retrieves sector leaders separately without the accompanying data file. The term "ten-byte leader" refers to the ten bytes at the beginning of each sector immediately following the track and sector identifiers and preceding the data.

If "track leaders only" are specified, each 16-byte line (starting at 3000H) will have the track number followed by the ten-byte leader of the first sector of each track.

If "all sector leaders" are specified, each 16-byte line will have the track and sector numbers followed by the ten-byte sector leader.

Although GET-TRAX is a general-purpose utility for the inspection of disk tracks, it also enables one to retrieve any scratched file WHICH WAS NOT SCATTER-LOADED and which has not been overwritten, regardless of its track layout on the disk.

An example will make the method clear:

Suppose the file VERSENKT has been inadvertently scratched, and you want it back. First call GET-TRAX, using code D (data only) and specify track 0 (the directory) and 1 track to be retrieved.

Now DUMP from 3200 on until the scratched file appears. If you are using my ASCII dump modification, it will appear as .ERSENKT . If you are using the original Micropolis dump, look for the sequence FF 45 52 53 45 4E 4B 54 20 20.

Following the filename will be the starting track number (with the 7-bit set high), the file type, a binary zero, and (the number of sectors+1). All of these numbers are in hexadecimal. Suppose in this case that these numbers are B4 06 00 35. We now know that the file starts on track 52 (34H), that it is a Type 6 file, and that it uses three full tracks plus four sectors of a fourth track.

Now get the track map by dumping 3100 to 314F. The free tracks are listed in order starting at 3100 and ending at the byte which immediately precedes 80 (the directory, or track 0). Copy them down (or print them out), for if the file tracks that we are about to find are not all on this list, the complete file cannot be resurrected, since some of it will have been written over. This unfortunate situation can occur only if something has been saved on the disk since the inadvertent SCRATCH.

In our hypothetical case, the track map shows the following sequence:
30 31 32 34 3E 3F 46 47 4A 4B 4C 80 .....

Invoke GET-TRAX again (entering APP will do it) and use code L. This will produce a list (starting at 3000H) of the Sector 1 leaders of each of the 77 tracks. The first byte in each row of the dump is the track number, and the second byte is the file track which preceded this track (unless the 7-bit is set high, in which case it is the first track of the file.)

We know that VERSENKT started at track 34H, so we look for 34 in column 2. We find the entry 46 34, so we know the second track of the file was 46H. Similarly. we find entries 47 46 and 3F 47. Now we know the exact track sequence of the file: 34 46 47 3F. In decimal, the tracks are 52, 70, 71, 63.

Now check these tracks against the list from the track map. Fortunately, they are all there among the free tracks.

We know the last track was 3F (63) and that four sectors of this track were used, but we don't know the exact number of bytes stored on this track. So we say APP again, calling for code S, track 63, 1 track. This gives us the 16 sector leaders of Track 63. When we DUMP 3000 30FF, we look at the 11th and 12th bytes on each line, which tell us (in low-high hex) the number of bytes of each sector that were used. We find the first three sectors completely filled (bytes are 00 01) and the fourth sector has 5A(hex) bytes (bytes are 5A 00). Thus the last data byte of the file was in position 0359H on Track 3F (63).

The rest is easy. We call GET-TRAX (or APP) again, requesting code D, track 63, 1 track, then calling MOVE 3000 3FFF 6000. Next APP, code D, track 70, 2 tracks, then calling MOVE 3000 4FFF 4000. Finally, another APP, code D, track 52, 1 track. Now the complete original file is in correct sequence in memory, starting at 3000H and ending at 6359H.

We finish with SAVE "VERSENKT" 3000 6359 6, and make a mental note not to get in a jam like this again.

Again it must be stated that this technique will NOT work on a file which has been scatter-loaded. Type 0-7 and 10-13 files are never scatter-loaded, but the others may be. One can check to see whether a suspected file has been scatter-loaded by examining the sector leaders. Using code S, get the leaders of each sector of each of the tracks in the file. Look at the 11th and 12th bytes of each leader displayed. If, for every sector except the last, these bytes read 00 01, the file was not scatter-loaded and the above technique may be used.

