APPENDIX 2

MASH COMMAND REPERTOIRE: SYNTAX AND SEMANTICS

Command Structure and Notation

MASH functions are invoked by the user through a series of commands. Each command directs the system to perform a specific action or set of actions. Execution of commands is generally immediate, but some commands are deferred pending the execution of another. Almost every MASH command begins with an imperative verb. All MASH commands are complete English sentences. Almost all commands may be preceded with a label, which is a 1 to 10 character name. If a label exists, the command will first be stored in the current user's dictionary with the label as its name, and the command will then be executed.

The general form of a MASH command is as follows:

* [label:] [commandstring] ; [comments]

Either an asterisk or an ampersand is the first character on the line and is printed by MASH as a *prompt* to signal the user that input text is expected by the system. In reproducing dialogue between the computer and user, we adopt the convention that text the computer types is underscored, while text the user types is not. The square brackets above indicate that their contents are optional and need not appear in the command. This is one of a number of syntactic notations that are generally used to define computer command syntax. The notation used in this appendix is the following:

Lower-case characters	represent information that must be supplied by the user. In most cases the characterization of the lower case symbol in the syntax definition suggests the type of information to be supplied.
Upper-case characters	represent words in the MASH lexicon that must be entered as they appear in the command syntax.
Braces { }	indicate that a choice must be made from the two or more lines enclosed.
Brackets [ ]	indicate an optional feature. The contents of the brackets are used according to the above rules if the feature is desired. If the brackets contain two or more options and one is underscored, the underscored option is the default condition in effect if the optional feature is not selected.
Ellipsis ...	indicates that the information preceding the ellipsis may be repeated at the user's option.

Following is an example of the use of the above notation for describing command syntax:

I WOULD LIKE TO { BUY / CHARGE } A [{ ONE WAY / ROUND TRIP }] { FIRST CLASS / ECONOMY }

TICKET [FROM city-1] TO city-2 [, PLEASE]

MASH commands are entered on the user's terminal in free form, and may be entered on as many lines as are required to specify the command. The first line of each command will be prompted by an asterisk * and subsequent lines will be prompted by an ampersand & to remind the user that the current command has not yet been terminated. The semicolon character ";" terminates a command. Command symbols may not be split between lines. A partially entered command may be aborted by typing the sequence "escape," "carriage return;" the escape key is often labelled "altmode" or "prefix." [1].

MASH uses a subset of the full ASCII code [U3] to define its symbols and operators. Figure A2-1 displays the characters that may be used to form MASH names and symbols. Lower case alphabetic letters are folded into their upper case equivalents when they are entered. MASH symbols may contain any number of characters from the symbol alphabet, but characters beyond the 10th are truncated. Figure A2-2 displays the characters used as special symbols for algebraic operation and for punctuation; these are also referred to as separator characters.

```
A B C D E F G H I J K L M N O P Q R S T U V W X
Y Z   a b c d e f g h i j k l m n o p q r s t u v w x y z
0 1 2 3 4 5 6 7 8 9 0 .
- (except in algebraic/boolean expressions)
```

Figure A2-1. MASH Symbol Alphabet.

```
+  -   *   /   <   >   ,   ;   :   @   [   !
"  # $ %   &   '   (   )   _   =   ]   ^   ?
space
```

Figure A2-2. MASH Separator Alphabet.

[1] The more basic DECsystem-10 monitor system rules for cancelling input either by character (DEL) or by line (ctrl-U) also apply to MASH. The added rule for cancelling an entire command must be part of MASH, since a multi-line command is not recognizable to the monitor system as a single entity.

Following is a list of MASH commands, ordered alphabetically by command name or key word. An example of the use of the command appears with each command.

Command	Example
ADD	REGION, URBANCODE TO LIST DEMOGRAPH;
ALLOW	BIRTHS, DEATH, MARRIAGE;
ALTER	GNP IN 1969 TO 6.75;
ASSOCIATE	UNEMP WITH HISTORIC U, ...
BROWSE	THROUGH POPULATION 34;
CANCEL	WATCH OF SUCCESSFUL BIRTHS, DEATHS;
CHANGE	WAGES OF PERSON 4 to 5000;
CHECKPOINT	AFTER 1964(P), 1965 THROUGH 1969(P);
CLEAR	TRACKING ARRAY, ASSOCIATION ARRAY;
*COMPUTE	REGRESSION OF INCOME ON AGE, ...
CONDENSE	DICTIONARY;
CONDUCT	SAMPLE SURVEY CPS OF ...
COPY	SMITH'S EQUATION Al NAMING IT A3;
CREATE	POPULATION USING SAMPLE RURAL;
DEFINE	LIST EXTRACT AS NAME1, NAME2, ...
DELETE	GRADE, CLASS FROM LIST SCHOOL;
DEPOSIT	INTEGER 5 IN FAMAD(1,3);
DISCARD	ATTRIBUTES OCCUP, MILITARY, ...
DISPLAY	SMITH'S EQUATION DEMAND;
END	MASH;
ERASE	LIST DEMOG, COMMANDS X, Y, Z;
EXAMINE	FAMAD(1,3) INTEGER;
EXECUTE	READY, SET, GO;
EXHIBIT	STRUCTURE OF FAMILY 45;
EXPORT	TO PLANETS SERIES LIST RESULTS;
EXTRACT	FROM SURVEY FILE 2500P ...
FIND	FIRST PERSON WITH WAGES > 10000;
GENERATE	HIGHEST 4 YEAR SERIES FOR WAGES, ...
IDENTIFY	WEDSTATE AS MARRIED, ...
INCLUDE	SURVEY ATTRIBUTES AGE, SEX, RACE, ...
LOOK	AT PERSON 462;
OBTAIN	ATTRIBUTES AGE, RACE, SEX;
ORIGINATE	DICTIONARY ...
PREPARE	MACROMODEL SAMUELSON FOR PASS 3;
PREVENT	INCEST, DEATH, WORKING;
PROCEED	THROUGH NEXT PERSON;
PRODUCE	SURVEY FILE Sl AND CODEBOOK FILE Cl;
PROTECT	SERIES NEWGNP, CODE EDUCODE, ...
RENAME	CODE WAGECODE AS MONEYCODE;
REPLACE	JOB BY INDUSTRY IN LIST ...
REPLICATE	PERSON HAVING PERNAM = 257 ...
REPORT	STATUS OF POPULATION, SIMULATION;

```
SAVE          POPULATION;
SEARCH        PARAMETER LIST SMITH'S BASICPARS;
SET           BASE YEAR TO 1959;

SHOW          MACROMODEL TRACKING, ASSOCIATIONS;
SIMULATION    POPULATION NAME IS 59;
START         DICTIONARY FOR HARVEY;
SUMMARIZE     BIRTH, DEATH, MARRIAGE YEARLY;
*TAKE         CENSUS LABORFORCE YEARLY;

TERMINATE     SIMULATION AND CHECKPOINT;
TITLE         EXPERIMENT: 30% REDUCED WORKADJ;
TRACK         UNEMP FROM 1969 TO 1970 USING ...
TURN          ON OPTIONS 3, 8, 25, 63;
TYPE          SERIES GNP, UNEMP, BIRTHS, ...

UNPROTECT     SERIES NEWGNP, LISTS A, B, C,
USE           SMITH'S DICTIONARY;
WATCH         THWARTED BIRTHS, DEATHS,
WHAT          IS AGE OF PERSON 692;
WHERE         IS FAMILY 619?;

WHICH         PERSON LIVES AT ADDRESS 2851?;
WHO           IS THE CURRENT FAMILY?;
```

*Not yet implemented.

The current set of commands may be divided according to function into the following categories: (1) commands that use and maintain entities in the dictionary; (2) commands used to create a micropopulation for simulation experiments; (3) commands for browsing in a micropopulation; (4) commands that produce micro and macro outputs of simulated states; (5) commands for defining and manipulating aggregate models; (6) commands for executing and controlling simulations; and (7) MASH system control commands. In the table below, the MASH commands listed above have been assigned to these categories. A brief description of its function is associated with each command.

1. Dictionary Commands

ADD	inserts names into existing lists
ALTER	modifies existing time series
CONDENSE	removes deleted entities from the dictionary
COPY	duplicates existing entities
DEFINE	creates entities of all types
DELETE	removes names from existing lists
DISPLAY	types dictionary entries, status, etc.
ERASE	removes entries from the Dictionary
ORIGINATE	used once per system to create Dictionary
PROTECT	gives Dictionary entries erase protection
RENAME	changes name of a Dictionary entity
REPLACE	replaces a name in an existing list
START	defines a new Dictionary owner

UNPROTECT	removes entity erase protection
USE	specifies default or current Dictionary owner

2. Micropopulation Creation Commands

CREATE	triggers population creation process
DISCARD	discards attributes previously included
EXTRACT	specifies survey and codebook file names
GENERATE	creates micro time series attributes
IDENTIFY	renames population attributes during creation
INCLUDE	specifies which attributes are selected
REPLICATE	selects an individual for replication
SET	specifies base year and population name

3. Micropopulation Browsing Commands

BROWSE	specifies browsing population
CHANGE	alters micro entity attribute value
EXHIBIT	displays history, structure of micro entity
FIND	searches for an entity with specific attributes
LOOK	sets a browsing pointer explicitly
WHAT	displays micropopulation attribute values
WHERE	gives the address of a specific entity
WHICH	names the entity at a specific address
WHO	gives the name of the current entity

4. Output Commands

*COMPUTE	defines and executes a statistical or summary output procedure
CONDUCT	takes a survey of a micropopulation
EXPORT	prepares a time series data file for analysis programs
OBTAIN	names attributes to be included in a survey
PRODUCE	names survey data and codebook files
*TAKE	schedules and initiates population censuses

5. Macromodel Commands

ASSOCIATE	binds macromodel variable to time series
CLEAR	initializes macromodel tracking tables, etc.
PREPARE	initializes macromodels for a simulation run
SEARCH	initializes scalars in a macromodel
SET	sets default page width or time period
SHOW	displays various macromodel tables
TRACK	names exception sources for exogenous variable values
TYPE	displays time series from databank

6. Simulation Control Commands

ALLOW	enables disabled operating characteristics
PREVENT	disables operating characteristics
PROCEED	initiates or continues simulation process
SIMULATION	names the micropopulation for simulation
SUMMARIZE	requests yearly or final summary output

TERMINATE	triggers completion of simulation run
WATCH	requests individual entity results

7. System Control Commands

CANCEL	cancels checkpoints, summaries and micro outputs
CHECKPOINT	schedules or invokes checkpoints
DEPOSIT	stores data in virtual memory cell
END	terminates MASH run and returns to Monitor
EXAMINE	displays contents of a virtual memory cell
EXECUTE	fetches and executes stored commands
REPORT	displays job, simulation status, etc.
SAVE	closes population files
TITLE	specifies title for protocol page heading
TURN	turns console or options on and off

* not yet implemented

The remainder of this appendix contains detailed syntactic and semantic descriptions of each MASH command. Descriptions of the commands are ordered alphabetically.

