019238

UM-HSRI-82-24

DRIVER RECORD ANALYSIS SYSTEM ENHANCEMENT

Christopher R. Ford
John A. Green

FINAL REPORT
JULY 1982

THE UNIVERSITY OF MICHIGAN
HIGHWAY SAFETY RESEARCH INSTITUTE

Report Number UM-HSRI-82-24

DRIVER RECORD ANALYSIS SYSTEM ENHANCEMENT

Christopher R. Ford
John A. Green

The University of Michigan
Institute of Science and Technology
Highway Safety Research Institute
Ann Arbor, Michigan

July 1982

Prepared under Contract No. MTR-81-003A

Office of Highway Safety Planning
Department of State Police
Lansing, Michigan

Technical Report Documentation Page

HSRI

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
UM-HSRI-82-24		47319

4. Title and Subtitle	5. Report Date
Driver Record Analysis System Enhancement	July 1982
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Christopher R. Ford, John A. Green	UM-HSRI-82-24
9. Performing Organization Name and Address	10. Work Unit No.
Highway Safety Research Institute 2901 Baxter Road Ann Arbor, MI 48109	11. Contract or Grant No. MTR-81-003A
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Office of Highway Safety Planning Department of State Police 111 S. Capitol Avenue, Lower Level Lansing, MI 48933	Final May 1981 to May 1982
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The Michigan Driver Incident Record Analysis (MDIRA) computer program is an interactive analysis package that allows special counts to be made of Michigan driver incidents (convictions, accidents, and improvement actions). These counts can be quite complex. For example, the question "how many drivers have had two speeding tickets within three years, or three drunk driving convictions within six years, or both" could be answered by the program. Such information has value for policy planning within the Department of State.

As originally implemented, the first version of the MDIRA program developed by HSRI proved to be a valuable tool in analyzing driver incident data. Time did not permit, however, for the program to be fully refined and optimized. With the completion of the current project, the MDIRA program has been greatly enhanced. Operational costs have been dramatically reduced, and the program's user-interface has been smoothed and polished. As a result, the potential usefulness of the MDIRA program may now be more fully realized.

17. Key Words	18. Distribution Statement
Driver analysis computer program; Driver improvement; Policy planning; Highway Safety.	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
		120	

TABLE OF CONTENTS

1.0 INTRODUCTION

The Highway Safety Research Institute (HSRI) has assisted the Michigan Department of State since 1974 in the establishment of sample files based on driver registration records. These sample files have been built in a form which, unlike the state's Master Driver Record file, can be readily analyzed using available statistical analysis packages resident on the University of Michigan computing system. These files have contributed to improved planning and problem identification in the driver licensing and improvement area. Comparison of driver records in time periods that differ by a year or more have yielded results which throw considerable light on the distribution of accidents and violations among Michigan's drivers.

1.1 Background

The Highway Safety Research Institute conducted an earlier study for the Office of Highway Safety Planning entitled "Estimation of the Coverage of New Programs Based on Driver Incident Data" (HS Project MDL-78-001A).[1] The purpose of that study was to develop a system for reformatting and analyzing driver registration records maintained by the Michigan Department of State in order to determine the effect of proposed legislative standards and regulations. The program developed to analyze the reformatted driver incident data became known as the "Michigan Driver Incident Record Analysis" (MDIRA) system. The system was completed on schedule in June, 1979, and has been used successfully in subsequent Department of State operations.

---

[1] John A. Green, Paul E. Lehner, and Christopher R. Ford, Estimation of the Coverage of New Programs Based On Driver Incident Data (Ann Arbor: The University of Michigan Highway Safety Research Institute, 1980, UM-HSRI-80-7).

1.2 Objectives

Since the MDIRA program became available, the need for operational refinements not dealt with in the original study became apparent. These included modifications to the program to make it more "user friendly" and therefore useable by a wider variety of people, production of better documentation to promote program proficiency in system utilization as well as in initial training, and changes to the program code to decrease the operational costs of the program. In addition, changes in the structure of the Master Driver Record (MDR) file maintained by the Department of State necessitated corresponding changes to the reformatting procedures used to produce the Driver Incident Record file (used as input to the MDIRA program).

The general design and implementation of the system was completed in the previous study. The intent of the current study was to update, enhance, and document the system to provide a more efficient program oriented to a wider group of users in state government.