GET-TRAX is available on MUG Library Disk 9.

....

MICROPOLIS UTILITIES

by William A. Miller II of BONJOEL ENTERPRISES
P.O. Box 2180 DesPlaines, ILL. 60018

In response to NEED FOR UTILITIES in the #12 issue:

CONTROL P

The CONTROL P option in CP/M is a part of the op-system and is not a utility in the sense that it is loaded as a unique program. It could be implemented in the RES as follows:

Insert a trap in the console input routine to catch the CONTROL P. When found in the input you would toggle a byte in memory. This byte is used to determine whether or not to call the printer output routine from the console output routine. (I have implemented it in the enclosed listing as EDIT-P)

SCREEN EDITOR

The screen oriented editor would be a true utility and there are some editors available (mainly for CP/M) that do most of what you want.

SCREEN DUMP

A screen dump would not be implemented as a utility but as a built in function due to the requirement that anything put on the screen must be saved in memory or on disk (heaven forbid) in a constant scrolling fashion so that when the operator requests a screen dump the information is available. Since an 80 by 24 screen format would require up to 1920 bytes of buffer per page, along with the room required for the program to manage the buffer, this function is generally reserved for memory mapped CRTs which, by their very nature, require the buffer and managing program to already be resident.

DOUBLING YOUR KEYS

As a side note to those of you who never seem to have enough keys, one trick that I implemented on my system was to use the parity line connected to a separate key called EDIT. This gives me the standard 128 (0-7f) keys times two for a grand total of 256 key stroke combinations. This method works due to the fact I decode the parity line in my section of the RES console routines and RES strips the parity bit in any case. The extra keys are used for a variety of uses such as direct control of my CRT (programable VDB-8024) and my most used function which is to control a software-variable speed scroll for my CRT. Enclosed is a listing of my RES overlay. It may help give you some ideas for the future.

William's RES is available on MUG Library Disk 7.

....

PASSWORDS

by Dave Risley

Implements security into your system

This routine issues, upon booting your system, a request for a security code and gives you two tries to enter it correctly. After the second try, the system issues a Halt command. The security code is not echoed to the terminal and is a three character alpha or numeric code, which can be expanded to any length. The code will reside within the "res module" without disturbing the existing coding of BASIC Vs 4.0. If more elaborate prompting messages are needed they may be inserted in the text definition line, but you would need to redefine the origin of the program for a scatter load to set it in the various padded portions of the res module. If you assemble my source code as is, the program will issue a home and clear screen command, prompt with "enter ID". The current ID is "RPD".

PASS is available on MUG Library Disk 9.

....

SEARCH

by Carl Singer

SEARCH is an enhancement of the Micropolis SEAR command. The new command searches for any operator-defined string, and prints the addresses of the matches on the console, eight to a line. In order to get the new code into MDOS without using patch area, it was necessary to eliminate SEARN, which I have never found a use for. In any case, its function can be duplicated by FILLing another area of memory with the character, and performing a COMP. All of the non-matches would then appear.

The seven bytes occupied by SEARN in the command table have been used (after moving up TYPE and the zero table terminator) to relocate a break subroutine formerly in SEAR. References to this subroutine by other commands have also been appropriately changed.

There is a small bug in the new SEAR command which could not be eliminated within the space requirement:

If a search to the end of memory is being conducted, and upon arrival at FFFF, the string is partially (but not completely) matched, the program will loop forever, and the computer will have to be reset. If desired, the CALL SPACEOUT near the end of the program can be replaced by CALL 214AH. This will allow a control-C to terminate the loop. The addresses printed on the console will then have only a single space between them.

SEARCH is available on Library Disk 9.

....

