

**N E W S L E T T E R   # 1 3**  
**S e p t e m b e r   1 9 8 5**

Aloha from Hawaii!    The Soft Warehouse Newsletter provides you with information on new Soft Warehouse products, and software extensions or corrections to existing products. In addition, the newsletter is a medium for the exchange of ideas and application programs within the growing community of **muMATH** and **muLISP** users.

If you would like to subscribe, or extend your subscription to the Newsletter for three issues beyond the expiration number on your mailing label, please send \$6 (\$10 for orders from outside the U.S. or Canada) by check, VISA, or Master Card to Soft Warehouse, Inc., P.O. Box 11174, Honolulu, Hawaii, 96828, U.S.A. A complete set of back issues is available on request for \$15 (\$20 for orders from outside the U.S. and Canada).

**m u L I S P - 8 5   R E L E A S E D !**

Soft Warehouse, Inc. is proud to announce the release of muLISP-85. This all new '85 version of muLISP provides the personal computer user with a window-based Artificial Intelligence programming environment. The system is based on the fastest Lisp language system available for PC-DOS and MS-DOS computers.

Compare muLISP with its competitors: a) muLISP programs typically execute 3 times faster, b) in a given amount of computer memory, muLISP can store 2 to 3 times as much program and data, and c) since muLISP runs on any generic MS-DOS computer, applications written in muLISP are not limited to the IBM PC market.

muLISP-85 has more than twice the power of the well received muLISP-83: a) 260 primitives vs. 120, b) use of up to 512K bytes vs. 256K, c) exact and approximate rational arithmetic as well as integer arithmetic, d) user-definable macros as well as functions, e) a soon to be released native-code compiler in addition to the existing pseudo-code compiler, and f) the power of Common LISP while maintaining muLISP's reputation for high performance and compactness. And yet for all the power of the '85 version, the retail price of muLISP is still only \$250!

In short, muLISP-85 provides the personal computer user with a fully integrated LISP programming environment for the development of sophisticated Artificial Intelligence software systems. The performance of muLISP-85 in both speed and capability far exceeds that of all other microcomputer based LISP systems.

muLISP-85 is available now from Soft Warehouse, Inc. and will soon be available from our publisher Microsoft. For pricing and availability contact your software dealer or write directly to Soft Warehouse, Inc., P.O. Box 11174, Honolulu, HI, 96828-0174, U.S.A.

## C A L L   F O R   P A P E R S

The fifth National Conference on Artificial Intelligence, AAAI-86, will be held in Philadelphia, Pennsylvania August 11 - 15, 1986. The conference sponsors, the American Association for Artificial Intelligence (AAAI), has put out a call for papers. For information, write AAAI, 445 Burgess Drive, Menlo Park, CA, 94025-3496, U.S.A.; or telephone (415) 328-3123.

The SYMSAC-86 Computer Algebra Conference will take place at the University of Waterloo from July 21-23. Professor Keith Geddes at the computer science department (Ontario, Canada, N2L 3G1) is in charge of local arrangements, and the program chairman is Professor David Yun at the Computer Science Department of Southern Methodist University in Dallas, Texas, 75275.

## \* \* \* \* \*   T h e   m u M A T H e m a t i c i a n   \* \* \* \* \*

muMATH is continually undergoing improvement. Currently Soft Warehouse, Inc. is shipping version 4.12 of muMATH-83. In addition to a number of bug fixes and internal improvements, version 4.12 extends the class of cubic equations that muMATH can solve. Also the efficiency of the muSIMP compiler and the density of the code it produces has been improved. If you have version 4.xx of muMATH-83 and want to update to 4.12, pack your master disks in an unbendable container and send them to Soft Warehouse, Inc. with \$20 (\$25 for customers outside the U.S. and Canada). We will update your disks and return them along with approximately 20 pages of manual revisions. Payment must accompany update orders. Money orders, checks, Visa or Master Card accepted.

### **muMATH Users**

The following people have expressed an interest in communicating with other muMATH users for the exchange of application programs and user tips. Send the Newsletter editor your name and address if you want to be listed in the next issue.