ADD

The ADD command is used to add names to dictionary entries that are lists of names. The general format of the ADD command is:

ADD name1 [, name2, ...] TO { LIST / CENSUS / PROCEDURE / TABLE / MACROMODEL } entityname [{ AT BEGINNING / AT END / BEFORE name / AFTER name }] ;

The acceptable dictionary entity types for the ADD command are contained above. Unless otherwise specified, new names are added to the end of the existing list of names. The BEFORE and AFTER options of the command make it possible to insert one or more names into the middle of an existing list. One or more names may be added, so long as the length of the dictionary entry buffer (150 names) is not exceeded. The rules for constructing names are explained in the description of the DEFINE LIST command.

If the buffer size is exceeded, or if the command references a name not found in the list-like entry, the command is not executed. Otherwise, the command is executed and the augmented entry supersedes the previous entry.

ALLOW

The ALLOW command re-enables previously disabled operating characteristics and other functions in an existing micro model. The syntax of the command is:

ALLOW opchar1 [, opchar2, ...] ;

The operating characteristic(s) specified must be among those which are included in the micromodel being simulated. The operating characteristics of the current Urban Institute Microanalytic Model may be ALLOWed with the following names, where parentheses enclose alternate names:

BIRTH (BIRTHS)	DEATH (DEATHS)
MARRIAGE (MARRIAGES)	AGING
STERILITY	DIVORCE (DIVORCES)
EDUCATION	WEDDING (WEDDINGS)
INCEST (INBREEDING)	WORKING
MOBILITY (MIGRATION)	WEALTH
LEAVING	

The ALLOW command operates by resetting options that have been set by the PREVENT command. The same options can be set or reset with the TURN command, but the ALLOW and PREVENT commands are preferred since they are self-documenting.

Initially in each simulation exercise all operating characteristics are ALLOWed (enabled).

ALTER

The ALTER command is used to either alter current values of a time series in the dictionary or extend a series in either direction by adding more values to it. The general format of the ALTER command is:

ALTER [SERIES] seriesname { FROM year1 THROUGH year2 / IN year1 } TO value1 [, value2, ...];

The value of year2 + 1 must not precede the initial year for which the time series is defined, nor can the value of year1 be more than one year beyond the last year for which the series is defined, i.e. the series cannot have gaps of undefined values. The value of year2 must be at least as great as year1, and must be

sufficiently small so as to keep the altered series within the dictionary entry buffer, which currently can hold 150 time series values. Years correspond to calendar years, e.g. 1975.

One or more values may be entered to alter existing or new entry slots in the series. The number of values must be consistent with the number of years specified. The word THROUGH implies "starting with year1 and going through year2, inclusive."

ASSOCIATE

The ASSOCIATE command is used to associate, or bind, any time dependent (subscripted) macromodel variable with any time series in the dictionary. The general form of the ASSOCIATE command is as follows:

ASSOCIATE variable1 WITH [{ HISTORIC / owner'S }] series1

[, variable2 WITH [{ HISTORIC / owner'S }] series2, ...] ;

Associations are possible between macromodel variables and time series belonging to other users if the variables are exogenous in the model, since by definition such variables do not appear on the left side of an equation. An endogenous macromodel variable should not be associated with a time series belonging to another user since equation results must be stored, and MASH does not allow one user to modify another's dictionary.

More than one association may be specified with one command, and multiple ASSOCIATE commands may be executed for a given simulation run. Furthermore, new associations may be specified during pauses in a simulation run. The last recently entered applicable association (if any) will be used.

The TRACK command is closely connected with the ASSOCIATE command, and it may also be used to specify the source for values for macromodel variables. The difference between tracking and association is that the latter specifies both the source and the destination for macromodel variable values, while the former can specify only the source. Additionally, a tracking specification can be restricted to a given simulation year or years, while an association specification cannot. Lastly, the TRACK command permits the use of the console as a source for macromodel variable values, while the ASSOCIATE command does not.

MASH always uses the most recently entered applicable association or tracking instruction, if any. A relevant tracking specification will override an association specification for the same variable. Since tracking applies only to the source for macromodel variable values, the calculated values for a macromodel variable will always be stored in the time series with which it is associated. Required values will be taken from the associated time series also, unless a tracking specification applies to the variable for the year in question.

BROWSE

The BROWSE command is used to specify and open a micro simulation population for on-line browsing. The general format of the BROWSE command is:

BROWSE THROUGH POPULATION popnumber ;

Populations are numbered from 1 through 99; any population which has been previously created and whose 10 data files reside in the current user's area may be opened for browsing. The BROWSE command opens these files for browsing. If a population other than the one specified in the command is currently open, it is first closed, and the population requested is opened for browsing. If the population specified is currently open -- either as a result of prior microsimulation activity or prior browsing -- the command is ignored.

Three "browsing" pointers are established during every browsing activity; each one points to a specific INTUNIT, FAMILY, and PERSON in the population. Initially these pointers point to no one. Specific browsing commands alter the values of these pointers. The browsing pointers are independent of the microsimulation pointers, so that it is possible to browse through a micropopulation during microsimulation without generating any logical inconsistencies.

CANCEL

The CANCEL command is used to cancel the effect of WATCH, SUMMARIZE or CHECKPOINT commands issued previously. The syntax of the CANCEL command is:

```
                  { WATCH OF  }  [ {ALL       } ]
                  { WATCHING  }  [ {SUCCESSFUL} ]  opchar1 [ opchar2 ... ]
                                 [ {THWARTED  } ]
                  { YEARLY    }
                  { FINAL     }  SUMMARY OF opchar1 [ opchar2 ... ]
CANCEL {
       IMPLICIT CHECKPOINTS
       CHECKPOINT[S] AFTER [YEAR[S]] year1 [ year2 ... ]
                                                                    } ;
```

The first two forms of the CANCEL command reset the options set by WATCH and SUMMARIZE commands. Thus, the CANCEL command can suppress both the output of data describing micro events and final or yearly summaries. The options involved can be set or reset with the TURN command, but the WATCH, SUMMARIZE and CANCEL commands are preferred because they are more self-documenting. The operating characteristics which may be used with these commands are a subset of those listed in the description of the ALLOW command. The user is informed if he attempts to apply one of these commands to an operating characteristic for which the command has no meaning.

The second pair of CANCEL commands is used to unschedule checkpoints. "Implicit" checkpoints are those that are to take place after a specified interval; they are scheduled with a command like "CHECKPOINT AFTER EVERY 5 YEARS;".

CHANGE

The CHANGE command is used to alter the value of an attribute of an entity in the micropopulation. The general format of the CHANGE command is:

```
CHANGE attributename [ OF {INTUNIT} n ] TO value ;
                         {FAMILY  }
                         {PERSON  }
```

The attribute name used must be included in the micropopulation's codebook, with the exception of micro time series attributes which can be referenced with subscripts (see description of the WHAT command). The value must be included in the set of values associated with the attribute in the codebook. Only integer values may be entered.

If the OF clause is omitted, the current entity at the level at which the attribute is defined is selected as the one to be changed, provided that the browsing pointer at that level points to an existing entity. If the OF clause is included, the browsing pointer for the entity level specified is reset to point to entity "n" at

that level, provided that the attribute named is defined at that level and the entity named exists in the micropopulation. Thus, if several changes are to be made to one micro entity, the OF clause needs to be specified in only the first of several CHANGE commands.

The value entered is checked in the population codebook for validity. If the value is defined, then the corresponding value label from the codebook is typed on the user's console to verify the change. If the value is not defined for the attribute named, the user is asked whether he wishes the list of valid value specifications to be typed for him. He must then re-enter the command with a legal value if he wishes to effect a change.

CHECKPOINT

The CHECKPOINT command is used to schedule, unschedule or invoke checkpoints.[2] The general form of the command is as follows:

CHECKPOINT [INTO project,programmer]

```
{ NEVER
{ NOW [(option)]
{ AFTER EVERY [n] YEAR[S] [(option)] [ STARTING { NOW } ]
                                     [         { IN year } ]
{ AFTER [YEAR[S]] year1 [(option)]  [THROUGH year2 [(option)]  [  year3  ... ] ]
};
```

In the above syntax, "project,programmer" defines a PDP-10 disk area where the checkpoint files are to be written. The project-programmer number is *not* enclosed in square brackets. The "year" of a checkpoint is the calendar year being simulated, e.g., 1964. "Option" distinguishes between temporary and permanent checkpoints, and is either the letter "T" or the letter "P" and must be enclosed in parentheses. If "option" is not specified, (T) is assumed.

The command "CHECKPOINT NOW;" may be used any time that MASH is at command level to invoke an unscheduled or immediate checkpoint. MASH will write a copy of its own core image on disk and prompt for another command. When the file produced is run, MASH will again prompt for a command with an asterisk. If simulation is in progress at the time such a command is given, the micropopulation is

---

[2] The CHECKPOINT command and associated processing were designed and implemented by Kenneth Jacobs. It is necessary to include this function within MASH since the PDP-10 operating system does not provide a checkpoint facility for general user jobs.

closed, and is re-opened again when the run is restarted. Simulation can then be continued with the PROCEED command.

Using the last form of the CHECKPOINT command, checkpoints may be scheduled to take place automatically after specific years have been simulated. Lists and/or ranges of years may be specified in any combination. If a temporary or permanent option is specified, it applies only to the checkpoint after the year the option immediately follows, unless the option follows *both* the first and last years of a range. Thus, the command

```
CHECKPOINT AFTER 1960, 1961(T), 1962(P), 1963 THROUGH 1967(P),
                        1968(P) THROUGH 1970(P);
```

sets up temporary checkpoints after 1960, 1961, 1963, 1964, 1965 and 1966. Permanent checkpoints are set up for 1962, 1967, 1968, 1969 and 1970.

An end-of-year checkpoint, whether scheduled or unscheduled, causes the following actions: (1) the micropopulation in use, if any, is closed, updating the population status file; (2) the core image of MASH is written to the disk in save-file format; (3) the "MASH.TMP" file containing the current writeable copy of all non-resident links in use is copied; (4) the ten population files are copied; (5) the population is re-opened; (6) and simulation continues with the next year. (MASH exits to the Monitor after step (4) in the case of an unscheduled end-of-year checkpoint.)

A checkpoint may be taken at the end of simulation with the TERMINATE command. This avoids copying the population, yet does provide a runnable core image for possible later continuation of the run and a copy of the ".TMP" file.

Previously scheduled checkpoints may be cancelled with the "CHECKPOINT NEVER;" command, which unschedules all checkpoints, or with the "CANCEL CHECKPOINT ..." command. A checkpoint specification may be superseded with a CHECKPOINT command specifying the same year as an existing checkpoint.

The "CHECKPOINT AFTER EVERY ..." form of this command has not yet been implemented.