In summary, the project had three objectives: (1) to enhance the cost-effectiveness of the MDIRA program through reduced operating costs, (2) to make the program easier to use through an improved user-interface, and (3) to incorporate recent changes made to the Michigan Master Driver Record file into the MDIRA system.

1.3 Overview

The work performed to accomplish the first objective of the project, improved program performance, proved to be quite significant. The results of this effort are discussed in Section 2.

Enhancements to the user-interface of the program were greatly facilitated by the rational, coherent data structures developed to improve program performance. The success at producing a more user-friendly program is covered in Section 3.

Finally, recent changes to the Master Driver Record file have also been incorporated into the MDIRA system. Work in this area is discussed in Section 4.

2.0 OPERATIONAL EFFICIENCY

One of the primary goals of the project was to increase the operational efficiency of the MDIRA program. This goal was met in three ways: by a reduction in the amount of time spent by the program in the computer's central processing unit (CPU), by a reduction in the amount of memory occupied by the program, and by an improvement in the accuracy of the program's count results.

2.1 Reduction of CPU Time

Early tests of the MDIRA program indicated that the program was spending most of its time (and thus incurring most of its cost) performing two actions: reading and writing the data (input/output) and determining condition set satisfactions (checking/counting).

2.1.1 Input/Output Improvement

The program's method of input/output was addressed first. In the original version of the MDIRA program, standard input/output procedures were in use that processed the data one record at a time. A more efficient method was implemented in the new MDIRA version using subroutines that now process the data in blocks, with each block containing perhaps several hundred records. Only one read or write, rather than several hundred, is thus required to input or output an equivalent number of records.

2.1.2 Checking/Counting Improvement

A detailed look at the subroutines performing the actual incident checking and counting in the old version of MDIRA revealed several operational inefficiencies. In addition, the data structures that defined the condition sets were determined to be somewhat limited (e.g., a variable with more than 500 non-consecutive values could not be stored in a condition set) and generally in need of improvement.

Rather than attempt to patch the existing checking/counting subroutines, the decision was made to write entirely new routines. An algorithm was developed with one primary goal in mind: to determine as quickly as possible if a given incident caused a driver to satisfy a condition set specification. Parallel to the development of this algorithm, a new condition set data structure was conceived. Together, this new algorithm and data structure proved to be the heart of a completely rewritten MDIRA program.

2.1.3 Evaluation of Improvements

To evaluate the success of these efforts, several benchmark tests were made, running the old and new versions of MDIRA side by side and comparing the CPU time needed to complete the runs. All the runs were made on a test file consisting of 20,000 incident records. Each pair of runs varied only by the make-up of the count specification. Table 1 displays the results of these runs.

It became obvious early in this testing phase that a dramatic improvement in program efficiency had taken place. The last column in Table 1 indicates the degree of this improvement.

The count specifications of the first six runs each consisted of a single condition set. The definition of this single condition set was varied to determine if there were any significant differences between the two versions dependent on the way values were specified. A single variable with a single value (e.g., "V7=41"), a range ("V7=41-61"), and multiple values ("V7=41,51,61,71"), as well as a multiple variable specification and a condition set with the NUMBER and RANGE feature used, were all tried. Though there were differences, they did not seem to fluctuate greatly from run to run. Generally speaking, when

TABLE 1

Benchmark Test-Run Comparisons
of Old and New MDIRA Versions

Count Specification	CPU Time (seconds)		Percent Improvement
	Old	New	
1 variable, 1 value	3.383	.8	322.9
1 variable, 1 range	3.245	.849	282.2
1 variable, 2 ranges . . .	3.343	.885	277.7
1 variable, 12 values . . .	3.364	.885	280.1
2 variables, each 1 range .	3.319	.847	291.9
1 variable, 2 ranges RANGE, NUMBER set	3.607	.872	313.6
2 condition sets, each with 2 variables . .	5.166	1.055	389.7
3 condition sets, 3rd combination of 1st two	6.138	1.166	426.4
4 condition sets	7.113	1.273	458.8

a single condition set is being counted, the new MDIRA version shows nearly a 300% improvement over the old version in the CPU time taken to perform the count.