Dr. Arnold O. Allen, POBox 75158, Los Angeles, CA, 90075  
Massimo Andretta, Tema Societa per Azioni, Viale Aldo Moro 38,  
Bologna, 40127, ITALY  
Dr. F. S. Binkowski, Environmental Protection Agency, Mail Drop 80,  
Research Triangle Park, NC, 27711  
Thomas W. Bonsor, Econ., Eastern Washington U., Cheney, WA, 99004  
Frank Bryan, Princeton U., 3V Hibben Apt., Faculty Rd., Princeton,  
NJ, 08540  
Kenneth D. Carter, The Videographics Corp., 8701 Georgia Ave. #404,  
Silver Spring, MD, 20910  
Chan, Mr. Hing-Lun, 21 Robinson Rd., Good View Court 6B, HONG KONG  
Chapin Day, 5106 Bothe Ave., San Diego, CA, 92122  
Richard T. Dillon, 5808 Pojoaque Rd. NE, Albuquerque, NM, 87110  
Dr. Robert S. Eisenberg, Physiology, Rush Medical College, 1750 West  
Harrison St., Chicago, IL, 60612

Paul F. Emerson, U. of MN, 438 4th St. NE, Minneapolis, MN, 55413  
 James F. Epperson, 505 Sunset Dr., Athens, GA, 30606  
 Joel H. Fink, Negion, Inc., 4023 East Ave., Hayward, CA, 94542  
 Thomas B. Fowler, Mathematics, Christendom College, Rt. 3, Box 87,  
 Front Royal, VA, 22630  
 Dr. William Garland, Engrg. Physics, McMaster U., Hamilton, Ontario,  
 L8S 4M1, CANADA  
 Martin Griss, HewlettPackard, 1501 PageMill Rd, Palo Alto, CA, 94303  
 Ciro Guajardo, Teledyne Relays, 12525 Daphne Ave, Hawthorne CA 90250  
 Dr. J. O. Hirschfelder, Chem., U. of WI, Madison, WI, 53706 (Oct.-  
 Apr. @ Chem., UC Santa Barbara, Santa Barbara, CA, 93106)  
 Dr. Larry Husch, Math, U. of TN, 121 Ayres Hall, Knoxville, TN 37996  
 C. Allan Jorgenson, 9103 Quarter Ct., Vienna, VA, 22180  
 Richard Kohoutek, Civil Engrg., U. of Wollongong, POBox 1144,  
 Wollongong, NSW, 2500, AUSTRALIA  
 Willis E. Lamb, 848 N. Norris Ave., Tucson, AZ, 85719  
 Prof. M. Lieber, Physics, U. of Arkansas, Fayetteville, AR, 72701  
 Jeff D. Madura, 2501 Soldiers Home Rd. #53, W. Lafayette, IN, 47906  
 Jack Mallinckrodt, Communications Research Labs, 14141 Stratton Way,  
 Santa Ana, CA, 92705  
 Geoffrey R. McNall, 1664 Estevan Rd, London, Ontario, N5X 2G9 CANADA  
 Jose Luis Mora\_Castro, CISGO Dept., Univ. Auto. Metro. - Iztapalapa,  
 Apt. Post. 55-532, 09340, MEXICO, D.F.  
 Prof. Mitsuo Morimoto, Math., Sophia U., 7 Kioicho Chiyodaku, Tokyo,  
 102, JAPAN  
 Meyer Nahon, Inst. for Aerospace Studies, U. of Toronto, 4925  
 Dufferin St., Downsview, Ont., M3H 5T6, CANADA  
 Dr. M. Nair, Math., U. of Glasgow, Gilmore Hill, Glasgow, G12 8QW,  
 SCOTLAND  
 Dr. M. Ochiai, Math., Osaka U., Toyonaka, Osaka, 560, JAPAN  
 Gino Panetta, Rue Walkiers 7, Bruxelles, 1030, BELGIUM  
 Larry Pleskac, 17155 W. Bernardo Dr. #106, San Diego, CA, 92127  
 James M. Potter, 2245 47th St., Los Alamos, NM, 87544  
 Lewis W. Procop, 354 Pin Oak Lane, Westbury, NY, 11590  
 Art Rasmussen, 612 West Hillcrest, Keene, TX, 76059 (Has anyone  
 written a **CAI program** in muMATH/muSIMP that teaches HS math?)  
 Prof. D. Riordan, Computer Science, Rhodes U., POBox 94,  
 Grahamstown, 6140, SOUTH AFRICA  
 Ronald W. Satz, Transpower Corp., 1 Oak Dr., Parkerford, PA, 19457  
 Tom Scavo, Central Wyoming College, 2660 Peck Ave., Riverton, WY,  
 82501-1520  
 Robert C. Schober, Biomedical LSI, 9411 Tiki Circle, Huntington  
 Beach, CA, 92646  
 Dale Shultz, 509 E. Clark, Apt. 10, Champaign, IL, 61820  
 Terry Silva, Merced HS North, 205 W. Olive, Merced, CA, 95344  
 Mark Skinner, Physics, U. of WI-Madison, 5246 Chamberlin, 1150 Univ.  
 Ave., Madison, WI, 53706  
 Larry Wagner, 610 Wildcat Canyon Rd., Berkeley, CA, 94708  
 Dr. Rainer Walden, Rochusweg 1, D-4795 Delbrueck, WEST GERMANY  
 Andrew Walinski, 80 Broomleys Rd., Coalville, Leicester, ENGLAND  
 Kevin P. Walker, Engineering Science Software, P65 Log Rd.,  
 Smithfield, RI, 02917  
 Peter J. Welcher, Math., US Naval Academy, Stop 9E, Annapolis, MD,  
 21402  
 Marvin Youngstrand, 1221 Victoria St. #403, Honolulu, HI, 96814