MDOSPATCH

by Carl J. Singer

MDOSPATCH contains a patch to DUMP to provide ASCII as well as hex dump. Requires terminal with 80-character width. It also contains a patch to FILES to produce two-column list.

MDOSPATCH is available on MUG Library Disk 9.

....

DOCUMENT GENERATOR

by Daniel Jamba

I am enclosing a program called 'DOCGEN' which I personally developed. It is a software library documentation generating program.

After developing 'DOCGEN' I found it to be a little impractical in use and finally replaced it with my word processor, SPELLBINDER, which has very powerful macro programming capabilities.

Anyway, it should give someone a good base for altering to their specific requirements. It has some Sorcerer dependent coding but there is sufficient explanation in the file 'DOCGEN.DOC' to make the appropriate changes. Just load 'DOCGEN.DOC' from basic and LIST it.

DOCGEN is available on MUG Library Disk 11.

....

Z80 ASSEMBLER

by Robert C. Manderson

Included on this disk is the complete source code for a Z80 assembler which runs under MDOS. It accepts all existing ASSM source files and will correctly assemble them. Additionally, it will assemble all Z80 instructions except those associated with the interrupt vector register, the refresh register and those opcodes which perform block I/O and indirect I/O via the C register. It includes some new pseudo operations, DV, PUT, ELSE and the conditional pseudops have been renamed.

DV permits the programmer to define an assembly variable whose value can be changed during the assembly. It differs therefore from a constant which has been defined via an EQU pseudop. A constant defined by EQU cannot be changed later in the assembly without producing an error. A variable can be changed later by means of the PUT statement. Within conditional code blocks you can now specify an ELSE clause.

I have tried to maintain compatibility with the old ASSM assembler. Thus, this assembler extends the INTEL mnemonics to cover the Z80.

It includes SLF, which is an undocumented Z80 instruction. It works, however, on both my Z80 chips, one from MOSTEK and the other from ZILOG.

The INTEL RRC and RLC have been changed to RRCA and RLCA. I have also removed the 8085 RIM and SIM instructions. These are the only name changes between the two assemblers.

There is one reserved variable, called LVL. It is incremented once for every PUSH instruction and decremented once for every POP instruction. Thus, it may be used to signal stack errors during assembly.

The assembler now contains SYMSAVE. It is invoked at the start of assembly by passing a fourth ASCII parameter in the assembler parameter string.

```
ZASM "SOURCE" "OBJECT" "PLT" "SYMBOLS"
```

This invokes the assembler, which will assemble the file called SOURCE into a file called OBJECT. It will produce a paginated listing without line numbers and will also list the symbol table. Finally it will create a LINEEDIT file, called SYMBOLS, which contains the symbol table. There is no mask string option, as provided by the Micropolis SYMSAVE utility. This is not a problem, however, as the symbol table will be saved in alphabetical order.

There is an undocumented feature in the Micropolis ASSM which I have carried through into this one. You can pass a single quote in a literal string by triple quoting it.

```
 MVI A,'''' ;EVALUATES TO MVI A,27H
```

This assembler began life as the Micropolis ASSM. I disassembled it, commented it and then modified it. By now, however, I have made rather extensive modifications and I doubt if anyone apart from the original author and myself could undo my changes. That being the case, I consider myself justified in naming it the MANDERSONICS Z80 ASSEMBLER. I am currently trying to add MACRO facilities to it and when that is working I shall be adding relocatiblity via a linker.

The MANDERSONICS Z80 ASSEMBLER is available on MUG Library disk 9.

....

