Application Notes for *muLISP*™ *XM* Version 7.20

muLISP XM operates almost identically to ***muLISP-90*** except that it can take advantage of up to 4 gigabytes (4 billion bytes) of extended memory. The ***muLISP XM*** distribution diskette(s) include all the ***muLISP-90*** files plus the file MULISPXM.EXE. These notes summarize things unique to running ***muLISP XM***.

To run ***muLISP XM***, transfer all the files on the distribution diskette(s) to a MULISPXM subdirectory (see Section 1.4 of the ***muLISP* Reference Manual**) and then type MULISPXM at the MS-DOS prompt. The MULISPXM command can optionally be followed by a LSP or SYS file name. Type (RECLAIM) at the ***muLISP XM*** dollar sign prompt to see how many bytes of memory are free.

Controlling *muLISP XM* Memory Usage

muLISP XM normally takes all available extended memory when it starts running in order to maximize the size of application programs it can handle. However, taking all such memory may be undesirable in multi-tasking environments like Microsoft's Windows or IBM's OS/2.

If you are having difficulties running ***muLISP XM***, you can limit the amount of extended memory it uses by issuing the MS-DOS command

```
SET LXM=-MAXP n
```

before running ***muLISP XM***. *n* is the maximum number of bytes ***muLISP XM*** will use. It must be an integer entered in decimal notation or in hexadecimal notation followed by the letter H (e.g. 100000H). Note that ***muLISP XM*** requires a minimum of 1 megabyte = 100000H bytes of extended memory to run.

Often it is convenient to temporarily run other application programs from within ***muLISP XM*** using the EXECUTE function (see Section 5.6.5 of the ***muLISP* Reference Manual**). For this reason ***muLISP XM*** normally uses only a small amount of conventional memory (the computer's first megabyte of memory). This leaves the rest of conventional memory available for use by other application programs.

However, if your application program is large and you want to use every bit of available memory, you can force ***muLISP XM*** to use all conventional as well as extended memory by issuing the MS-DOS command

```
SET LXM=-MAXR 0
```

In the unlikely event that you want to both limit the use of extended memory *and* use conventional memory, issue the command

```
SET LXM=-MAXP n -MAXR 0
```

Running *muLISP XM* Under Microsoft Windows

From the Windows Manager select the File Run command and type MULISPXM preceded by the drive and directory in which it resides. To simultaneously load a LSP file, follow the MULISPXM command with the file name (see Section 2.9). For example, type

```
C:\MULISPXM\MULISPXM HANOI.LSP
```

to load ***muLISP XM*** from the MULISPXM directory on drive C and then run the Tower of Hanoi demonstration program file HANOI.LSP.

Included on the ***muLISP XM*** diskette is the file MULISPXM.PIF. It is necessary for running ***muLISP XM*** under Windows in standard mode. MULISPXM.PIF allocates all available extended memory to ***muLISP XM***. MULISPXM.PIF has no effect when running under Windows in enhanced mode.

Differences Between *muLISP XM* and *muLISP-90*

- If the third argument to the functions CSMEMORY and DSMEMORY is nonNIL, double words (4 bytes) instead of words (2 bytes) are accessed.
- The functions MEMORY and NEW-CODE-SPACE are unnecessary for ***muLISP XM*** and are therefore not defined.
- The function ALLOCATE merely returns the size of the ***muLISP XM*** machine code in bytes. It does not allocate memory.
- The function SNAPSHOT operates relative to the ***muLISP XM*** data segment rather than to absolute memory addresses.
- The native code compiler (see Chapter 8) generates 8086 machine code for ***muLISP-90***. It does not generate 80386 machine code for ***muLISP XM***.
- The description for implementing machine-coded routines (see Chapter 9) is for ***muLISP-90***. It is not completely applicable to ***muLISP XM***.
- ***muLISP XM*** can read source code (LSP) files produced by ***muLISP-90***, and vice versa. However, ***muLISP XM*** can *not* read memory image (SYS) files produced by ***muLISP-90***, or vice versa.
- ***muLISP XM*** does *not* have the ability to produce EXE or COM files. Thus, ***muLISP XM*** application programs can *not* be distributed as ***muLISP*** runtime systems.
- Some of the variables in the ***muLISP*** Base Page (see Appendix H) are not applicable to ***muLISP XM*** or they are located at different addresses.

muLISP XM Global Storage Area

This is a listing of the ***muLISP XM*** global storage area. The comments include the offset address in the data segment of various system variables. The function DSMEMORY (see Section 5.8.7) can be used by application programs to determine the current value of these variables.

```
BASSYMB  EQU  4600H     ;Base of symbols = (LOCATION NIL)
NXTSYMB  DD   ?         ;00H=0: Next symbol
NXTNUMB  DD   ?         ;04H=4: Next number
ENDATOM  DD   ?         ;08H=8: End of atom space

BASCONS  EQU  ENDATOM   ;08H=8: Base of cons space
NXTCONS  DD   ?         ;0CH=12: Next cons
ENDCONS  DD   ?         ;10H=16: End of cons space

BASPNS   EQU  ENDCONS   ;10H=16: Base of print-name string space
NXTPNS   DD   ?         ;14H=20: Next print-name string
MAXPNS   DD   ?         ;18H=24: Maximum print-name string

MAXVECT  EQU  NXTPNS    ;14H=20: Maximum number vector
NXTVECT  EQU  MAXPNS    ;18H=24: Next number vector
BASVECT  DD   ?         ;1CH=28: Upper end of vector space

BASCODE  EQU  BASVECT   ;1CH=28: Base of D-code space
NXTCODE  DD   ?         ;20H=32: Next D-code
MAXCODE  DD   ?         ;24H=36: Maximum D-code

MAXFCB   EQU  NXTCODE   ;20H=32: Maximum FCB
NXTFCB   EQU  MAXCODE   ;24H=36: Next FCB
BASFCB   DD   ?         ;28H=40: Upper end of FCB space

IFCB     DD   ?         ;2CH=44: Current input FCB pointer
OFCB     DD   ?         ;30H=48: Current output FCB pointer

         DD   ?         ;34H=52: Reserved space
         DD   ?         ;38H=56:
         DD   ?         ;3CH=60:
         DD   ?         ;40H=64:

THRVAL   DD   ?         ;44H=68: Thrown value (0=inactive)
PRCSN    DD   ?         ;48H=72: Current precision
UDFLW    DD   ?         ;4CH=76: Current underflow

GCCTR    DD   ?         ;50H=80: Total GC's counter
RACTR    DD   ?         ;54H=84: Total RA's counter
```