### muMATH-83 Utility Functions

Although muMATH-83 uses many integration formulas derived from integration by parts, muMATH does not attempt general integration by parts, since the algorithm may not terminate. However, the function INTPAR contributed by **William Squire** permits you to try integration by parts using a product partitioning that you choose.

```
FUNCTION INTPAR (F, X, U),  
  F: INT (F/U, X),  
  U * F - INT (F * DIF (U, X), X),  
ENDFUN$
```

For example, to integrate the expression  $x \arctan x$  (which cannot be integrated directly by muMATH) enter

```
? INTPAR (x ATAN x, x, ATAN x);  
@: -x/2 + x^2 ATAN x/2 + ATAN x/2
```

The muSIMP "=" operator just tests for identical appearance. Consequently, to test nonnumeric expressions for equivalence it is generally better to test their difference for zero, using appropriate transformations such as EXPD, etc. However, **Werner Mrazek** has pointed out that the ZERO function does not return TRUE for an array having all zero entries. Consequently, here is a recognizer that tests for zero scalars or arrays:

```
FUNCTION ZEROARR (EX1),  
  WHEN ARRAY (EX1),  
    POP (EX1),  
    LOOP  
      WHEN ATOM (EX1), EXIT,  
      WHEN NOT ZEROARR (POP (EX1)), FALSE EXIT,  
    ENDLLOOP EXIT,  
  WHEN ZERO (EX1), EXIT,  
ENDFUN$
```

**Jim Jara-Almonte** wanted INT to distribute over the elements of arrays. The following function and properties accomplish this.

```
FUNCTION MAPINT (LEX1), % refers to outside variable INDET %  
  WHEN ATOM (LEX1), FALSE EXIT,  
  ADJOIN (INT (POP (LEX1), INDET), MAPINT (LEX1)),  
ENDFUN$
```

```
PROPERTY INT, [, FUNCTION (LEX1),  
  ADJOIN ('[, MAPINT (LEX1)),  
ENDFUN$
```

```
PROPERTY INT, {, FUNCTION (LEX1),  
  ADJOIN ('{, MAPINT (LEX1)),  
ENDFUN$
```

### ALGEBRA.ARI Bug Fixes

If your copy of the file ALGEBRA.ARI (or the file ALGEBRA.OPT if using the CP/M version of muMATH) is dated earlier than 05/30/85, change the second to last line of the property **\***, **!** from

```
        WHEN MINUS1 (EX1), EX2 ^ -1 EXIT EXIT,  
to  
        WHEN MINUS1 (EX1), (EX2+1) ^ -1 EXIT EXIT,
```

and change the file date to 05/30/85. **Jon Silver**, Bethesda, MD.

If your copy of the file ALGEBRA.ARI (or the file ALGEBRA.OPT if using the CP/M version of muMATH) is dated earlier than 09/09/85, change the definition of the complex conjugate function **CONJ** to

```
FUNCTION CONJ (EX1),  
    EVSUB (EX1, #I, -#I),  
ENDFUN$
```

and change the file date to 09/09/85. **Colin Gyles**, Danvers, MA.

### LOG.ALG Bug Fix

If your copy of the file LOG.ALG is dated earlier than 07/19/85, change the property **PRTMATH**, **LOG** to read

```
PROPERTY PRTMATH, LOG, FUNCTION (LEX1),  
    WHEN SECOND (LEX1) EQ #E,  
        PRTPAR ('LN, LPAR),  
        PRTOPE ('LN, LIST (FIRST (LEX1))),  
        PRTPAR ('LN, RPAR), TRUE EXIT,  
    WHEN SECOND (LEX1) = LOGBAS,  
        PRTPAR ('LOG, LPAR),  
        PRTOPE ('LOG, LIST (FIRST (LEX1))),  
        PRTPAR ('LOG, RPAR), TRUE EXIT,  
ENDFUN$
```

and change the file date to 07/19/85. **Daoud Serang**, Spring House, PA.