NEW ADDITIONS TO THE MUG LIBRARY

MUG MOD II Library Disk 08, Revision 00

Name	Typ	Rv	Sze	Author/Description
MENU	BAS	00	00B	Rudow, B. Master Menu, including auto-configuration.
MENU.A	BAS	00	006	Rudow, B. Applications & Utilities Menu.
DATE.CK	BAS	00	006	Rudow, B. Verify/Change System Date.
TIMECARDS	BAS	00	008	Campbell, J. Converts Clock Time to Decimal Hours, Totals Timecard.
PAYROLL	BAS	00	017	Rudow, B. An Alabama Version of M. Mac Kenzie's Ontario Payroll.
FINANCE	BAS	00	015	Geyer, K. More or less as supplied on VG systems. Future value of investments, regular deposits, etc.
CONVERTNUM	BAS	00	007	Smith, B. Converts numbers between number systems.
PLMENU	BAS	00	005	Peters, R. Profit & Loss Program Menu.
PLCREATE	BAS	00	009	Peters, R. Data Entry for P&L.
PLPRINT	BAS	00	011	Peters, R. Data Print for P&L.
PLREVISE	BAS	00	00B	Peters, R. Data Revision for P&L.
PLINIT	BAS	00	005	Peters, R. Data Initialization for P&L.
PL1	DAT	00	000	Peters, R. Data File for P&L.
PL2	DAT	00	000	Peters, R. Data File for P&L.
PL3	DAT	00	000	Peters, R. Data File for P&L.
MODREC	BAS	00	00F	Grace, J. Allows viewing, correcting, inserting, or deleting of records within a file.
DAY	BAS	00	011	Dembinski, N. Converts back & forth between dates and a unique day number. Good for 1/1/1901 to 12/31/2099.
MENU.G	BAS	00	005	Rudow, B. Games Menu.
DRAWPOKER	BAS	00	008	Geyer, K.
SLOTS	BAS	00	007	Geyer, K.
CHIEF	BAS	00	00A	Geyer, K.
CHECKERS	BAS	00	010	Geyer, K. Doesn't work well.
MAZE	BAS	00	00F	Geyer, K. Doesn't always work well.
BIORHYTHM	BAS	00	00D	Peters, R. Biorhythm charts.

....

MUG MOD II Library Disk 09, Revision 00

Name	Typ	Rv	Sze	Author/Description
$BATCHCOPY	SRC	00	01C	Singer, C. Batch FILECOPY. Can move up to 50 files in a single call. Doesn't have to be resident on any disk involved in the copying.
BATCHCOPY	USR	00	005	Singer, C.
FILECOPY	SRC	00	00E	Singer, C. FILECOPY disassembly.
$SEARCH	SRC	00	00C	Singer, C. Replacement for SEAR to allow string searches.
RESTORE.D	SRC	00	00C	Singer, C. Finds and reconstructs SCRATCHed files.
RESTORE	USR	00	006	Singer, C.
ASMDOC	SRC	00	020	Manderson, R. This, and the following 12 files are a Z80 assembler.
ZASM	SYS	00	017	Manderson, R.
ASS	SRC	00	00E	Manderson, R.
ASS1	SRC	00	02C	Manderson, R.
ASS2	SRC	00	020	Manderson, R.
ASS3	SRC	00	026	Manderson, R.
ASS4	SRC	00	016	Manderson, R.
ASS5	SRC	00	02D	Manderson, R.
ASS6	SRC	00	024	Manderson, R.
ASS7	SRC	00	013	Manderson, R.
ASS8	SRC	00	00C	Manderson, R.
Z80	SRC	00	01B	Manderson, R.
SYSQ3	SRC	00	00C	Manderson, R.
GET-TRAX.D	SRC	00	019	Singer, C. General purpose utility for inspection of disk tracks without recourse to DIR.
GET-TRAX	USR	00	005	Singer, C.
$PASS	SRC	00	00F	Risley, D. Implements a security code into your system. User selectable size & contents of code.
$MDOSPATCH	SRC	00	00D	Singer, C. Patch to DUMP to provide ASCII as well as hex dump. Requires terminal with 80-column width. Patch to FILES to provide two-column list.
$SYMSHELL	SRC	00	01A	Singer, C. Symbol Sort-List Utility.
SYMSHELL	USR	00	004	Singer, C.
$RENUM	SRC	00	049	Singer, C. Utility for renumbering BASIC lines for Version 3.0.
RENUM	SYS	00	008	Singer, C.