CLEAR

The CLEAR command is used to clear macromodel specification tables (arrays). The general form of the command is:

CLEAR arrayname ARRAY [, arrayname ARRAY ...] ;

where *arrayname* is one of the following: EQUATION, POLISH, SCALAR, ITERATION, ASSOCIATION, TRACKING, CONSTANT, MACROMODEL, or ALL.

The CLEAR command zeroes the count of entries in the corresponding tables. See the description of the SHOW command for an explanation of the content and format of the various macromodel arrays.

This command is primarily for use in debugging the MASH macromodel link, although it can be used in developing macromodels from the command level. It is recommended that the command be used sparingly, since unpredictable results may result if the integrity of a macromodel is not preserved.

COMPUTE

The COMPUTE command is used to initiate a statistical or summary output procedure in which the input data will be obtained from the current micropopulation. The general form of the COMPUTE command is:

COMPUTE procedurename OF dataphrase [WITH OPTIONS option1 [, option2 ...]]
[ON SAMPLE samplename] ;

The list of procedure names which may be invoked by the COMPUTE command are TABULATION, REGRESSION and FREQUENCY. The data phrase syntax for each of the above procedures is specific to that procedure, as is the list of option words that have meaning to the procedure. The ON SAMPLE phrase may be appended to any COMPUTE command.

In order to minimize computing time spent reading large micro data files, it is often desirable to satisfy several COMPUTE requests in parallel. This may be accomplished by defining several COMPUTE commands in the dictionary and then defining a CENSUS to consist of the names of those commands. The TAKE CENSUS command may then be used to execute these computations in parallel.

The COMPUTE command is not yet implemented. Precise descriptions of the data phrases and option words for the above procedures will be formulated during the implementation process.

CONDENSE

The CONDENSE command is used to move the unused space that is created in the MASH dictionary when entries are erased or are superseded by entries that overflow the space originally assigned. The syntax of the CONDENSE command is:

CONDENSE DICTIONARY;

The MASH command dictionary is condensed periodically by the person responsible for maintenance of MASH. The dictionary remains open during a condensing operation, so that condensing may be performed even if other users are concurrently using MASH. However, any attempt on their part to access the dictionary until the condensing operation is completed will be met with the message "DICT ARRAY IN USE - PLEASE WAIT". This is a privileged command, available only to the individual responsible for maintaining the MASH system.

CONDUCT

The CONDUCT command triggers the process of taking a sample survey of a micropopulation and produces either a rectangular or a hierarchically structured data file described by the contents of an associated machine readable codebook file. The syntax of the CONDUCT command is:

CONDUCT SURVEY [OF { { PERSONS / FAMILIES / INTUNITS } [IN POPULATION n] / POPULATION n }]

[USING SAMPLE samplename] [AFTER SKIPPING n RANDOM NUMBERS] ;

The CONDUCT command should be executed after all other commands defining the survey have been completed.

The first phrase in the command specifies the type of surveys to be taken. If an entity level is named, then the survey file will be rectangular, and the units of enumeration of the survey will be occurrences of the entity named. If no entity level is named, then the survey results in a 3-level hierarchically structured file in which the units of enumeration will be interview units, in which families

will be nested and in turn in which persons will be nested. If a population is named, that population is surveyed; if no population is named, the population that is currently open for simulation or browsing is surveyed.

The USING SAMPLE suffix phrase may refer to a sample defined at any level. If the survey is rectangular and the sample is defined over a higher level than the survey level, all entities at the survey level whose containing entity meets the sample requirements will be included. Similarly, if the survey is rectangular and the sample is defined at a level below the survey level, all entities whose first contained entity meet the sample requirement will be included. If the survey is hierarchical, the sample condition is applied to the attributes in the interview unit, the first family in the interview unit and the first person in that family in order to determine whether the interview unit is to be included in or excluded from the survey.

The AFTER SKIPPING clause may be used to generate pseudo-independent streams of random numbers to be used in any attribute or sample definitions with stochastic components. The CONDUCT command uses its own random number generator whose initial seed is independent of any and all preceding simulation and survey activity.

The CONDUCT command provides a way of obtaining a well-documented rectangular data file as a micro data output from a microsimulation run. This micro data file may be used as input to a variety of data summarization and statistical analysis programs both on the PDP-10 and, with the use of code translation programs, on other computer models. Alternatively, the CONDUCT command may be used to restructure a population that has been advanced through simulated time into a hierarchical survey file, with an accompanying codebook. This file could then be used as a source file for further MASH micropopulations.

COPY

The COPY command creates a copy of any entity in the MASH dictionary. The syntax of the COPY command is:

COPY [{ HISTORIC / owner'S }] entitytype entityname [NAMING IT newname] ;

If the owner's name is not specified, the entity to be copied is assumed to belong to the current owner. If the NAMING IT phrase is not included in the command, the new entity is assigned the same name as the old entity.

If the new name is the name of an existing entity of the same type as the copied entity, the existing entity will be superseded unless protected. If it is protected, an error message is typed for the user and no change is made to any existing entity.

CREATE

The CREATE command triggers the creation of an initial micropopulation for simulation, whose characteristics have been specified by preceding SET, INCLUDE, GENERATE, DISCARD, IDENTIFY, REPLICATE, and EXTRACT commands. The syntax of the CREATE command is:

```
CREATE POPULATION n [USING SAMPLE samplename]
        [AFTER SKIPPING n RANDOM NUMBERS] ;
```

The population's initial (or base) year must have been previously declared with a SET BASE ... or SET INITIAL ... command. The population's name, *n*, must be an integer between 1 and 99; it is specified within the CREATE command.

If the USING SAMPLE phrase is invoked, the sample definition must obey certain restrictions. The sample defined must be at the interview unit level, and it must not contain any computed attributes or recoding operations. These restrictions can be circumvented to some extent through use of specially coded subroutines in the population creation link.

The AFTER SKIPPING clause may be used to generate pseudo-independent streams of random numbers to be used in any stochastic components in the sample definition and in the initializing operating characteristics. The CREATE command uses its own random number generator whose initial seed does not affect any subsequent stochastic computations.

The invocation of the CREATE command triggers a sequence of operations that (1) search the survey codebook for attributes to be extracted from the survey file; (2) search the attribute library for attribute definitions; (3) create the initial simulation population's codebook; (4) extract the population data from the survey data file and reformat it; (5) initialize library attributes to zero; and (6) perform an initialization pass with initializing operating characteristics over the micropopulation. At each step, a message is typed informing the user of the progress of this operation.

DEFINE ATTRIBUTE, SAMPLE

The DEFINE ATTRIBUTE and DEFINE SAMPLE commands are used to create derived algebraic and boolean results from attributes of micropopulation entities. The syntax of the DEFINE ATTRIBUTE and DEFINE SAMPLE commands is:

DEFINE { ATTRIBUTE / SAMPLE } entityname OF [INTUNIT / FAMILY / PERSON] AS algebraic/boolean expression;

Attributes and samples consist of mixed algebraic and boolean expressions whose domain is -- in general -- a set of attributes and literal values, and whose range is the real number system for attributes and TRUE or FALSE for samples.[3] The domain of attribute and sample definitions may include attributes in a micropopulation, attributes which are computed and whose definitions are also stored in the MASH dictionary, and possibly other dictionary entities, such as codes. The DEFINE command does not by itself cause a value to be associated with an attribute; rather it describes how such a value is to be calculated. In this sense it describes virtual data that is only realized when the attribute is named in an output command.

Both attributes and samples are associated with a specific entity level, i.e. they are defined as attributes or samples of interview units, families or persons. Their definitions must include only attributes from the level of definition or from higher levels in order to avoid ambiguity of computation. [4]

Expressions that define attributes and samples may be written in free form with operators and operands intermixed and spread out over multiple lines. No replacement operator appears in the definition; the word AS performs that function. Contrary to Fortran tradition, the "=" sign appears as a synonym for the .EQ. relational operator.

The operators represent a mixture of Fortran and Algol usage, with some redundancy provided for ease of use. The operator BEGIN appears implicitly at the beginning of every expression, and the semicolon operator terminates the scan. No differentiation is made between subscripts and functions in notation, and subscripts are regarded as mappings from the domain of integers to a fixed set of values in the array. The Algol IF ... THEN ... ELSE sequence has been adopted, partly to compensate for not having a

---

[3] In fact, the evaluation of samples may rely upon the host language's boolean representation; for PDP-10 Fortran IV, TRUE is represented by a full word whose contents are -1 (36 l-bits) and FALSE is represented by a word whose contents are 0 (36 0-bits).

[4] Such a restriction is not desirable, since one often wants to aggregate lower level attributes at a higher level or otherwise use lower level information. For an exploration of the syntax required to accomplish such an extension, see Kidd [K2].

statement oriented programming language within MASH. The words ELSE and OTHERWISE are semantically identical and represent a concession to the MAD language. Unary minus is represented explicitly in the precedence table, but notationally it is represented as "-" just like binary minus.

Precedence Level	Operator	
0	(	BEGIN
1	;)	,
2	IF ELSE	THEN OTHERWISE
3	.OR.	
4	.AND.	
5	.NOT.	
6	.EQ. .NE. .LT. .LE. .GT. .GE.	= /= < <= > >=
8	+ UNARY -	-
9	*	/
10	**	∧
11	CODED.BY	(functions)

Table A2.1. MASH Algebraic and Boolean Operator and Precedence Levels

The CODED.BY operator binds a numeric attribute on the left with a code definition on the right; the result is a discrete mapping into that finite set of integers which is the range of the code function. This operator is special to MASH, but may be used like any other in forming an expression.

The full set of operators and their hierarchies or precedence levels used by MASH appears in the above table, although certain statement forms restrict the use of certain operators or other subsidiary dictionary references. The scanning of the expressions is performed from left to right, using an extension of the algorithm in the Burroughs Compilogram [B8]. The entities are stored in Polish string form, and are interpreted at execution time.

The functions available within MASH for defining attributes, samples and equations are listed and described in table A2-2.

ABS (X)	returns the absolute value of the argument X
ABSOLUTE (X)	is synonymous with ABS (X)
AVG (A,B,...)	returns the average of its arguments. The argument list may be of variable length.
AVERAGE (A,B,...)	is synonymous with AVG (A,B,...)
EXP (X)	returns *e* raised to the power X, or e^X.
EXP10 (X)	returns 10 raised to the power x, or 10^X.
FIRST (N)	is used to select a population sample from the beginning of a survey file. It returns a *true* result if the interview unit sequential number is <= N and a *false* result otherwise. This function is meaningful only in sample definitions that are used in a CREATE POPULATION command.
GOTO (NAME)	causes a transfer of control within a macro submodel to the equation having the name NAME. No result is returned. Use of this function is legal only in equation definitions.
LOG (X)	returns the natural logarithm of X
LOG10 (X)	returns the logarithm of X using the base 10.
MAX(A,B,...)	returns the maximum value of its arguments. The argument list may be of variable length.
MAXIMUM(A,B,...)	is synonymous with MAX(A,B,...)
MIN (A,B,...)	returns the minimum value of its arguments. The argument list may be of variable length.
MINIMUM(A,B,...)	is synonymous with MIN (A,B,...)
RANDOM (X)	returns either a *true* or *false* result. A random number R is drawn from a uniform distribution over (0,1), i.e. f(Z)=l for 0<=Z<=l and f(Z)=0 elsewhere. If R<=X, the function returns a *true* value; otherwise, *false* is returned. It follows that the function is *false* for all negative X and *true* for all X>=1.
RNORM (M, V)	returns a random number drawn from a normal distribution with mean M and variance V.
SQRT (X)	returns the square root of X.
UNITS (M, N)	is used to subselect population units from a survey data file. It returns the value *true* if the sequential number of the interview unit, S, is such that M<=S<=N; otherwise it returns the value *false.* This function is meaningful only in sample definitions that are used in a CREATE POPULATION command.