### ODEMORE.ODE Bug Fix

If your copy of the file ODEMORE.ODE is dated earlier than 07/10/85, change the 20th line of the function **RICCATI** from

```
    (#UNK - FIRST (EX2)) * (THIRD (EX2) - SECOND (EX2)) *  
to  
    (#UNK - FIRST (EX2)) * (THIRD (EX2) - SECOND (EX2)) -
```

if it has not already been changed, and change the file date to 07/10/85. **Alan K. Head**, C.S.I.R.O, Melbourne, Australia.

### MATRIX.ARR Bug Fix

A bug in the matrix package occurred when taking the transpose of ragged and nonsquare arrays. The following fix, contributed by **Michael Wochnik**, corrects the bug while still yielding a result that is an attractive row of columns. If your copy of the file MATRIX.ARR is dated earlier than 09/09/85, replace **PROPERTY ` , {** with

```
FUNCTION MAX (EX1, EX2),
  WHEN EX1 > EX2, EX1 EXIT,
  EX2
ENDFUN$

FUNCTION LONGEST (LEX1,
  % Local: % EX1),
  EX1: 0,
  LOOP
    WHEN EMPTY (LEX1), EX1 EXIT,
    EX1: MAX (LENGTH (POP (LEX1)), EX1)
  ENDLLOOP
ENDFUN$

PROPERTY ` , {, FUNCTION (LEX1),
  WHEN ROW (FIRST (LEX1)),
    IDMAT (LONGEST (LEX1) - 1) .
    ADJOIN ('[, MAPFUN ('`, LEX1)) EXIT,
  ADJOIN ('[, MAPFUN ('`, LEX1))
ENDFUN$
```

and change the file date to 09/09/85. The following example should now work correctly:

```
? EG: {[2, 3], [4, 1, 6], [5]};
@: {[2, 3],
    [4, 1, 6],
    [5]}

? EG`;
@: {[2, 4, 5],
    [3, 1, 0],
    [0, 6, 0]}
```

### Polynomial Addition Speed Up

**Alan K. Head** has given us permission to publish the following file, which can dramatically speed up large polynomial computations. The file can be read in after loading ARITH.MUS and ALGEBRA.ARI, or it can be incorporated into ARITH.MUS and ALGEBRA.ARI after deleting the functions MERGELIST and MERGETERM. Either way, be sure to include his copyright notice in the file.