....

MUG MOD II Library Disk 10, Revision 00

Name	Typ	Rv	Sze	Author/Description
MENU	BAS	00	00C	Rudow, B. This is a catalog system for Micropolis. Allows three files of 100 disks each. Output lists contents sequentially by disk, or sorted by program name. Sorted output also shows TYPE, SIZE and disk location.
DATE.CK	BAS	00	006	Rudow, B.
DIRINPUT	BAS	00	00A	Rudow, B.
DIRSORT/A	BAS	00	005	Rudow, B.
DIRSORT	BAS	00	009	Rudow, B.
DIRLST.SRT	BAS	00	007	Rudow, B.
DIRLST.SEQ	BAS	00	006	Rudow, B.
DIDAT.SEQA	DAT	00	000	Rudow, B.
DIDAT.SEQB	DAT	00	000	Rudow, B.
DIDAT.SEQM	DAT	00	028	Rudow, B.
DIDAT.SRTA	DAT	00	000	Rudow, B.
DIDAT.SRTB	DAT	00	000	Rudow, B.
DIDAT.SRTM	DAT	00	011	Rudow, B.

....

MUG MOD II Library Disk 11, Revision 00

NAME	TYP	RV	SZE	AUTHOR/DESCRIPTION
DOCGEN.BAS	BAS	00	032	Jamba, D. This, and the following ten files, are a document generator (word processor), written in BASIC. Contains EXIDY dependent code and must be modified for operation on other systems.
VUTIL..OBJ	OBJ	00	002	Jamba, D.
DGCONT.BAS	BAS	00	008	Jamba, D.
DGENTR.BAS	BAS	00	02E	Jamba, D.
DGLIST.BAS	BAS	00	00F	Jamba, D.
DGSAVE.BAS	BAS	00	007	Jamba, D.
DGREAD.BAS	BAS	00	006	Jamba, D.
DGEROR.BAS	BAS	00	004	Jamba, D.
DGALTR.BAS	BAS	00	00F	Jamba, D.
DOCGEN.LIB	DAT	00	013	Jamba, D.
DOCGEN.DOC	BAS	00	019	Jamba, D.
SINITIAL	SRC	00	064	Hall, L. Converts North Star disks to Micropolis - if you have BOTH controllers in your system.
LDOUT	SRC	00	00A	Hall, L. RES mod for Diablo 1610 or 1620 and SIO-2 board.
LIFE	BAS	00	008	McGraw, D. This, and the following eight files, are the game of LIFE. Contains Compucolor dependent code and must be modified for operation on other terminals.
SRCDIS	SRC	00	003	McGraw, D.
SCRSET	SCR	00	010	McGraw, D.
SRCCOM	SRC	00	00B	McGraw, D.
SETUP	OBJ	00	003	McGraw, D.
DISPLAY	OBJ	00	002	McGraw, D.
COMPUTE	OBJ	00	002	McGraw, D.
INSTRUCT	BAS	00	00F	McGraw, D.
DEARBUZZ	SRC	00	00C	McGraw, D.

....

CONTROL-P FOR RES

by Benoit Carignan

At address 04EB (on my system), there is a byte that can be changed to assign the CRT to the printer. If the byte is 03 then the CRT and the printer are receiving data from the same stream. If it is 02 then only the CRT receives data.

As shown in the following routine, what I do is check every keyboard input for a Control-P. If so, then if byte 04EB is a 02, I change it to 03. If it's 03, I change it to 02. Very simple, but I don't know if it will work for all systems.

I found the answer in the DEV055 routine, where they force assignment states to 01/02.

If you do not use the software init (CDINIT) routine, you can make the following patches. If you do use CDINIT, you will have to change the CALL address at location 635, and the routine at 0661 would have to be put elsewhere. In this case, do not change the content of location 04FCH.