Table A2-2. MASH Function Table

DEFINE CODE

The DEFINE CODE command defines a recode mapping from one attribute into another. Its general syntax is:

DEFINE CODE codename AS (codephrase1/ ... / codephrasek) ;

Each "codephrase" consists of one or more discrete values or closed intervals of the real line which are to be mapped into another real value. The general syntax of each code phrase is:

[Vl [,V2, ...]] [V3-V4 [,V5-V6, ...]] [=w]

where the V's are real numbers specifying the domain of this subset of the mapping and *w* is the single real value which is the range of this subset of the mapping. A single value, such as Vl, represents a mapping of Vl into w, while an interval mapping such as V3-V4 represents a mapping of the closed interval [V3,V4] into w. If w is omitted, it defaults to i, where i is the number of the code phrase in the command; w defaults to 1.0 for the first code phrase, to 2.0 for the second, etc. If the symbol "*" appears in an interval mapping specification, it represents negative infinity if it appears on the left side of the dash and positive infinity if it appears on the right side of the dash. For example, the following command defines a recode mapping which might be used to categorize an individual's level of education:

DEFINE CODE EDUCODE AS (*-6=1 / 7,8=2 / 9-11=3 / 12=4 /13-15=5 / 16=6 /
17-20=7 / 21-*=8);

The code defined above, EDUCODE, might be used to define a "quantized" education attribute, as follows:

DEFINE ATTRIBUTE EDULEVEL OF PERSON AS EDUCATION CODED.BY EDUCODE;

The attribute EDULEVEL would be a person attribute (which could be evaluated while CONDUCTing a sample survey) whose range of values would be the interval [1,8].

Codes are interpreted sequentially by code phrase when the code is applied in an algebraic or boolean evaluation. Thus, even if the mapping is not consistent as a set of code phrases, it will be executed

consistently; the first code phrase whose domain includes the value to be coded will yield the result of the coding operation and the code definition scan will terminate.

DEFINE COMMAND

The DEFINE COMMAND command is used to enter MASH command strings into the MASH dictionary without executing them. The syntax of the DEFINE COMMAND command is:

DEFINE COMMAND commandname AS commandstring;

where "commandstring" consists of any MASH command *except* a TITLE command or another DEFINE command of any type. The command is entered as if it were to be executed, except that it is parsed and stored in the user's dictionary as an entity of type "command" with name "commandname".

The above command is identical with the command:

commandname: commandstring;

except that the latter labelled command both stores the command in the user's dictionary *and* executes it immediately, whereas the DEFINE COMMAND form only stores the command string in the user's dictionary.

DEFINE COMMAND is useful for defining procedures or census-type computations that are to be invoked subsequently by an EXECUTE command. It may also be useful for setting up background batch runs and for establishing libraries of user commands.

DEFINE EQUATION

The DEFINE EQUATION command is used to enter one macromodel equation into the MASH dictionary. The syntax of the DEFINE EQUATION command is:

DEFINE EQUATION equationname AS endogenousvariable [(T[±k])] :=
algebraic-expression-in-predetermined-variables ;

The Algol replacement operator ":=" denotes the replacement of the value associated with the variable on the left hand side of an equation with the result that is computed from the right hand side.

Equations consist of algebraic expressions that define the value of an endogenous variable which is either a time series variable or a scalar variable. If the subscript (T) follows the variable name in the equation, the variable defined is a time series variable. The subscript T may be offset by an integer *k* in either direction so that values in either past or future time periods may be defined.

The expression on the right side of the equation is a combination of predetermined variables, parameters whose values have been assigned, and constants. Time series variables in the expression must have their time period noted, either absolutely, e.g. GNP(1967), or relatively, e.g. GNP(T-l). Any of the operators or functions in the tables included in the description of the DEFINE ATTRIBUTE command may be used except CODED.BY, which means that code definitions are not allowed in macromodels. The GOTO function, however, may be used to program either conditional or absolute transfers of control among evaluation of equations constituting a macromodel.

DEFINE LIST, CENSUS, MACROMODEL, PROCEDURE, TABLE

The DEFINE LIST command is used to define a list within the MASH dictionary. A MASH list is a non-null ordered sequence of names or symbols. Other MASH dictionary entity types which are also composed of an ordered list of names and which are syntactically indistinguishable from lists except for their type are CENSUSes, PROCEDUREs, TABLEs, and MACROMODELs. The general syntax for defining these entity types is:

DEFINE { LIST / CENSUS / PROCEDURE / TABLE / MACROMODEL } entityname AS name1 [, name2 ...] ;

Each name consists of from 1 to 10 characters from the MASH symbol alphabet. Lower case letters are converted to upper case equivalents.

Although the MASH syntax for defining LISTs, CENSUSes, PROCEDUREs, TABLEs and MACROMODELs is syntactically identical, the statements differ semantically. A LIST consists of an ordered collection of names of several types, but a CENSUS consists of a list of output procedures that are to be invoked at certain times during a simulation. A PROCEDURE consists of a list of names of commands; a TABLE consists of an ordered list of aggregate time series; and a MACROMODEL consists of an ordered list of equations which define an aggregate economic model.

DEFINE SERIES

The DEFINE SERIES command is used to define an aggregate time series to be entered into the MASH dictionary. The syntax of the DEFINE SERIES command is:

DEFINE SERIES seriesname { FROM year1 THROUGH year2 / IN year1 } AS value1 [, value2 ...] ;

The time period clause may specify a time interval of 1 or more calendar years, i.e. year2 must be at least as large as year1. Both year1 and year2 refer to calendar years, e.g. 1972. The number of values specified must agree with the number of years specified in the time period clause. Values must consist of character strings with characters from the following set: < 0 1 2 3 4 5 6 7 8 9 . - > . Each value string may consist of from 1 to 10 of these characters.

No unit of measure or scaling factor is assumed in the MASH dictionary at the present time. Therefore, time series should be entered with a convenient scaling factor, e.g. U. S. GNP in billions, and macromodel equations that use or update the time series should assume the scale factor used to enter the series initially.

DELETE

The DELETE command is used to delete one or more entries from any dictionary entity type that consists of a list of names. The syntax of the DELETE command is:

DELETE name1 [, name2, ...] FROM { LIST / CENSUS / PROCEDURE / TABLE / MACROMODEL } entityname ;

The entity types from which list constituents may be deleted are specified above.

The names to be deleted may occur in any order and do not need to correspond to their ordering in the entity referenced. If any of the names to be deleted do not exist in the entity, the command is aborted without modifying the list-like entity. If the entity referenced is protected, the command will be aborted and the original entry will not be modified.

DEPOSIT

The DEPOSIT command is used to change the contents of any location in any of the virtual data arrays used by MASH. The syntax of the DEPOSIT command is:

DEPOSIT { ALPHABETIC / INTEGER / OCTAL / REAL } value IN arrayname (row, column) ;

The DEPOSIT command is analogous to the PDP-10 Monitor level deposit command, and like that command it should only be used by experienced users who have an adequate knowledge of the internal workings of the system.

Any of the MASH virtual array names may be referenced by the DEPOSIT command. These virtual array names are IU, FAM, PER, FAMAD, PERAD, FMINU, PRINF, CBOOK, ATABL, DICT, and INDEX. The row and column specification in the command may refer to any cell defined in the virtual array. If an illegal combination is entered, an error message will be typed and control will be returned to the Monitor, but software channels will not be released nor will files be closed. Typing CONTINUE will cause MASH to prompt for another MASH command without altering any virtual memory cell.

The first three virtual arrays, IU, FAM, and PER, are byte-structured arrays. Each column of these arrays is divided into a collection of variable length bytes that are packed into the words comprising the column. For these arrays, the DEPOSIT command interprets the row number to be the number of the byte pointer in the pointer array for that virtual array which is to be used to deposit the value specified.

A micropopulation must be open for browsing or simulation or both before the user can DEPOSIT into any of the nine virtual memory arrays which comprise the population. Unless a BROWSE or SIMULATION command has been performed, using the DEPOSIT command on any micropopulation virtual array is meaningless and will produce an error indication. Depositing into the DICT and INDEX arrays is a privileged function available only to the individual responsible for maintenance of the MASH system.

The composition of the value string depends upon the input mode selected. Octal numbers should be entered as halfwords separated by one or two commas (123,,456 or 777777,777777), and will be right adjusted in halfwords. Integers may be up to 10 digits long and may be either signed or unsigned. Alphabetic strings may consist of from 1 to 5 characters from the MASH symbol alphabet (figure A2-1);

they will be left adjusted and padded with blanks, if necessary. Real numbers may be signed and contain a decimal point; however, exponential notation is not accepted.

DISCARD

The DISCARD command is used to discard attributes from an initial micropopulation that were specified for inclusion in a previous INCLUDE command. The syntax of the DISCARD command is:

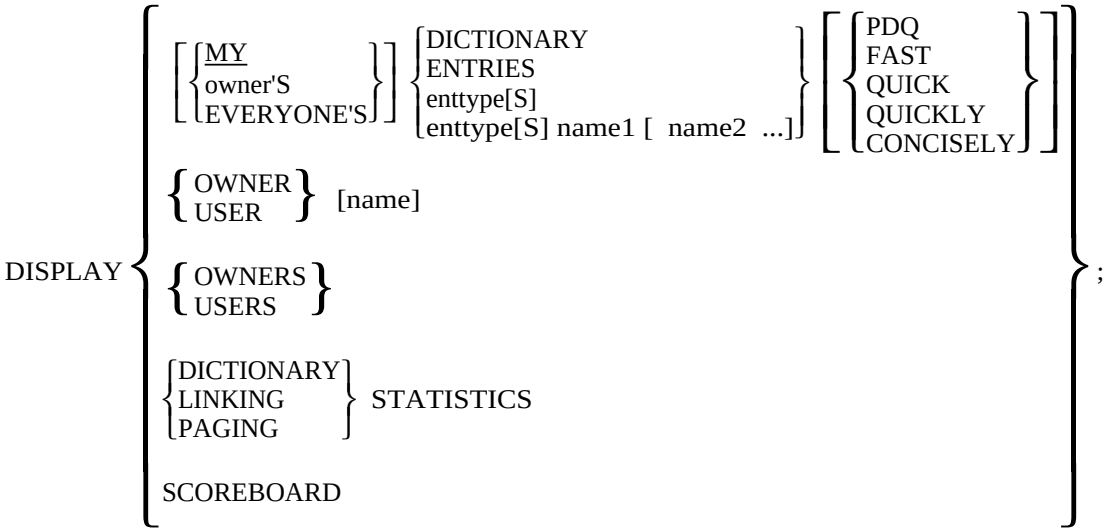
DISCARD ATTRIBUTE[S] { LIST listname
attributename1 [atributename2 ...] } ;

The DISCARD command should be given prior to the CREATE POPULATION command which triggers the creation of the population. Attributes flagged for discarding will be extracted from the survey population or the attribute library, but will not be included in the micro population.