% File: SPEED.UP (C) 08/08/84 Alan K. Head, C.S.I.R.O. %

% Changes and additions to muMATH-83 source file ARITH.MUS. The  
functions MERGELIST and MERGETERM are now unused and can be deleted.%

```
FUNCTION ADDLIST (LEX1,  
  % Locals: % LEX2, LEX3, LEX4, EX1, EX2, EX3, EX4, EX5, EX6, EX7),  
LOOP  
  LEX2: LISTSUM (POP (LEX1)),  
  BLOCK  
    WHEN LEX1, LEX3: LISTSUM (POP (LEX1)) EXIT,  
  ENDBLOCK,  
  LOOP  
    BLOCK  
      WHEN SUM (EX2: FIRST (LEX2)),  
        PUSH (POP (LEX2), LEX1) EXIT,  
      WHEN SUM (EX3: FIRST (LEX3)),  
        PUSH (POP (LEX3), LEX1) EXIT,  
      WHEN EMPTY(LEX4),  
        WHEN LEX3 AND ORDERED (CODIV (EX3), CODIV (EX2)),  
          PUSH (POP (LEX3), LEX4) EXIT,  
          PUSH (POP (LEX2), LEX4) EXIT,  
        WHEN EX1: ADDEXS (EX2, EX4: FIRST (LEX4), EX5: CODIV (EX2),  
          EX7: CODIV (EX4)),  
          POP (LEX2), POP (LEX4),  
          WHEN ZERO (EX1) EXIT,  
          PUSH (EX1, LEX4) EXIT,  
      WHEN LEX3,  
        WHEN EX1: ADDEXS (EX3, EX4, EX6: CODIV (EX3), EX7),  
          POP (LEX3), POP (LEX4),  
          WHEN ZERO (EX1) EXIT,  
          PUSH (EX1, LEX4) EXIT,  
        EX2: ORDERED (EX5, EX7),  
        WHEN (EX3: ORDERED (EX6, EX7)) AND EX2,  
          APPLEEX () EXIT,  
        WHEN NOT (EX2) AND (EX3 OR ORDERED (EX5, EX6)),  
          PUSH (POP (LEX2), LEX4) EXIT,  
          PUSH (POP (LEX3), LEX4) EXIT,  
        WHEN ORDERED (EX5, EX7),  
          APPLEEX () EXIT,  
          PUSH (POP (LEX2), LEX4),  
    ENDBLOCK,  
    WHEN EMPTY (LEX2) AND EMPTY (LEX3) EXIT,  
  BLOCK  
    WHEN EMPTY (LEX2),  
      LEX2: LEX3,  
      LEX3: FALSE EXIT,  
    ENDBLOCK,  
  ENDLLOOP,  
  WHEN EMPTY (LEX1),  
    REVERSE (LEX4) EXIT,  
  APPLEEX (),  
ENDLOOP,  
ENDFUN$
```

```

FUNCTION APPLEX ( ),
    WHEN LEX4,
        LEX1: APPEND (LEX1, LIST (MKSUM (REVERSE (LEX4)))),
        LEX4: FALSE EXIT,
ENDFUN$

FUNCTION LISTSUM (EX1),
    WHEN SUM (EX1),
        REST (EX1) EXIT,
    LIST (EX1),
ENDFUN$

FUNCTION ADDEXS (EX1, EX2, EX3, EX4),
    WHEN ADDTERMS (EX1, EX2) EXIT,
    WHEN EX3=EX4,
        ADDTERMS (COEFF (EX1), COEFF (EX2))*EX3 EXIT,
ENDFUN$

FUNCTION PLSTRAP (LEX1),
    MKSUM (ADDLIST (LEX1)),
ENDFUN$

% Changes to the muMATH-83 source file ALGEBRA.ARI. %

FUNCTION SQUARE (EX1,
    % Locals: % LEX1, LEX2, LEX3, LEX4),
    WHEN SUM (EX1),
        LEX1: REVERSE (REST (EX1)),
        LOOP
            PUSH ((EX1: POP (LEX1))^2, LEX4),
            LEX3: LEX1,
            EX1: 2*EX1,
            LOOP
                WHEN EMPTY (LEX3) EXIT,
                PUSH (POP (LEX3)*EX1, LEX4),
            ENDLLOOP,
            LEX2: ADDLIST (LIST (MKSUM (LEX2), MKSUM (LEX4))),
            WHEN EMPTY (LEX1),
                MKSUM (LEX2) EXIT,
            LEX4: FALSE,
        ENDLLOOP EXIT,
    EX1*EX1,
ENDFUN$

FUNCTION DISTRIB (EX1, LEX1,
    % Locals: % NUMNUM, DENNUM, LEX2),
    LOOP
        PUSH (EX1*POP(LEX1), LEX2),
        WHEN EMPTY (LEX1),
            ADDLIST (LIST (MKSUM (REVERSE (LEX2)))) EXIT,
        ENDLLOOP,
ENDFUN$

RDS()$

```



\* \* \* \* \* T h e m u L I S P e r \* \* \* \* \*

### muLISP Users

The following people have expressed an interest in communicating with other muLISP users for the exchange of application programs and user tips. Send the Newsletter editor your name and address if you want to be listed in the next issue.

Philip Bartlett, 419 Pacific Circle, Newbury Park, CA, 91320-4461  
David Bristor, Hewlett Packard Labs, 3U, 1501 Page Mill Rd., Palo Alto, CA, 94304  
Dr. Mark H. Chignell, Indus. & Sys. Engrg, USC, OHE 400, MC-1452 Univ. Park, Los Angeles, CA, 90089-1452  
Prof. Jerald D. Cole, Math and Computer Science, Central Connecticut State University, 1615 Stanley St., New Britain, CT, 06050  
Rusty Craine, #6 Western Oaks, Fort Worth, TX, 76108  
Paul G. Cuddy, 2411 Holmes St., Kansas City, MO, 64108  
Richard T. Dillon, 5808 Pojoaque Rd. NE, Albuquerque, NM, 87110  
Walter Ebert, Maschinenfabrik Augsburg-Nurnberg, Frankenstr. 150, 8500 Nurnberg 44, WEST GERMANY  
Joseph B. Elad, Data Base Logic, 2700 Philadelphia Pike, Wilmington, DE, 19809  
Robert H. Flast, 660 Fairmont Ave., Westfield, NJ, 07090  
Robert Gregor, 2200 Columbia Pike #903, Arlington, VA, 22204  
Martin Griss, Hewlett-Packard, 1501 Page Mill Rd., Palo Alto, CA, 94303  
James B. Heathman, 8436 Creek St. NE, Albuquerque, NM, 87113  
Ernest L. Hughes, 2520 East Lee, Seattle, WA, 98112  
Larrie V. Hutton, 701 Warren St., Marietta, OH, 45750  
Daniel D. Kessler, Arizona St. U., 2158 E. Concordia Dr., Tempe, AZ, 85282  
Willis E. Lamb, 848 N. Norris Ave., Tucson, AZ, 85719  
Dr. H. G. Lieberman, 6640 Newburgh Rd., Evansville, IN, 47715  
John M. Morrill, Ohio St. U. Hospital, 410 W 10th Ave., Doan Hall 182, Columbus, OH, 43210  
Jack Oslake, 18 Walking Woods Dr., Lake Oswego, OR, 97034  
Edward J. Panzer, 857 Maryhurst Dr., Claremont, CA, 91711  
William E. Pittman, 6302 150th NE, Redmond, WA, 98052  
Paul Renzetti, 16 Village Ct., Brooklyn, NY, 11223  
J. B. Robertson, 814 Wavecrest, Houston, TX, 77062  
Robert C. Schober, Biomedical LSI, 9411 Tiki Circle, Huntington Beach, CA, 92646  
Craig Schwandt, Asymetrix, 110 110th Ave. NE, #617, Bellevue, WA, 98004  
Dr. W. A. Skelton, LTV Aerospace Corp., 901 Rankin, Bedford, TX, 76022  
Jurgen Unger, Bruckenstr. 64, Hofheim-Lorsbach 6238, WEST GERMANY  
Dr. S. L. Wang, Union Carbide Corp. - G2, Old Ridgebury Rd., Danbury, CT, 06817  
Rickey J. Williams, 1544 Green End Ave., Middletown, RI, 02840