Differences, however, did begin to be observed when the count specification consisted of more than one condition set. As can be seen from the last three runs in Table 1, the times for the old version got progressively worse with each additional condition set. Though test runs were made using count specifications that included only up to four separate condition sets, the degradation is striking and gives every indication of continuing as the specification size continues to increase. Here the improvement is 400% or more, and only getting better.

In an attempt to see more clearly what was happening here, a separate set of test runs were made. These runs sought to determine the amount of time spent by each of the two versions performing the two major program actions:

reading in the data and checking and counting it. Two sets of test runs were made, one using a count specification with a single condition set, the other using a specification with three condition sets. The results of these tests are displayed in Table 2.

TABLE 2

Percentage of Time Spent Reading Versus Checking For Old and New MDIRA Versions

Count Specification	READ		CHECK/COUNT	
	Old	New	Old	New
1 condition set	53.7	85.6	46.3	14.4
3 condition sets	36.6	77.2	63.4	22.8

The input/output actions of any program are usually quite time-consuming. In an optimized program, in fact, one would expect to find a majority of the program's time performing these actions. Clearly, as Table 2 indicates, this describes the new version of MDIRA. The table also graphically displays the inefficiency of the old version's checking/counting subroutines, particularly as the complexity of the count specification is increased.

2.2 Reduction of Virtual Memory

Incidental to the improvement of CPU timings, the new version of MDIRA also experienced a reduction in the amount of space taken up by a loaded program in the computer's memory. The old version of MDIRA occupied 97 pages of virtual memory, while the new version occupies only 36 pages. Most of the reduction can probably be attributed to the improved condition set data structure used by the new version.

2.3 Improvement of Program Accuracy

Another important aspect of this testing phase was verification that the new program was producing correct results. During the early development of the new version, initial test runs, though not extensive and made on a relatively small file, did, in fact, produce results identical to those produced by the old version. As extensive tests of the programs proceeded, however, discrepancies between the two count results began to appear. Usually the differences were minor (e.g., same number of drivers, incident count off by three), but on occasion they were major (drivers off by 28, incidents off by 177). And though the the old version usually came up with numbers less than those produced by the new version, one set of runs had the undercount going to the new version.

Needless to say, these developments were somewhat disturbing. To determine which set of results were correct, a separate, 32 statement counting program was written. This "mini-MDIRA" program used an algorithm and data structure uniquely its own. In addition, because of its small size and simplicity, its own results were more easily verifiable. The results of this third program were identical to those of the new version, thus confirming that the counts of the old version were incorrect. It would seem from the evidence, then, that not only has a greater degree of efficiency been achieved with the new program, but a greater degree of accuracy as well.

3.0 OPERATIONAL EASE-OF-USE

Another major objective of the project was to enhance the interface between the program and the user, that is, to improve the operational "ease-of-use" of the MDIRA program. Included in this effort was improvement of the general command processing (or program control) of the program, enhancement of the separate commands, addition of several new commands, implementation of a HELP/EXPLAIN facility, and improvement of the external documentation of the system.

3.1 Improved Program Control and Syntax

The manner in which the program processes command input has been improved. The old version of MDIRA had used an interesting method to process commands. Once a command had been issued, the user was placed into a separate mode of operation specific to that command. Exit from this mode was (usually) only possible by issuing a special "END" command. Though a useful method for the DEFINE command (a similar technique is still used), it added an unnecessary level of complexity to most of the commands and was a frequent source of complaint of MDIRA users. Except for the DEFINE command, all program commands are now handled as single-entry operations.

The way in which condition sets are specified has also been improved. The inconsistent manner in which condition sets were specified in the old program was a source of frustration. For example, the DEFINE command required that condition set one be specified as "C1", while the DESCRIBE command required that it be input as "C=C1", and the COUNT command required "C=1". Attempting to specify condition set one as "C=1" for the DEFINE command or "C=C1" for the COUNT command only produced error messages. A single subroutine now decodes all condition set specifications, allowing for a uniform condition set syntax throughout the program. This subroutine is also quite tolerant, and would accept "C1", "C=1", or "C=C1" as identifiers for the same condition set.

As a result of these and other improvements, the "look" of the program has changed considerably. Two sequences of commands are displayed in Table 3, one from the old MDIRA version, the other from the new version. Both sets of commands perform virtually the same operations.