The purpose of the DISCARD command is to provide a mechanism for using attributes in certain operations while creating a micropopulation without the necessity of retaining them in the population. A subroutine named MODISP is available if the programmer wishes to code specific instructions making use of such attributes. In addition, the REPLICATE command and the USING sample phrase of the CREATE command may include attributes and flag them for discard if they have not explicitly been included by the user. The mechanism for discarding these attributes is identical to the mechanism used by this command.

DISPLAY

The DISPLAY command is used to display individual dictionary entries, dictionary status and other information regarding the status of a MASH session. The syntax of the DISPLAY command is:



The DISPLAY DICTIONARY and DISPLAY ENTRIES commands type a list of all entities currently in the dictionary on the user's console. If no owner is specified, or if the word MY is included, only a list of the current user's entries is displayed. If a specific owner is named, the output will be restricted to the list of entities belonging to that owner. If the pseudo-owner EVERYONE is named in the command, data about all entries in the dictionary will be displayed.

If one of the "short-form" words (QUICKLY, etc.) is appended to the command, a list of entity names only, sorted by owner and type, is printed. In the absence of a "short-form" word, one line for each entity is typed, giving the name of the owner, the name of the entity, its type, its length, the date it was created or modified, the last date it was accessed, and its protection status.

If, instead of the word DICTIONARY or ENTRIES, a valid entity type (see the DEFINE command) is given, the output will be of the same form, but will be restricted to the entity type specified. If the entity type is followed by one or more names, the individual entities themselves will be displayed. Unless the command includes a "short-form" word, the data displayed will include the housekeeping information for each entity named.

The DISPLAY OWNER and DISPLAY USER commands are identical in meaning. If no owner name is specified, the name of the current owner is typed. If an owner name is specified, the list processing information associated with that owner is displayed, such as pointers to the first and last entities. Each of these commands types a list of the active owners and the number of entries each has in the dictionary. The DISPLAY DICTIONARY STATISTICS command displays the structural information stored at the beginning of the INDEX array.

The remaining forms of the DISPLAY command do not pertain to the dictionary. The DISPLAY LINKING STATISTICS command produces a list showing the number of times each of the non-resident links of MASH was loaded. The DISPLAY PAGING STATISTICS command displays statistics which are kept on the performance of the virtual memory simulator. The last form of the DISPLAY command, DISPLAY SCOREBOARD, gives the user access to the micropopulation counters, i.e., the size of the population in interview units, families and persons, that MASH maintains during a simulation exercise.

END

The END command is used to terminate a MASH run and return control to the PDP-10 Monitor system. The syntax of the END command is:

END { RUN / MASH } ;

The two forms of the END statement are identical.

The END statement closes the user's virtual files, closes the protocol file, and updates the MASH usage log by adding the time off for this MASH run and the amount of CPU time used. Thus, this is not equivalent to entering control-C, which should not be used to terminate a run.

ERASE

The ERASE command removes entries from the MASH dictionary. The syntax of the ERASE command is:

ERASE entitytype entityname [, [type] name ...] ;

A user may only erase his own entities, and then only if they are not protected from being erased.

More than one entity, as well as entities of different types, may be erased with a single ERASE command. The entity type need only precede the first of a list of entities of the same type, e.g.:

ERASE LISTS A, B, C, COMMAND X, SERIES GNP1, GNP2;

EXAMINE

The EXAMINE command displays a location in one of the MASH virtual memory arrays in a specific display mode. The syntax of the EXAMINE code is:

EXAMINE arrayname (row, column) [{ ALPHABETIC / INTEGER / OCTAL / REAL / ALLMODES }] ;

Any of the virtual memory arrays may be examined using this command. The command provides an analogue to the PDP-10 Monitor command EXAMINE for its main level store. The initial default display mode is octal. Once specified, however, any of the other display modes becomes the default; the display mode is "sticky". The ALLMODES option of the command displays the data in all modes.

MASH currently uses 11 virtual memory arrays whose names are IU, FAM, PER, FAMAD, PERAD, FMINU, PRINF, CBOOK, ATABL, DICT, and INDEX. The first 3 arrays are byte structured arrays and are addressed indirectly through an array of PDP-10 byte pointers in main store. For these 3 byte structured arrays, the row number in the EXAMINE command refers to the number of the byte pointer rather than the word in the array; for all other virtual arrays the row number points directly to the word to be displayed.

The first 9 virtual arrays are population specific. In order for the EXAMINE command to be executed correctly with these arrays, some micropopulation must have been opened with a BROWSE or SIMULATION command. The other 2 arrays, DICT and INDEX, are available to the user at any time during a MASH run.

EXECUTE

The EXECUTE command is used to invoke execution of one or a sequence of MASH commands. The syntax of the EXECUTE command is:

EXECUTE { commandname1 [commandname2 ...] / procedurename } ;

The EXECUTE command may reference either a sequence of names of commands or a procedure, which is a list of names of commands. If any of the commands named in the EXECUTE command or in

the procedure is not found, MASH will abort execution of the EXECUTE command. However, commands encountered prior to the non-existent command will have been executed. Likewise, if any of the commands cannot be successfully interpreted, the procedure will be aborted at that command and a message will be typed on the user's console indicating where the error occurred.

EXHIBIT

The EXHIBIT command is used to exhibit the structure of a micropopulation entity, its genealogy, or its history. The syntax of the EXHIBIT command is:

$$\text{EXHIBIT} \left\{ \begin{matrix} \text{GENEALOGY} \\ \text{STRUCTURE} \\ \text{HISTORY} \end{matrix} \right\} \text{ OF } \left[\left\{ \begin{matrix} \text{INTUNIT} \\ \text{FAMILY} \\ \text{PERSON} \end{matrix} \right\} \right] \text{ [n] ;}$$

If no entity name is specified, the current entity at the level named is assumed. If no entity level is named, the current person is assumed. The browsing pointers are reset so that they point to the interview unit, family and person referenced. If the command specifies the family or interview unit as the exhibit level, then the browsing pointers for the lower levels are set to the first entity contained in the entity one level higher.

The EXHIBIT command contains 3 distinct options: (1) exhibiting the genealogy of an entity; (2) exhibiting the structure of an entity; and (3) exhibiting the history of an entity.

The GENEALOGY option is applicable only to persons. It describes -- to the extent that there is information available in the micropopulation -- the ancestors of the person. This option has not yet been implemented.

The STRUCTURE option describes the structure of the interview unit that an entity is contained in, if it is not an interview unit itself. The names and addresses of all families and persons are listed that are currently contained in the same interview unit as the entity named.

The HISTORY option traces the history of either a family or person (since interview units do not move) through its various moves. Each time that a significant demographic change occurs for an individual or family, the entity moves to a new address and maintains a trail of both forward and backward pointers. The HISTORY option traces those pointers and specifies the time and reason for each entity move. Also, for interview units and families, current and former members are listed.

EXPORT

The EXPORT command is used to write MASH time series data into an external file in a form suitable for input to another program which can then perform specific types of analysis. The syntax of the EXPORT command is:

EXPORT [TO { PLANETS / TROLL / TSP }] [IN FILE filename] SERIES { LIST listname / name1 [name2 ...] } ;

The "filename" in the above syntax is the name of the PDP-10 file where the EXPORTed series are to be written. If the file name is not specified in the command, the series will be written in a file named nnnnX.DAT, where nnnn is the (protocol) number of the current MASH run. The series named in the command or in the list named in the command may be qualified with owners' names, but the list entity, if one is named, must be the property of the current user.

Because of its local availability and support, PLANETS is the default system to which series are EXPORTed. It is also the only system for which the command is currently implemented.

EXTRACT

The EXTRACT command is used to specify the names and locations of the PDP-10 files containing the survey data and its associated machine readable codebook that are used to create an initial micropopulation for simulation. The syntax of the EXTRACT command is:

EXTRACT FROM SURVEY FILE pdp10filename DESCRIBED BY CODEBOOK pdp10filename ;

Both the survey file and its codebook file are referenced by their PDP-10 file names, which have the implicit extension DAT. It is assumed that both the survey file and its codebook file reside on the public disk structure.

FIND

The FIND command is used to search the micropopulation to find a micro entity whose attribute values satisfy a specific condition. The syntax of the FIND command is:

FIND [{ FIRST / NEXT / ANOTHER }] { INTUNIT / FAMILY / PERSON } [STARTING { AT / WITH / AFTER } { INTUNIT / FAMILY / PERSON } n]

{ WITH attributename rel value / IN SAMPLE samplename } ;

The FIND command initiates a search through the current micropopulation until either it finds an entity that meets the specified criterion or the population has been searched through to its end. The command has three major items for specification: (1) the entity level to be searched; (2) the starting point for the search; and (3) the specification of the search criterion.

The search may be conducted for an interview unit, a family, or a person. The starting point is specified by the first optional word and by the optional STARTING phrase. The search may start at or after any micropopulation entity named in the STARTING clause. If this entity is not an interview unit, then its containing entities are determined, and the search proceeds by interview unit from that one to the interview unit with the highest name; that is, the search is not in ascending name order of person or family, but rather in ascending name order of interview unit, with all contained entities traced through in left list order. If a STARTING clause appears in the command, it takes precedence over the choice of FIRST, NEXT, or ANOTHER. However, if the clause is not included, then the first optional word of the command may be used to set the search to start with interview unit 1 or to start immediately after the current position of the browsing pointers. Thus, successive searches for entities having the same condition to meet may be performed by initially entering FIND FIRST... and subsequently entering FIND NEXT... commands. The words NEXT and ANOTHER are synonymous. If the word is omitted, the search begins with interview unit 1, i.e. at the beginning of the population.

The condition which is to be searched for may be specified in one of two ways. A relational operator from the set { .LT. < LE. <= .NE. /= .GE. >= .GT.> } may be used to bind a population attribute name to a numeric value to create a boolean condition which is either true or false for each micropopulation entity at the search level.

Alternatively, a sample definition containing only names of population attributes and literal values may be applied to each entity at the search level to yield a true or false condition. If the sample definition phrase is selected, then all attributes in the definition must be at least at the level of the entity type searched for in order to avoid ambiguity of evaluation. The IN SAMPLE phrase is not yet implemented in the FIND command.