Advertisement  
**M a x F A C T S: an Expert System Engine**

Thanks for your interest in MaxFACTS. I'd like to use this space to address the typical questions I have been receiving:

**What are the capabilities of MaxFACTS?** MaxFACTS is a programming language designed to be specifically useful in Expert Systems and A.I. related work. It aids in decision making tasks by performing automatic backtracking in order to 're-evaluate' possible decisions. This frees the knowledge engineer from low level programming tasks otherwise required in conventional languages. In fact, only the backward chainer of MaxFACTS would be considered very 'Prolog-like'. But MaxFACTS has the same power both backward and forward; plus the advantage of software hooks to conventional languages.

**What type of user interface (dialog) does MaxFACTS have?** As part of the current MaxFACTS language you have available both input and output facilities. The output facility allows for clean, English sentences, output with the use of TEMPLATES (similar to FORMATS). Of course, the software hooks also allow access to screen handling routines or conventional language I/O. A more complete natural language facility will be in future releases.

**How are rules and facts entered into MaxFACTS and how are Expert Systems actually developed with MaxFACTS?** MaxFACTS does not provide its own development system. Rules can simply be entered directly into MaxFACTS when it is loaded. OR - simply use your favorite editor! I use Microsoft's WORD<sup>™</sup> to create files of many rules and use the LOADTHEORY command to load them. If I add one or two while in MaxFACTS, then I resave the file with the SAVETHEORY command.

**What are the future plans for MaxFACTS?** Distributed with each MaxFACTS package is a cover letter. It gives users a perspective of where MaxFACTS is and where I plan to take it. One exception to a normal release time table will be the compiled code. The speed increase will be an advantage everyone can use as soon as it becomes available.

**What is included in the MaxFACTS package?** Two different versions of MaxFACTS are available. The full feature version **requires muLISP-85** and includes: 1) the MaxFACTS source file, 2) the user manual with explanation and reference material, and 3) two demonstration files that contain working examples. The stand alone version is the same except that the benefits of Software Hooks are lost since the system is distributed as an executable COM file rather than as a source file.

As of July 1, 1985 the new price for the full feature package is \$240.00 and the price for the stand alone version is \$175.00. Please include \$10.00 for shipping and handling. To order write to **Robert R. McKenzie**, 2518 England St. #1, Huntington Beach, CA, 92648, U.S.A.

Advertisement  
**L I S P - W I N D O W S**

Designed for the IBM PC family of computers and compatibles, LISP-WINDOWS aids in creating window based user interfaces for your muLISP application programs. LISP-WINDOWS accepts keyboard or Microsoft Mouse input. You can create pull-down menus and multiple split-screen or overlapping windows. LISP-WINDOWS is written in and **requires muLISP-85**. Since the source is included, it can be modified to suit your needs. LISP-WINDOWS features include:

- split-screen and overlapping windows
- pull-down menus
- display page access
- keyboard and mouse input
- color-graphics and monochrome support
- fully commented source code

To receive a copy on an IBM-PC formatted diskette, send \$39.95 to **IntelliTECH**, 777 Kapiolani Blvd., Suite 2603, Honolulu, Hawaii, 96813-5211, U.S.A.