PROGRAM TO USE CNTR-P TO ENABLE/DISABLE PRINTER

```
(location)
0630H   MOV   A,B
        ANI   07FH
        CPI   10H   ;IS IT A CNTL-P
        CZ    0661H ;YES, CALL ROUTINE
        RET         ;IF NOT, RETURN

0661H   PUSH  PSW   ;SAVE ACCUMULATOR
        LDA   04EBH ;LOCATION OF ASSIGN
        CPI   03    ;IS PRINTER ASSIGNED TO CRT
        JNZ   066C  ;IF NOT, ASSIGN IT
        DCR   A
        DCR   A
066CH   INR   A
        STA   04EBH ;STORE AT ASSIGN LOCATION
        POP   PSW
        RET
```

....

SPEEDUP FOR SORCERER BASIC

by John Donaldson

Listed below is a patch to Micropolis MDOS which improves BASIC's speed considerably. The speed improvement comes by deleting some of the syntax checking. The BASIC works in a Sorcerer and should work with other Z80 machines. It is entered through MDOS. When entering the following lines, press RETURN after each. From MDOS, type -

```
>ENTR 690
CD D1 EA 20 02 3C C9 FE 03 20 08
CD 09 E0 20 FB 3E 10 C9 CD 09 E0
28 FB CD 09 E0 FE 03 28 F1 FE 1B
28 F3 3C C9/
>ENTR 788
CD 90 06/
>DUMP 690 6B4
690 CD D1 EA 20 02 3C C9 FE 03 20 08 CD 09 E0 20 FB
6A0 3E 10 C9 CD 09 E0 28 FB CD 09 E0 FE 03 28 F1 FE
6B0 1B 28 F3 3C C9
>DUMP 788 78A
788 CD 90 06
>TYPE "RES" 0
>SCRATCH "RES"
>SAVE "RES" 2B1 1598 3
```

If you want to start/stop the display with RUNSTOP/ESC, type the fourth line under "ENTR 690" as

```
20 F5 3C C9/
```

The DUMP commands are needed only to verify what you put in under ENTR. If you made a mistake, go back and reENTR the proper codes.

ED NOTE: As with all patches and saves of RES, you should try the changes on a backup disk. Never modify your prime system disk. Sooner or later you'll make the system hang and won't be able to recreate your original disk. Another general rule is to do as little as possible before saving RES. Some MDOS operations change variable values. When RES is saved with these changed values, it sometimes won't initialize correctly.

....

80 COLUMN DISPLAY FOR THE SOL

For those of you who love your SOLs (me included), but who have trouble with application software that requires an 80-column video terminal, here's a solution.

Micro-Digital has a modification for the SOL that allows both 64 and 80 column display, switch selectable. Included in the mod is a new Dual Personality Module which can be configured to put the Monitor at F000 instead of C000, allowing you to run with 60K of contiguous memory. The combination of 80 column display and 60K of memory is perfect for business applications.

The cost is $370, $275 for the Dual 80/64 Board and $95 for the Dual Personality Module. The Personality Module can be ordered separately if you want to expand your memory but don't care about display size. The screen mod does require changes to the main SOL board. If you live in the San Diego area, Micro-Digital will install the mod for $50. Also required is a CRT monitor that writes line by line (non-interleaved). Micro-Digital offers a BMC CRT green phosphor monitor for $245. Other products for the SOL include 64KRA RAM and Keyboards.

For further information, contact Grover Yowell at P.O. Box 23357, San Diego CA 92123. Phone (714) 569-8540 Ext. 3798.

....

A LITTLE DIVERSION

The following files were found in a professional data base. Hope they don't offend anyone. Just thought you might find it iteresting to see what the "big-boys" do to pass their time.