Micro time series attributes may be referenced in the FIND command with subscripts, as explained in the description of the WHAT command.

GENERATE

The GENERATE command is used to generate one or more time series of attributes for each entity at one level of the micropopulation. The syntax of the GENERATE command is:

GENERATE [{ HIGHEST / LOWEST / FIRST / LAST }] k YEAR SERIES FOR
{ LIST listname / attributename1 [attributename2 ..] } ;

For each attribute referenced, 'k' additional attributes are created for each micropopulation entity at that level. The names of these new attributes are obtained by extending the original attribute name with zeros until the name is eight characters long and then appending a two digit sequential number to that root. For example, the micro time series attributes associated with the attribute named WEEKS would be WEEKS00001, WEEKS00002, ... etc.

Only one type and length of micro time series may be generated for any one survey or library attribute. Each of the attributes mentioned explicitly or addressed implicitly must have been previously included in an initial micropopulation being constructed with an INCLUDE command.

The GENERATE command must be issued prior to the CREATE which initiates the population creation process. The generated attributes are included in the micro population and have values generated for them during the course of the simulation process. Before simulating with a new population, the user can select an option which will cause the original values of attributes in the micro time series to be included as the initial values in the series.

IDENTIFY

The IDENTIFY command is used to alter the name of an attribute being included in an initial micro population. The syntax of the IDENTIFY command is:

IDENTIFY oldname1 AS newname1 [, oldname2 AS newname2, ...] ;

The IDENTIFY command changes the names of population attributes as they are being transferred to the new micro population. Attributes are selected by using the INCLUDE command and referencing their old names, but the codebook generated for the new micro population will refer to each attribute by its new name. Operating characteristics and on-line browsing commands incorporating the new name should refer to the attributes by their new names.

This feature allows some flexibility in renaming attributes when they are selected for inclusion in a population.

INCLUDE

The INCLUDE command is used to specify the inclusion of attributes from the survey file and from the attribute library in an initial micropopulation. The syntax of the INCLUDE command is:

INCLUDE { LIBRARY / SURVEY } ATTRIBUTE { LIST listname / attributename1 [attributename2 ...] } ;

The attributes to be included may be specified explicitly in the INCLUDE statement or they may be referenced indirectly by specifying the name of a list of attribute names in the command. Each invocation of the INCLUDE statement may refer either to attributes from the previously mentioned survey file or to attributes whose definitions exist in the attribute library, but the two sources cannot be intermixed within the same command. The effect of repeated INCLUDE commands is cumulative; subsequent executions of the command augment the list of attributes to be included.

Attribute names must be specified for the initial population with the INCLUDE command before they may be referenced in GENERATE, IDENTIFY, REPLICATE, and DISCARD commands.

LOOK

The LOOK command sets one of the browsing pointers to point to the entity named. The syntax of the LOOK command is:

LOOK AT { INTUNIT / FAMILY / PERSON } n ;

Using this command is the most explicit way of setting a browsing pointer. A population must be opened for browsing or simulation before the LOOK command is legal.

The LOOK command sets only one of the three browsing pointers; the others remain unchanged. Thus, it is possible to have the family browsing pointer referring to one family and the person browsing pointer referring to a person contained in a different family.

The name of the entity is checked for legality. If the name is illegal, a message is typed for the user and the browsing pointer remains unchanged.

OBTAIN

The OBTAIN command is used to specify which attributes of a micropopulation are to be extracted for a sample survey. The syntax of the OBTAIN command is:

OBTAIN [THE] ATTRIBUTES { IN LIST listname / attribute1 [attribute2 ...] } ;

Multiple OBTAIN commands are cumulative and may be issued in order to construct the complete list of attributes to been extracted and recorded in the survey data file. The attributes to be extracted may be initial population attributes, attributes whose definitions were originally extracted from the attribute library, or attributes whose definitions are stored in the user's dictionary.

ORIGINATE

The ORIGINATE command is used to initialize a new MASH dictionary virtual array and a new index virtual array. The format of the ORIGINATE command is:

ORIGINATE DICTIONARY codeword ;

where "codeword" is a password that must be entered correctly for the command to execute.

A dictionary is originated generally only once on any particular computer system. ORIGINATE creates and initializes the two virtual memory array files that form the nucleus of the MASH dictionary. The effect of invoking the ORIGINATE command is to supersede any existing dictionary files, so that the command should be used with great caution.

The "codeword" does not appear in this documentation, but must be obtained from the author of the system. Even then, the command is available only to the privileged user who is responsible for maintenance of the MASH system.

PREPARE

The PREPARE command is used to define macromodels prior to a simulation run. The general form of the command is as follows:

PREPARE MACROMODEL [owner'S] name [FOR PASS n] [, [owner'S ...]] ;

An owner's name may be specified in the command if the macromodel definition and the definitions of all its equations belong to an owner other than the current user. The pass number in the command indicates the microsimulation pass after which the macromodel should be executed. If omitted, the pass number defaults to 1.

As combined micro-macromodels grow in complexity, it may be desirable to interleave several macro submodels between the micro submodel passes. The PREPARE command can therefore be used to prepare a number of macro submodels, one for execution after each micromodel pass.

One, and only one model may be executed after each pass, and one and only one PREPARE command may be given. Each execution of a PREPARE command erases the actions of previous PREPARE commands, so all macromodels must be established with one command. The PREPARE command clears the association, tracking and iteration tables, so that ASSOCIATE, TRACK and other macromodel preparation commands should follow the execution of the PREPARE command.

PREVENT

The PREVENT command logically disables operating characteristics or other functions of a micromodel. The syntax of the command is:

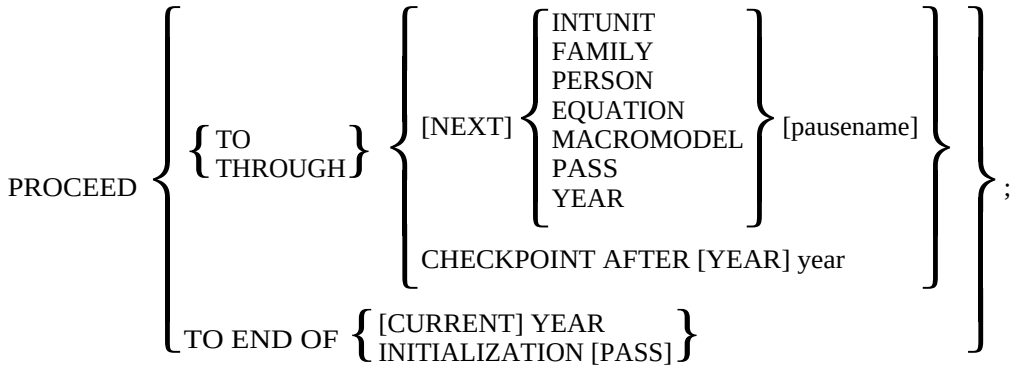
PREVENT opchar1 [, opchar2, ...] ;

The operating characteristics which may be PREVENTed in the current version of the Urban Institute Microanalytic Model are listed in the description of the ALLOW command.

The PREVENT command operates by setting (turning on) certain options. The options selected could also be set by the TURN command, but this command is self documenting. Initially all operating characteristics are enabled (i.e., all options are turned off).

PROCEED

The PROCEED command is used to initiate a micro-macrosimulation, and to continue from interrupts or pauses in the simulation process or in the population initialization pass. It determines the length of a simulation run (in simulated calendar years). The general form of the PROCEED command is:



When the "pausename" is specified, and the "pauselevel" is YEAR, the word YEAR may be omitted, e.g.:

PROCEED THROUGH 1984;

The "pauselevel" represents a specific processing step in the overall simulation or initialization process. PROCEEDing TO a processing step means stopping before it is executed, while PROCEEDing THROUGH it means pausing after it is executed. Thus, it is possible to pause before or after a specific PASS, YEAR, MACROMODEL, EQUATION or PERSON has been processed. A pause before a FAMILY or INTUNIT means stopping before pre-processing of the entity, and a pause after a FAMILY or INTUNIT means stopping after post-processing of the entity.

The PROCEED...CHECKPOINT command permits pausing before or after a previously scheduled checkpoint, and is included for convenience.

The PROCEED TO END OF YEAR command functions identically with the command PROCEED THROUGH YEAR n, where n is the calendar year currently being simulated.

The command PROCEED TO END OF INITIALIZATION PASS should be used to request completion of the process of creating an initial population since no year is being simulated, and the initialization pass is not part of a simulation exercise.

The PROCEED command determines the number of years that will be simulated by specifying the stopping point as a specific year. To complete the simulation, and to invoke the last (eleventh) entry point in a simulation agenda, the command TERMINATE SIMULATION should be used.

PRODUCE

The PRODUCE command initiates the process of taking a sample survey of a micropopulation. The syntax of the PRODUCE command is:

PRODUCE SURVEY [FILE] FILENAME [AND CODEBOOK [FILE] filename]
[IN project,programmer] ;

The PRODUCE command should be the first command specified in taking a sample survey. The first filename specified -- from 1 to 5 alphabetic characters -- will be the system file name for the survey data file. The second filename will be the system file name for the codebook file. The creation of the codebook file may be bypassed by omitting the phrase naming it; this is recommended only as part of a procedure in which several surveys are being taken, all of which are known *a priori* to have identical codebooks. Finally, the survey data and codebook files may be written in a disk area other than the current user's area by specifying an alternate user area by its project,programmer identifier.

PROTECT

The PROTECT command protects one or more dictionary entries from erasure or modification. The general form of the PROTECT command is:

PROTECT entitytype entityname [, [type] name, ...] ;

The entity type may be any legal dictionary type, and the entity name may be any existing entity of that type. Entities to be protected must be in the user's own dictionary. More than one entity, and entities of different types, may be protected with a single command. The entity type need only precede the first of a list of names of the same type. An example appears in the description of the UNPROTECT command.

Every dictionary entity is either protected or unprotected. With the exception of the special owner HISTORIC, new entities are not protected when created. Such entities can therefore be erased, modified or superseded in a number of ways, and the prior version of the entity will be lost. If the entity is protected, an attempt to erase or supersede the entity through any kind of modification will not succeed. The entity must first be unprotected and then modified or erased.

Protected entries are protected only against erasure or modification. They may be displayed and copied just like unprotected entities. Only one level of protection has been implemented in the dictionary system because of the cooperative nature of users of the system; hence, entity protection is really protection against accidental erasure only.

RENAME

The RENAME command is used to assign a new name to a dictionary entry in the current user's dictionary. The syntax of the RENAME command is:

RENAME entitytype entityname { AS / TO } newentityname ;

Any entity of any type may be renamed. The only restrictions are that the entity must exist in the current user's dictionary and that the new name does not duplicate the name of an existing entity of the same type. Entities may be renamed even if they are protected. Renaming an entity does not alter its protection status.