**An MS-DOS Command Line Reader**

The following muLISP-83 utility function reads the MS-DOS command line that initiated muLISP and returns it as a name. The muLISP-85 version of this function is simplified by the availability of the function CSMEMORY that facilitates reading bytes in the 8086 code segment.

```
(PUTD 'READ-COMMAND-LINE '(LAMBDA (
  NUM BYTE LST)
  (SETQ NUM 129)
  (LOOP
    (SETQ BYTE (MEMORY (PLUS (TIMES (REGISTER 0) 16) NUM)))
    ((EQ BYTE 13)
     (PACK (REVERSE LST)) )
    (SETQ NUM (PLUS NUM 1))
    (PUSH (ASCII BYTE) LST) ) ) )
```

**Microsoft Mouse Interface Functions**

The file MOUSE.LSP provides an interface between muLISP and the Microsoft Mouse™. If you have an IBM PC or "look-alike" computer and have properly installed the Microsoft Mouse hardware and software (see the Microsoft Mouse Operation Manual), this source file defines functions that determine the position of the mouse, the status of the buttons on the mouse, and the number of times the buttons have been pressed and released. The Microsoft Mouse is sold by **Microsoft Corporation**, 10700 Northup Way, Bellevue, Washington, 98004, U.S.A. The '85 version of MOUSE.LSP is included with the muLISP-85 package.

```

; File:  MOUSE.LSP           09/09/85           Soft Warehouse, Inc.

;   Microsoft Mouse Interface Functions  --  muLISP-83 Version

; This file requires that the Microsoft Mouse hardware and device
; driver software be properly installed.
; The functions automatically update the following global variables:

; *MOUSE-ROW*           the vertical mouse position
; *MOUSE-COL*           the horizontal mouse position
; *LEFT-BUTTON*         T if and only if left button pressed
; *RIGHT-BUTTON*        T if and only if right button pressed
; *BUTTON-PRESSES*      the number of button presses
; *BUTTON-RELEASES*     the number of button releases

(DEFUN SHOW-MOUSE (LAMBDA ()           ;Display mouse cursor
  (REGISTER 0 1) (INTERRUPT 51) ))

(DEFUN HIDE-MOUSE (LAMBDA ()           ;Hide mouse cursor
  (REGISTER 0 2) (INTERRUPT 51) ))

(DEFUN POSITION-MOUSE (LAMBDA (ROW COL) ;Current mouse position
  ((AND (NUMBERP ROW) (NUMBERP COL))
   (REGISTER 0 4)
   (REGISTER 2 COL) (REGISTER 3 ROW)
   (INTERRUPT 51) )
  (REGISTER 0 3)
  (INTERRUPT 51)
  (SETQ *MOUSE-ROW* (REGISTER 3))
  (SETQ *MOUSE-COL* (REGISTER 2)) ))

(DEFUN STATUS-MOUSE (LAMBDA ()         ;Current button status
  (REGISTER 0 3)
  (INTERRUPT 51)
  (SETQ *LEFT-BUTTON* (NOT (EVENP (REGISTER 1))))
  (SETQ *RIGHT-BUTTON* (NOT (EVENP (QUOTIENT (REGISTER 1) 2)))) ))

(DEFUN BUTTON-PRESS (LAMBDA (BUTTON)   ;Mouse status at last
  (REGISTER 0 5)                       ;button press
  (SETQ *BUTTON-PRESSES* (BUTTON-INFO)) )

(DEFUN BUTTON-RELEASE (LAMBDA (BUTTON) ;Mouse status at last
  (REGISTER 0 6)                       ;button release
  (SETQ *BUTTON-RELEASES* (BUTTON-INFO)) )

(DEFUN BUTTON-INFO (LAMBDA ()
  (REGISTER 1 (COND ((EQ BUTTON 'LEFT) 0) (1)))
  (INTERRUPT 51)
  (SETQ *LEFT-BUTTON* (NOT (EVENP (REGISTER 0))))
  (SETQ *RIGHT-BUTTON* (NOT (EVENP (QUOTIENT (REGISTER 0) 2))))
  (SETQ *MOUSE-ROW* (REGISTER 3))
  (SETQ *MOUSE-COL* (REGISTER 2))
  (REGISTER 1) ))

(RDS)

```

## **X T R A S T A R**

**Frank O'Brien** of Huntington, New York has added two new facilities to the muSTAR editor distributed with the '83 version of muLISP. A multiple file handler makes it possible to edit and then save function definitions occurring in several different source files. A directory display option has been added to the muSTAR menu that displays all .LIB files on the current MS-DOS directory. The following is his documentation:

**Multiple File Handler:** This facility makes it possible to handle multiple files in memory at one time and be able to keep track of what functions belong in what files. This is done by establishing global variables ACTIVEFILE and OTHERFILES. On the property of each of the file names in OTHERFILES is a list of all functions belonging to that file under the property 'SAVELIST. The functions belonging to the file ACTIVEFILE are contained in the list SAVELIST as before by muSTAR.