TABLE 3

Command Sequence Comparison
of Old and New MDIRA Versions

Old Version	New Version
SETIO IN=6POINTSUB END DEFINE C1 V7=43 V10=1-5 C2 V7=43 V10=6 C3 COM=(C1),(C2) END DESCRIBE C=C1-C3 END COUNT C=1-3 STOP	DATA INPUT=6POINT.SUB DEFINE C=1-3 V7=43 V10=1-5 V7=43 V10=6 C1 OR C2 DISPLAY C=1-3 COUNT C=1-3 STOP

The program's attention interrupt processing has also been upgraded. The old MDIRA version performed a degree of interrupt processing, but not in the best way possible. For example, the program could never be temporarily exited by issuing a break (a common feature of other programs). In addition, issuing a break while the program was processing a data file produced only a message reporting the cost for reading 100 drivers (usually only a few cents) and asking if processing should continue. Not much information was provided to help the user decide whether the program should, in fact, continue. Strangely enough, when the old version was operated in batch, this question had to be answered, despite the fact that no interrupt had been issued. In contrast, the new MDIRA version will take the user out of the program whenever two interrupts are issued in succession. If an attention interrupt is given while the program is processing a data file, the user is informed how many drivers and incidents have been read up to the interrupt, how much money has been spent, and is asked whether processing should continue or not, or whether control should temporarily return to MTS. No unnecessary questions need be answered in batch.

3.2 Command Improvement

Almost all of the program commands have been improved in one way or another. In some instances this improvement has been in providing more information to the user, while in other instances the improvements have been to require less of the user in order to operate the program smoothly.

For all commands, error messages have been added that describe more fully and more clearly incorrect user input. All of the commands that perform some internal change within the program (e.g., DEFINE or CLEAR) now provide "feedback" messages that confirm those changes that have successfully taken place (e.g., "Defined: C1").

The DESCRIBE command has been renamed "DISPLAY" and provided with additional functions. The command's improved ability to display condition set definitions is shown in Figure 1. As can be seen in the examples, the display of condition set definitions is more readable and now includes the names of the variables, as well as the specific values the variables have been defined with.

Figure 1 - Condition Set Display Comparison

DESCRIBE Command Output from Old MDIRA Version

```
CONDITION SET   1          TYPE = SIMPLE
NUMBER =   1           RELATIVE RANGE = NONE
VAR =   3      MINIMUM =      31      MAXIMUM =      46
VAR =   7      MINIMUM =      43      MAXIMUM =      43
VAR =  10      MINIMUM =       1      MAXIMUM =       5
```

DISPLAY Command Output from New MDIRA Version

```
---------------
Condition Set 1   -   Simple Type

  V3 LICENSE CODE              = 31,36,41,46
  V7 COUNTY OF RESIDENCE       = 43
 V10 POINTS FROM INCIDENT      = 1-5
```

The MOUNT command has been improved so that tapes may now be mounted with a default pseudo-device name or, optionally, with any name selected by the user. Tapes may now also be mounted from within the program with their write-enable rings in. MDIRA tapes that are mounted prior to running the program are always recognized by the program, no matter what pseudo-device name they happen to be using.

The SETIO command has been renamed "DATA" and improved so that any input or output disk or tape file may be accessed, regardless of whether or not the file name is included in the list of "officially" recognized MDIRA files. Many of the processes that were formerly required by the user when writing to magnetic tape (e.g., positioning to the correct file, setting blocking parameters, etc.) are now handled internally by the new input/output subroutines used by the program.

3.3 New Commands

A TRANS command has been added that performs single Gregorian to Julian and Julian to Gregorian date translations for the user. Three of the MDIRA variables are dates, two of which are stored in a Julian format, while the third has a YYMMDD format. The old program accepted values for all three of these variables in a MMDDYY format only. But when displaying the values with the DESCRIBE command, the program printed them in their internal format. With the thought that use of these variables would be confusing when a MDIRA data file was accessed with another data analysis system that wouldn't automatically perform these translations, the new version of MDIRA now requires that values be input exactly as they appear in the data file. The TRANS command has been provided to assist with this requirement when either of the two Julian-format variables are used.

A RECOUNT command has been added that will reprint the results of the last COUNT command. This command is useful if a COUNT table has disappeared off the terminal screen and needs to be viewed again.

A RELEASE command has been implemented that will automatically release MDIRA tapes.