REPLACE

The REPLACE command is used to modify list-like dictionary entries by replacing one name in the list with another. The syntax of the REPLACE command is:

REPLACE oldname { WITH / BY } newname IN { LIST / CENSUS / PROCEDURE / TABLE / MACROMODEL } entityname ;

Replacement is performed on a one-for-one basis; addition and deletion of names may be performed using the ADD and DELETE commands.

Replacement of names in this manner is logically equivalent to superseding the earlier version of the entity. Entities that are protected therefore cannot be modified by a REPLACE command without first being unprotected.

If the list element newname is not found in the entity named, the command is aborted without modification of the entity.

REPLICATE

The REPLICATE command is used to replicate a specific person, with his family and interview unit, a prespecified number of times when creating a micropopulation for simulation. The syntax of the REPLICATE command is:

REPLICATE PERSON HAVING attributename = value n TIMES ;

The REPLICATE command must be entered prior to the CREATE command that begins the extraction process. Only one REPLICATE command may be given in this process; if more are given, the last command issued is the one used.

The first person encountered whose attribute value meets the condition specified in the REPLICATE command is flagged for replication. After all interview units, families, and persons have been extracted and entered into the micropopulation, that person and all his or her relatives in the interview unit are fetched, and *n* copies of the interview unit, families, and persons are created. Each created entity has a new and unique name and address.

At the present time, there is no explicit way to identify which micropopulation entities are the result of replication after the population has been formed. The command is of use in preparing a population containing one interview unit, replicated *n* times, and observing distributions of outcomes generated by the simulation process.

REPORT

The REPORT command gives the user access to status information about the current job and about the simulation process. The general form of the REPORT command is:

REPORT STATUS [OF statustype [, statustype, ...]] ;

where "statustype" is one of RUN, JOB, OPTIONS, POPULATION, SIMULATION, CHECKPOINTS, or EVERYTHING.

If no "statustype" is specified in the command, EVERYTHING is the default. The "statustypes" may appear in any order, although the output will always be in the order listed above.

RUN and JOB are synonymous, and refer to the "environmental" information, including MASH run and version numbers, project-user, user console and job numbers, date and time elapsed and CPU time used, and number of disk reads and writes executed since the beginning of the job. Also included in the RUN or JOB status is the name of the current user (current dictionary owner), the run mode (batch or interactive), and the title of the current MASH run, if any.

The OPTIONS status consists of the list of options which have been turned on, either explicitly with the TURN command or implicitly by other commands or procedures.

The POPULATION status includes the population number, the base year, the number of years that have been simulated, the original and current weighting factors, and the maximum addresses and the current browsing pointers at each entity level.

The SIMULATION status includes the base and current calendar years, and the number of years that have been simulated. MASH reports whether microsimulation has been requested (whether a SIMULATION command has been used) and whether micro or macrosimulation, or neither, is in progress. Also, MASH lists the names of the "current" micro entities and the simulation passes for which macromodels have been PREPAREd, and the place at which simulation has paused, if appropriate.

The CHECKPOINT status indicates when checkpoints are scheduled to occur, and whether it is currently the end of a simulation year.

The REPORT STATUS command may be executed at any time. Some of the machine usage information may be useful for estimating the cost or duration of planned runs, etc. The status of EVERYTHING is automatically displayed whenever a fatal error occurs, before a checkpoint is taken, and when a session has been restarted.

SAVE

The SAVE command is used to save the current status of a micropopulation.[5] The syntax of the SAVE command is:

SAVE POPULATION;

Once a browsing or microsimulation function has been initiated, the 9 virtual array files associated with the population are kept open, with part of the population data in main memory, including the data contained in the sequential status file for the population, designated the PSF. The SAVE command closes all of the virtual array files associated with the population, and rewrites an up-to-date copy of the PSF in the user's disk area. This allows the user to continue processing this population at any future time.

SEARCH

The SEARCH command is used to initialize the values of scalars in equations constituting a macromodel. The syntax of the SEARCH command is:

SEARCH PARAMETER LIST [owner'S] listname ;

A parameter list is an ordinary MASH list of elements, with the important exception that the elements are paired; the first element of a pair is a scalar name, and the second element is the value for that scalar. The following command defines a MASH list that can be interpreted as a parameter list:

DEFINE LIST PARAMS AS PI, 3.14159, ALPHA, 0.92, BETA, 0.075, ADJ, 1.35;

Since a parameter list is syntactically no different from any other MASH list, care should be taken when using ADD, DELETE or REPLACE commands to modify a parameter list that a sequence of paired elements remains.

---

[5] The SAVE command is intended for use only when a population has been used for browsing, and attribute values of selected individuals, families or interview units have been altered with the CHANGE command. In particular, the SAVE command is not necessary before checkpoints are taken or before a TERMINATE command is used, since both operations close the current population. If the user desires to suspend a simulation in mid-year, the CHECKPOINT NOW command should be used. It closes the population without updating the population status file, which is the operation desired in this case. The SAVE command should only be used when it is desired to close a population *and* update the PSF file.

The SEARCH command provides a mechanism for assigning values to scalars in a macromodel that are undefined at the time of the SEARCH command. For this reason, the macromodel(s) to be used must be designated with a PREPARE command before a SEARCH command is used.

The SEARCH command retrieves the list named, and attempts to match every name in the list with the name of all undefined scalars in the macromodel(s) prepared. If a match occurs, the numeric value following the scalar name is assigned to the scalar as its initial value. As many searches of different parameter lists as desired may be performed in sequence. However, only the *first* name match is used to initialize a scalar; subsequent matches are ignored.

Parameter lists belonging to other owners may be searched by specifying an owner name in the SEARCH command. An option is provided which will cause the names of the scalars defined and the values assigned during a search procedure to be typed on the user's console.

SET BASE, INITIAL

The SET BASE (or SET INITIAL) command is used to specify the base year of a simulation or the survey year of a micropopulation. The general form of the SET BASE or SET INITIAL command is:

SET { BASE / INITIAL } YEAR TO year ;

The YEAR clause is used to set the base or initial year of a macro-only simulation, as well as to specify the year that a micropopulation represents. The base year must be specified with the SET command before any of the other commands which describe an initial population should be used.

SET TYPING

The SET TYPING command is used to specify the default time period for the TYPE command, or the number of macro time series that should be typed across the page. The general form of this command is:

SET TYPING { RANGE TO year1 THROUGH year2 / WIDTH TO n } ;

Initially, if no time period is specified in the TYPE command, the series named will be typed for the entire period for which they are defined. The SET TYPING RANGE command can be used to alter that default to any range of years. The value of year2 must be greater than or equal to the value of year1.

If the typing width is not altered with a SET command, five series will be printed across the page when a TYPE command names more than one series. In some circumstances it may be useful to alter the page width. For instance, if simulation results are to be compared with historical time series, it is convenient to have the series typed in pairs, four across the page. The SET TYPING WIDTH command can be used to set the page width to 1, 2, 3, 4 or 5 columns.

SHOW

The SHOW command is used to display the current state of all or part of the macromodel(s) that have been prepared. The syntax of the SHOW command is:

SHOW MACROMODEL [arrayname [, arrayname, ...]] ;

where "arrayname" can be any of: TABLE, EQUATIONS, STACK, SCALARS, CONSTANTS , TRACKING, ASSOCIATIONS, BINDINGS , or ALL.

Each of the "arraynames" represents an internal control table that contains information pertinent to macromodels. These are the tables which are constructed with PREPARE, ASSOCIATE, TRACK, and SEARCH commands. The tables are reset by a PREPARE command or by a CLEAR command.

The contents of these tables will not be of interest to the general user. Rather, they are useful either for debugging purposes or for obtaining a more detailed understanding of the operation of the macromodel definition and solution module.

SIMULATION

The SIMULATION command is used to specify the micropopulation which is to be used in a microsimulation. The general form of the command is:

SIMULATION POPULATION $\left[\left\{ \begin{matrix} \text{NUMBER} \\ \text{NAME} \end{matrix} \right\} \right]$ IS n [, TREAT AS [IF] NEW] ;

Unless the SIMULATION command is used to specify a micropopulation name, only macrosimulation will take place -- assuming one or more macromodels have been PREPAREd -- when a PROCEED command is given. When the SIMULATION command is executed, MASH reads the population status file associated with the population, opens the virtual memory population files, and performs other initialization appropriate for that population.

The TREAT AS IF NEW clause of the command can be used to "pretend" that the population in question resulted from a survey taken during the year it represents.[6] For populations which have not been moved forward in time, the clause has no effect. However, if the population has been moved forward in time using simulation, the following operations take place: (1) the base year of the population is set to the "current" year; (2) the number of years already simulated is set to zero; and (3) the original population weight is made equal to the current population weight. When simulation is begun, the first of the eleven entry points in the agenda is used. In other words, each operating characteristic is directed to perform "pre-simulation processing", which consists of zeroing tables that accumulate statistics for an entire run. The micropopulation is *not* re-initialized; all entities retain their "histories" and attribute values.

START

The START command is used to create a dictionary for a new owner. The syntax of the command is:

START DICTIONARY FOR ownername ;

The "ownername" given in the command may be from 1 to 10 alphanumeric characters in length. MASH confirms the creation of a new sub-dictionary with the statement:

DICTIONARY STARTED FOR ownername;

The new owner also becomes the current owner, as if the command USE owner'S DICTIONARY; had been executed.

---

6 The TREAT AS IF NEW feature may be useful for evaluating the accuracy of a model in use. For instance, a 1960 Census population may be moved forward in time until it represents a 1970 survey. The decade of the 1970's can then be simulated using this generated population, as if it came from the 1970 Census. The results can then be compared to those obtained by simulating with a population created directly from the 1970 Census.

SUMMARIZE

The SUMMARIZE command is used to request that summary information be retained by one or more summary operating characteristics, and that it be displayed at the end of each simulated year or at the completion of the entire simulation run. The general form of the SUMMARIZE command is:

SUMMARIZE opchar1 [, opchar2, ...] { YEARLY / FINAL } ;

The operating characteristics which may be referenced in the SUMMARIZE command are among those listed in the description of the ALLOW command. The user is informed if he attempts to request a summary for an operating characteristic which does not have a paired summary characteristic. Summaries may be obtained about any combination of operating characteristics for which corresponding summary characteristics exist, both yearly and at the end of the simulation run.

TAKE

The TAKE command is used to schedule output operations during the course of a simulation. The syntax of the TAKE command is:

TAKE CENSUS name [{ IN year1 [year2 ...] / NOW / { EVERY { YEAR / k YEARS } / YEARLY } [STARTING { IN year / NOW }] }] ;

A *census* is an ordered list of command names. Each command named is an output command that operates upon the current micro population. Thus, a census is a list of outputs to be obtained from a micro population.

The TAKE command is used to schedule the named census either immediately or one or more times in the future. The absence of a time schedule clause implies the default alternative now. Alternatively, the census may be scheduled in an arbitrary sequence of years year1, year2, ... or at regular intervals, either starting immediately or in some future year.

The TAKE command has not yet been implemented. When it is implemented, the CANCEL command will be extended to allow cancellation of censuses scheduled for future execution.