At any time you can set a new active file by invoking the "A" option on the muSTAR main menu. This will CONS the old active file and its SAVELIST to OTHERFILES. Note that the "A" option only accepts unique file names; no file extensions or drive names are allowed. (This could be a future improvement; see RD-FIL-NAM for how muSTAR handles this.)

When using the Read and Write options of muSTAR the prompt will accept three inputs (1) no input, meaning return to main menu, (2) a file name as before, and (3) the letter "A", meaning use the ACTIVEFILE for the read or write operation. The most recent file read in or written to from the muSTAR menu is already set to the active file. If you wish to specify a drive or file extension other than the default (muSTAR's default extension is LIB), that is OK so long as you use the MS-DOS naming convention (e.g. A:FOO.BOO).

If desired, the active file can be changed from the muLISP prompt or from within a function by the call

```
(APPLY (GET 'A 'EXECUTIVE) (LIST filename))
```

where <filename> is the name of the new active file.

**Directory Display Command:** The new "L" option in the muSTAR main menu displays a listing of all the .LIB files on the current MS-DOS directory. It is useful if you can't remember the name of a source file. To use this option, the COMMAND.COM file must be on the diskette in drive A. The display can also be invoked from the muLISP prompt or from within a function by the call

```
(APPLY (GET 'L 'EXECUTIVE))
```

; File: XTRASTAR.LIB      Ver 1.0      August 18, 1985      Frank O'Brien

```
(PUTQQ A EXECUTIVE (LAMBDA (FILENAME)
  ( ((NULL FILENAME)
    (TERPRI) (PRIN1 " Active File Name: ")
    (SETQ FILENAME (CAR (READLINE))) ) )
  ((NULL FILENAME) NIL)
  ( ((NONNULL ACTIVEFILE)
    (PUT ACTIVEFILE 'SAVELIST SAVELIST)
    (PUSH ACTIVEFILE OTHERFILES) ) )
  (SETQ ACTIVEFILE FILENAME)
  (SETQ SAVELIST)
  ((MEMBER FILENAME OTHERFILES)
    (SETQ OTHERFILES (REMATM1 FILENAME OTHERFILES))
    (SETQ SAVELIST (GET FILENAME 'SAVELIST))
    (REMPROP FILENAME 'SAVELIST) ) ))

(PUTQQ L EXECUTIVE (LAMBDA ()
  (EXECUTE "A:COMMAND.COM" "/C DIR *.LIB /P")
  (TERPRI) (PRIN1 "Hit Return When Ready")
  (READCH) ))

(SETQQ MENU$ (
  (F EDIT FUNCTION)                    (V EDIT VARIABLE)
  (P EDIT PROPERTY)                    (E EVAL LISP)
  (T TRACE FUNCTION)                   (U UNTRACE FUNCTION)
  (A SET ACTIVE FILE)                   (L LIST LIBRARY FILES)
  (R READ FILE)                        (W WRITE FILE)
  (X EXIT TO DOS) ))

(DEFUN RD-FIL-NAM (LAMBDA (NAM$ EXP$ NUM$)
  (SETQ NAM$ (CAR
    (QUERY$ "Enter a file name or an "A" for the active file")))
  ((NULL NAM$) NIL)
  ( ((EQ NAM$ 'A)
    (SETQ NAM$ ACTIVEFILE) )
    (APPLY (GET 'A 'EXECUTIVE) (LIST NAM$)) )
  ( ((EQ (SUBSTRING NAM$ 2 2) ':)
    (SETQ EXP$ (SUBSTRING NAM$ 1 1))
    (SETQ NAM$ (SUBSTRING NAM$ 3)) ) )
  ((SETQ NUM$ (FINDSTRING "." NAM$))
    (LIST (SUBSTRING NAM$ 1 (SUB1 NUM$))
      (SUBSTRING NAM$ (ADD1 NUM$)) EXP$) )
  (LIST NAM$ 'LIB EXP$) ))

(DEFUN REMATM1 (LAMBDA (ATM LST)
  ((EQUAL ATM (CAR LST))
    (CDR LST) )
  (CONS (CAR LST) (REMATM1 ATM (CDR LST))))

(SETQQ ACTIVEFILE)
(SETQQ OTHERFILES)
(RDS)
```