COMPUTATION CENTER
Massachusetts Institute of Technology
Cambridge 39, Massachusetts

November 5, 1964

TO: FMS Users

FROM: Lynda Korn and Marjorie Merwin-Daggett

SUBJECT: The Use of MAD in the MIT Version of the FMS System

MAD is a translator, that is comparable to Fortran, flexible versatile, and easy to learn. The Translation process is faster than that of Fortran though the object program may be less efficient and larger than the one produced by Fortran. In general, an efficient object program is only important when the program will be used repeatedly.

The MAD language, based on ALGOL 58, is similar to that of Fortran. A complete description is available in the MAD Manual, which is on sale at the Technology Store.

Before running with the Computation Center Fortran Monitor System, the following differences with regard to the MAD Manual should be noted:

A. Control Cards

To start a MAD compilation, a card with an asterisk in column 1 and "MAD" appearing between columns 7 and 72 is necessary.

No "Label" card is needed. Columns 2 through 6 of the first remark card (i.e. R in Column 11) are punched in Columns 73 through 77 of the binary deck which is automatically sequence numbered. Row binary cards cannot be requested.

No "List" card is needed.

B. The output subroutines are under sense switch control since the Fortran input-output routines are used. Thus the following two MAD statement pairs are equivalent:

READ BCD TAPE 4, X, List and READ FORMAT X, List

WRITE BCD TAPE 2, X, List and PRINT FORMAT X, List

The statement: PUNCH FORMAT X, list causes Hollerith cards to be punched on-line if sense switch 4 is down or written off-line if sense switch 4 is up.

C. The MAD translator automatically compiles for main programs a calling sequence for .SETUP which initializes the appropriate locations in core for (FPT), (F2PM) and (TIME).

D. To terminate a job, one should use the MAD statement EXECUTE EXIT. which transfers control to (F2PM) and then to the system. The DEC mode of (F2PM) should be used to dump integers. (MAD expects integers in the address; Fortran expects integers in the decrement.)

E. A complete description of Fortran and MAD format statements is given in Memorandum CC-186, which is available in Room 26-139.

F. Program Common Erasable and Equivalence Statements

In the MIT Version of MAD, Program Common and Erasable are both assigned upper memory storage locations. Only Program Common, though, may be preset by Vector Values statements. (When Program Common is preset by a Vector Values statement, the BSS loader of the Monitor uses tape B2 as temporary storage.)

A point of difference between Fortran and MAD occurs in the ordering of variables in Common storage. In Fortran, variables in Common are ordered in a manner dependent on both the Common and Equivalence statements. In MAD, the ordering is that of the Program Common statement solely.

Programmers should beware that ERASEABLE COMMON starts at 77461 while program COMMON is below this.

G. Chain Jobs

Within a MAD subprogram, the arguments for the CHAIN subprogram refer to the record number of the desired link and the "Chain Tape" number. For example,

EXECUTE CHAIN. (R,T), where T may be 1, 2, 3, or 4; these "Chain Tape" numbers refer to physical tapes: A4, B2, B3, and A4 respectively.

Thus, to call in link 407 from chain tape A4, the statement may be:

```
EXECUTE CHAIN. (407,1)
```

Since B2 is used by BSS for temporary storage of any information that is preset into Program Common by a Vector Values statement, tape B2 must not be used as a chain tape, but it may be used as a scratch tape during execution.

In a chain job, the Program Common of the first link only can be preset by Vector Values statements.

H. FMS Library

All subprograms on the FMS Library tape except scope subprograms, XSIMEQF and INDVF have been revised in such a way that they may be used by either MAD or Fortran compiled programs. A list of the subprograms on the library tape is available in the consulting Office, Room 26-153.

I. Fortran and MAD Subprogram Mixing

MAD subprograms may call FORTRAN subprograms if all parameters involve only one dimensional arrays and non-integer constants and variables. (Integers are excluded since Fortran places them in the decrement and MAD places them in the address.)

When one-dimensional array names are parameters, certain adjustments in the calling sequence must be made to conform to the conventions of each compiler's calling sequence. If an array A of N locations is specified in a Dimension statement, Fortran reserves N registers whereas MAD reserved N+1 registers. In Fortran, references may be made to A(1).....A(N); in MAD, references may be made to A(0).....A(N). In a calling sequence where only the array name is given, the A(1) element is implied by Fortran and the A(0) element by MAD. This difference should be accounted for in the bounds within the subprogram.

When dealing with two-dimensional arrays, the programmer must be more careful. Fortran stores arrays by columns and MAD stores them by rows.

A MAD subprogram calling a Fortran subprogram must set the dimension vector for each array in the call to agree with the array sizes specified in the Dimension statement of the Fortran subprogram. In addition, the call to the Fortran subprogram should specify the (1,1) element and not be just the array name. For example,

EXECUTE SUBR2. (A (1,1)) would be used to call SUBR2, that was written in Fortran.

For the cases of higher dimension arrays storage reversal of the arrays is required and there are several ways this can be handled. Some suggestions are:

1) Code the Fortran program as if the dimensions were MxL where in the MAD program they were LxM.

2) Flip the array before going to the Fortran program and reverse it upon return to the MAD program.

3) Arbitrarily remap the arrays using FAP subprograms between the Fortran and the MAD subprogram usages.

J. Statements

The following statements are not available:

SET LOW DENSITY TAPE	the library routines SETDLO and SETDHI
SET HIGH DENSITY TAPE	should be used instead
UNLOAD TAPE	special non library routine UNLOAD should be used
suffix: IF LOAD POINT TRANSFER TO ...	

K. Subroutines

The following subroutines are not available:

SETEOF.
SETETT.
SETERR.
SYSTEM.
ERROR.

The following subroutines are available in the non-SHARE files in Room 26-058.

ZERO.
SYMM.
TRANSP.

L. FAP and MAD Subprogram Mixing

One should bear in mind that MAD subprograms only save index register 4.

M. Variable Names

The variable name EXIT cannot be used in a MAD program.

N. The range of floating point constants in MAD is

$$.1469369 \times 10^{-38} \leq |F| \leq .1618 \times 10^{39}$$