TERMINATE

The TERMINATE command is used to end a simulation run. The general form of the TERMINATE command is:

TERMINATE SIMULATION [AND CHECKPOINT [(option)] [INTO proj,prog]] ;

The TERMINATE command invokes the last of the microsimulation agenda entry points. The "post-simulation processing" includes closing the current micropopulation, and printing the final summary information, if requested. No values of any micropopulation attributes are changed, but the population status file is updated to reflect its current status.

The TERMINATE command may only be issued at a pause at the end of a simulation year. The PROCEED command is used to determine the length of a simulation run, and the TERMINATE command is used to complete the simulation. The following pair of commands is typical:

```
PROCEED THROUGH YEAR 1969;
TERMINATE SIMULATION;
```

The AND CHECKPOINT clause of the TERMINATE command invokes the same set of procedures that a scheduled end-of-year checkpoint does, except that the micropopulation files are not copied,

TITLE

The TITLE command may be used to assign a title to a MASH run that will appear at the top of each page of the protocol file. The syntax of the TITLE command is simple:

TITLE [title-of-up-to-65-characters] ;

The MASH run title can be composed of any text which is meaningful to the user. Since the title appears at the top of each protocol page, it can facilitate comparisons between similar MASH runs. The title may consist of up to 65 characters from the set of graphics (printing characters). The title may be changed at any time, and becomes effective at the top of the next protocol page.

TRACK

The TRACK command is used to specify alternate sources for values of variables in a macroeconomic model. The syntax of the TRACK command is:

```
TRACK variable1 [ { FROM year1 THROUGH year2 }
                  { IN year1               }
                  { FOR ALL YEARS          }
                  { BEFORE START           }
                  { WHEN MISSING           } ]

  { USING [ { HISTORIC } ] series1 }
  {       [ { owner'S  } ]         }  [, variable2 ... ] ;
  { AT CONSOLE                     }
```

Tracking a macromodel variable over a specific time period means that whenever a value for that variable is required from within that time period to evaluate an equation, the value is either taken interactively from the console or from a series other than the one with which it is otherwise associated. The values calculated for an endogenous variable within a macromodel are stored in the time series normally associated with the variable. Tracking affects only the input of values to the macromodel, i.e. variable references on the right hand sides of equations.

The time period specified in the TRACK command may be a single year, a range of years, or all years of a simulation run. Also, the user may specify that a variable be tracked using a specified source when values are required for the variable for a year prior to the first year of simulation. Lastly, a variable can be tracked when a value cannot be found in the associated time series.

The source of the values for the variable being tracked may be any time series in the dictionary. If a variable is tracked "AT CONSOLE", the user will be requested to enter values at the time they are required.

More than one tracking specification may be made with a single command. Multiple TRACK commands may be given during the course of a simulation run, but only the most recently entered tracking specification for a given variable for a given year will apply.

TURN

The TURN command is used for two functions; to set the status of MASH logical options interrogated and acted upon by sections of MASH code, and to suppress and restore printing on the user's console. The syntax of the TURN command is:

```
TURN { ON  }  { CONSOLE                    }  ;
     { OFF }  { OPTION[S] n1 [ n2 ... ]    }
```

There are currently a maximum of 1000 MASH logical options, approximately 200 of which are used by code in the current microanalytic object modules and in MASH. Any one or more of these options may be turned on (elected) or off (suppressed) in one TURN command.

The CONSOLE option in the TURN command is a command level way to suppress terminal output.[7] When the console is disabled by this command, output normally directed to the user's terminal is suppressed; however, all such output is written on the protocol file for the run, so that it will contain a complete record of the run. Thus, if the user wants to bypass a particularly long section of terminal output for the present, this may be done by turning the console off logically and turning it on later.

TYPE

The TYPE command is used to display time series data or to display the current "news" file from within MASH. The syntax of the TYPE command is:

```
TYPE { { FROM year1 THROUGH year2 } { TABLE tablename                          } }
     { { IN year                  } { SERIES [owner'S] series1 [ [owner'S ... ] } }
     {                                                                          }  ;
     { [MASH] NEWS                                                              }
```

Although the DISPLAY command can be used to display time series data, just as it can be used to display the contents of any dictionary entity, the format of the display has limitations. The TYPE command permits the user to specify the range of years for which the time series should be typed, and also gives the user control over the number of time series that are printed across the page.

---

[7] The TURN ... CONSOLE statement is logically independent of the control-O function of the PDP-10 Monitor for suppressing terminal output; either or both may be used. Terminal output that is cancelled temporarily with a control-O transmission also is retained on the protocol file for the run, so that no session output is lost using either method.

If the user does not specify a time period in the TYPE command, all years for which any value exists are displayed. The SET TYPING RANGE command can alter this default. MASH normally displays time series in up to five columns across each page.

The TYPE command allows the user to specify a list of series names, modified by owner name if desired, or the name of a TABLE, which is a MASH list of series names. Although the TABLE itself must belong to the current user, it may contain references to series other than the current user's series.

The other form of the TYPE command, TYPE NEWS, gives the user access to the news file that is maintained in the MASH library. This file contains descriptions of the latest changes to the system and other information of interest to MASH users. The date and time that the file was last updated usually will appear in the message that greets the user when MASH is first entered. Since the news file may be lengthy, the user may wish only to see the first part of the file which contains the most recent information. The typing of the news file can be interrupted with a carriage return while it is being typed. MASH will then stop printing the file and will prompt the user for another command.

UNPROTECT

The UNPROTECT command is used to remove erase protection from any dictionary entity. The syntax of the UNPROTECT command is:

UNPROTECT entitytype entityname [, [type] name, ...] ;

Every dictionary is either protected against erasure and modification or is not so protected. Protected entries may be displayed and copied, but may not be superseded or erased. New entries, except those of the owner HISTORIC, are not protected. They may be protected with the PROTECT command and protection may be removed with the UNPROTECT command.

More than one entity, and entities of different types, may be unprotected with the same command. Only the first name in a list of entities of the same type need be preceded by the entity type. For instance:

UNPROTECT SERIES U1, U2, COMMANDS INIT2, INIT3, INIT4;

USE

The USE command selects the user's sub-dictionary from the MASH dictionary and uses that part of the dictionary for subsequent commands referring to it. The syntax of the USE command is:

USE { owner'S DICTIONARY / DICTIONARY BELONGING TO owner } ;

Each entry in the dictionary has a type and a name and is qualified further by owner name. Invoking the USE command implicitly qualifies all references to the MASH dictionary with the owner name in the USE command. Entries belonging to other owners can be referenced in a variety of commands using an explicit possessive form of qualification in the command. In the absence of a USE command in a MASH dialogue, the current owner is assumed to be STAFF.

The two different forms of the USE statement are semantically equivalent; one is included for historical continuity.

WATCH

The WATCH command is used to request MASH to type individual entity results as one or more operating characteristics are applied to each micro entity. The syntax of the WATCH command is:

WATCH [{ ALL / SUCCESSFUL / THWARTED }] opchar1 [, opchar2, ...] ;

Many of the operating characteristics in the current Urban Institute microanalytic model include the ability to report on the micro events they are simulating. This micro output generally includes the entity's name, some demographic data, an indication of whether an event is simulated to occur or not, the probability of the event occurring for the individual entity, the random number upon which the decision was made, and other such intermediate data. These outputs generally are not neatly formatted, and their interpretation depends upon a specific knowledge of the program producing them. The output can be useful for debugging the object model form of an operating characteristic and for examining the behavior of the characteristic in detail.

For those operating characteristics which result in an event occurring or not occurring (e.g. birth and death), "success" and "failure" conditions are defined for the event. A "successful" application of an operating characteristic results in the event occurring, while a "thwarted" or unsuccessful application of an

operating characteristic results in the event not occurring. The WATCH command can restrict the output of micro results to those events that actually occur, or to those that do not.

The operating characteristics that can be WATCHed in the current version of the Urban Institute Microanalytic Model are among those listed in the description of the ALLOW command. If the user attempts to WATCH an operating characteristic for which the command has no meaning, he is so informed.

The CANCEL command contains forms that permit the user to negate the effect of the WATCH command. The three options of the WATCH command are treated as separate and independent items. Thus, if the user has been WATCHing THWARTED births, and wishes to stop doing so, he must cancel the watch of THWARTED births, not of ALL births.

WHAT

The WHAT command is used to determine the current value of an attribute of an entity in the micropopulation currently open for browsing. The syntax of the WHAT command is:

$$\text{WHAT} \quad \left\{\begin{matrix}\text{IS}\\ \text{ARE}\end{matrix}\right\} \quad \text{attributename} \left[\text{OF} \quad \left\{\begin{matrix}\text{INTUNIT}\\ \text{FAMILY}\\ \text{PERSON}\end{matrix}\right\} \quad \mathbf{n}\right] \quad [?] \;;$$

The attribute name specified in the command must be one of the attribute names in the machine readable codebook describing the micropopulation.[8] The words IS and ARE are synonymous. If an entity level is specified, the attribute must be defined at that entity level. If an entity level is not specified, the current entity at the level at which the attribute is defined is interrogated.

Invoking the WHAT command sets the browsing pointer at the entity level selected to the entity name specified. Thus, in retrieving several attribute values for the same entity, the entity need only be named in the first WHAT command.

WHERE

The WHERE command is used to determine the current address of an entity whose name is specified. The syntax of the WHERE command is:

---

[8] Names of micro time series attributes may be specified in the WHAT command as subscripted attributes. For instance, the attribute WEEKS(1) would refer to the attribute whose name is actually WEEKS00001.

WHERE IS { INTUNIT / FAMILY / PERSON } n [?] ;

If the entity name is legal, then the current address of the entity will be typed in return. If the entity no longer exists -- such as a person who has died -- then MASH will return the last address that the entity inhabited and a message that the entity no longer exists.

WHICH

The WHICH command is used to determine the name of the entity that lives at the address specified in the command. The syntax of the WHICH command is:

WHICH { INTUNIT / FAMILY / PERSON } LIVES AT [ADDRESS] n [?] ;

The WHICH command is the inverse of the WHERE command.

WHICH operates by scanning the relevant address array; for a large population such an operation may be time consuming. MASH does not keep any inverse lists of entity name by entity address. If an entity used to live at the address but has moved, MASH will report that no entity is currently living at the address. If an entity lived at that address and then ceased to exist, MASH will report the name of the entity that last lived there and expired.

WHO

The WHO command fetches the contents of one of the browsing pointers and displays the name of the current entity at that level. The syntax of the WHO command is:

WHO IS [THE] CURRENT { INTUNIT / FAMILY / PERSON } [?] ;

If the browsing pointer at the level specified has not been set, either implicitly or explicitly, MASH informs the user that there is no current entity at the level specified.

The WHO command reports the name of the interview unit, family, or person pointed to by a browsing pointer. Since these pointers are independent of the simulation process, the REPORT STATUS

OF SIMULATION command should be used to determine the names of the entities currently being processed during a simulation.