3.4 HELP/EXPLAIN Facility

Two commands have been added that provide online documentation for the program: HELP and EXPLAIN. These commands work in a similar fashion. For instance, issuing the command "HELP" while in MDIRA will produce a complete overview of all the program commands, while issuing the command "EXPLAIN COUNT" will produce a more detailed description of the COUNT command, along with an example.

3.5 Improved Documentation

External documentation for the MDIRA program has been rewritten. Each command is now fully described with examples. A copy of the documentation is presented in Appendix A of this report.

4.0 INCIDENT FILE UPDATE

Supplemental to the enhancement of the MDIRA program, recent changes made to Michigan Master Driver Record (MDR) file were incorporated into the MDIRA system. This work included a review of the changes, derivation of new code values to reflect these changes, and modification of the build program to incorporate these changes into the MDIRA data files.

4.1 MDR File Review and Coding Changes

During the project period, two separate reviews of the Michigan MDR file took place. Each time, up-to-date coding information was obtained from the Department of State Data Processing Center and compared with information used to build the previous incident file.

Many coding changes were found and procedures were devised to incorporate them in the new file build. The most significant change occurred with the value assignment of conviction incidents. A new coding scheme was implemented that now preserves the original DOS coding of the offense code, accommodates point differences for any individual offense in a logical fashion, and allows for changes in the future to be made much more easily.

4.2 Build Program Modification

The program used to reformat the driver records received from the state into a file structure useable by MDIRA was then modified to accommodate these changes. In the process, the program was improved by being broken down into several modules. For example, the program now has one subroutine that exclusively handles driver header records, another that handles driver convictions, and so on. In this form, the program has already proved to be much more manageable. Supplemental to these improvements, several program bugs were uncovered and eliminated, and sections of the program code were cleaned and commented.

4.3 Incident File Builds

Two incident file builds took place during the project period. A codebook that displays the results of the second build appears in this report as Appendix B.

The document describing the incident file build process itself has been updated to reflect the changes made to the program. A copy of this document is presented in Appendix C. For reference purposes, the MDR tape layout used by the Michigan Department of State Data Processing Center has also been included in this report. It appears as Appendix D.

5.0 CONCLUSIONS

As originally implemented, the first version of the MDIRA program developed by HSRI proved to be a valuable tool in analyzing driver incident data. Time did not permit, however, for the program to be fully refined and optimized. With the completion of the current project, the MDIRA program has been greatly enhanced. Operational costs have been dramatically reduced, and the program's user-interface has been smoothed and polished. As a result, the potential usefulness of the MDIRA program may now be more fully realized.

As in any endeavor, however, room is left for improvement. Several new commands that would perform additional functions have already been proposed. A CONVERT command, for instance, that would automatically convert data used by the MDIRA program into a form able to be input into other analysis programs would be useful. Another command could be added to perform summary counts of user-specified elements over certain time periods, an action that is currently being performed by a special program external to MDIRA.

Refinements to the current commands have also been suggested. For example, allowing the user to associate a label or tag with each defined condition set would be another useful enhancement.

Whether or not these improvements take place, the MDIRA program as currently implemented should continue to prove to be a valuable asset to those looking at the special nature of driver incident data.

APPENDIX A

MDIRA Program Documentation

MDIRA

Michigan Driver Incident Record Analysis

MDIRA is an interactive analysis program that allows special counts to be made of Michigan driver incidents. The program is invoked with the command:

$RUN SK4N:MDIRA

Several commands recognized by the program allow the user to, among other things, define condition sets, set input and output files, perform condition set counts, and subset files for further analysis.

MDIRA Data Files

The Michigan Department of State maintains a data file containing information on every driver registered in the state called the Master Driver Record file. Special subsets of the MDR file are made periodically by the Department of State Data Processing Center and provided to HSRI. These subsets may be, for example, 1% random samples of all the drivers, 10% random samples of probationary drivers, or all drivers with licenses issued in 1975. The raw data provided by the state is then condensed into specially formatted MDIRA data files for analysis.