DEFINITIONS

CARD READER:	Woman who tells fortunes.
DUMMY ARGUMENTS:	Discussions between operators.
DUMP:	Slang term for computer building.
ENTERNAL STORAGE:	Wastebasket.
FORMAT:	What you wipe your feet on before entering the computer building.
GO TO:	Often used in conjunction with Biblical locations.
LOOP:	See: Infinite loop.
MACRO:	The last half of an expression of surprise: "Holy Macro!"
POST MORTEM DUMP:	Place for dead programmers.
INFINITE LOOP:	See: Loop.
SIGNIFICANT DIGIT:	Middle finger of either hand, commonly used by programmers while talking to operators.
VARIABLE:	Bisexual programmer.
BIT:	The increment by which programmers slowly go mad.
COUNTER:	A device over which martinis are served.
TRAPEZOIDAL RULE:	Country ruled completely by trapezoids.
FUNCTION:	What a program never does on the first run.

OPCODES

CODE	INSTRUCTION	OP CODE
BH	Branch and Hang	1G
IIB	Ignore Inquiry and Branch	1H
TDB	Transfer and Drop Bits	1J
DO	Divide and Overflow	1K
SRZ	Subtract and Reset to Zero	1L
PI	Punch Invalid	1M
SSJ	Select Stacker and Jam	1N
FSRA	Forms Skip and Run Away	1P
RASC	Read and Shred Card	1R
SRSD	Seek Record and Scar Disk	1S
BST	Backspace and Stretch Tape	1T
RIRG	Read Inter-Record Gap	1V

UER	Update and Erase Record	1W
EM	Emulate 407	1X
SPSW	Scramble Program Status Word	1Y
EIOC	Execute Invalid Op Code	1Z
EROS	Erase Read-Only Storage	2P
PBC	Print and Break Chain	2R
CM	Circulate Memory	2S
MLR	Move and Lose Record	2T
CNR	Convert to Roman Numerals	2V
DMPK	Destroy Memory Protect Key	2W
DC	Divide and Conquer	2X
EPI	Execute Programmer Immediate	2Y
LCC	Load and Clear Core	2Z
HCF	Halt and Catch Fire	3Z

....

SAVING BASIC PROGRAMS

by Dick Rusczyk

Perhaps the members can use this programming quickie on how to save the right program every time.

If you have ever saved the current program using a valid but incorrect program name, then the following one-line programs within a program is for you. All my programs start with line 9 that will skip line 10.

```
 9 GOTO 11 (or your first program line)
10 SAVE "CURRENTPGM":PRINT"SAVED CURRENTPGM":END
```

By entering line 10 in each program, using the actual program name, and always typing 'GOTO10', the program resaved is always the correct one.

....

CLASSIFIED

WANTED: Vector Mindless Terminal, Flashwriter board and Documentation.

Dr. Kevin Geyer, (714) 962-8240
8591 Whitesails Cir., Huntington Beach CA 92646

....

WANTED: Any information available on Computer Aided Instruction (CAI) for the Micropolis system. Using computer in grade school.

Mauricio Gluck, (305) 673-0866
4510 Pinetree Dr., Miami Beach FL 33140

....

WANTED: Any programs running on the Micropolis system which aid in the management and accounting of stocks, bonds, and other investments.

John G. Scott, (919) 275-2881
E. F. Hutton & Co., Inc.
P.O. Box 749, Greensboro NC 27402

....

WANTED: The current address or phone number of Michael Shrayer. Or, can anyone help me patch Electric Pencil to run on 56K VG. Have 48K EP.

Hans Rakow, (403) 467-8630, home
(403) 436-4414, work

34 Wren Crescent, Sherwood Park, Alta. CANADA
T8A 0G5

....
01/01/82

Published Monthly by the MUG
Subscription rates:
U.S., Canada, Mexico; $18/year: Other, $25/year

FIRST CLASS MAIL
===== ===== ====

FIRST CLASS MAIL
===== ===== ====

==

MICROPOLIS USERS GROUP

Buzz Rudow, Editor
604 Springwood Circle
Huntsville AL 35803
(205) 881-1697

FIRST CLASS MAIL
===== ===== ====