The files used by the MDIRA program have a fixed-length format. Each record in the file represents one driver incident and is made up of several predefined variables. Data are stored internally for these variables as four-byte binary integers. Currently, every MDIRA data file consists of at least twelve "standard" variables. They are:

V1 - DRIVER NUMBER
V2 - INCIDENT NUMBER
V3 - LICENSE CODE
V4 - ORIG LICENSE ISSUE DATE
V5 - BIRTH DATE
V6 - AGE ON INCIDENT DATE
V7 - COUNTY OF RESIDENCE
V8 - SEX
V9 - INCIDENT CODE
V10 - POINTS FROM INCIDENT
V11 - INCIDENT DATE
V12 - VEHICLE TYPE

A listing of some of the data files available for analysis may be obtained by using the MDIRA DISPLAY command. A codebook documenting the data file variables and variable values is available from HSRI.

Program Control

The MDIRA program is controlled by entering a command statement in response to the prompt:

MDIRA Command
?

The command statement itself consists of a valid MDIRA command that may begin in column one of the input line or anywhere beyond. The command may be followed by one or several command modifiers or keywords separated from one another by blanks. The modifiers and keywords for a particular command are described in the write-up of that command. All commands, modifiers, and keywords have a minimum acceptable abbreviation that may be used. These minimum acceptable abbreviations are indicated in this documentation by an underlined portion of the command, modifier, or keyword. If the command statement is too long for the input device, the line may be terminated with a dash ("-") and continued on the next line. A single command statement, however, cannot be longer than 511 characters.

In addition to program commands, any valid MTS command may be issued within MDIRA for processing provided that it is preceded by a dollar sign ("$"). Null input lines, as well as those beginning with an asterisk (considered a "comment" line by MTS) may also be freely entered.

Attention interrupts (i.e., Break, Attention, Attn, etc.) may be issued anytime during the execution of MDIRA to stop the action currently underway. Issuing two attention interrupts in succession at the command prompt will return the user to MTS, in which case the program may be reentered by issuing the MTS command "$RESTART". If a data file is being processed at the time the interrupt is issued, the prompt

XX drivers, XX incidents read. $X.XX used. Continue?

is printed, where "XX" is the total number of drivers and incidents read up to the time the interrupt was encountered, and "X.XX" is the total dollar amount spent since the user signed on. The user may respond by entering "YES" to continue processing the data, "NO" to abort processing, or "MTS" to return temporarily to MTS command mode. If the user chooses to return to MTS, the command "$RESTART" will cause MDIRA to resume processing the data file.

Condition Sets

Central to the operation of MDIRA is the concept of the "condition set". Condition sets are user-determined specifications, precise requirements that incidents, or groups of incidents, are said to either pass (that is, "satisfy") or fail. Condition sets come into existence with the DEFINE command, and are subsequently referenced by the CLEAR, COUNT, DISPLAY, and SUBSET commands. Up to 25 condition sets may be defined at any one time during a single program run. Detailed information on the make-up of condition sets is provided in the description of the DEFINE command.

Condition sets may be specified in either one of two ways: with the condition set "modifier" or with the condition set "keyword". The condition set modifier has the form "Cn", where "n" is a number between 1 and 25. The condition set keyword has the form "CSETS=conditionset-list", where "conditionset-list" is a list of up to 25 condition set numbers, separated by commas or dashes, with no embedded blanks. The minimum acceptable abbreviation for "CSETS" is the letter "C".

The following table displays several examples of condition set specifications. Each pair of specifications is an identical way to specify the same condition sets.

Equivalent Condition Set Specifications

Modifier	Keyword
C1	CSETS=1
C1 C2	C=1,2
C1 C3 C4 C5	CS=1,3-5
C5 C6 C7 C8 C9	C=5-9

Throughout this document, the character string "cset" is used with many of the command prototypes to indicate that some condition set specification is required. Either the keyword or the modifier format may be used.

CLEAR

Purpose: To clear condition sets.

Prototype: (a) CLEAR cset
(b) CLEAR ALLCSETS
(c) CLEAR

Action: The CLEAR command clears (or erases) condition set definitions.

With prototype (a), only the condition set(s) specified are cleared.

With prototypes (b) and (c), all condition sets are cleared, and the program memory location where definitions are stored is reinitialized.

Examples of the CLEAR command

```
    MDIRA Command
→→ ?CLEAR C1

    Cleared: C1

    MDIRA Command
→→ ?CLEAR

    Cleared: All condition sets

    MDIRA Command
   ?
```

COUNT

Purpose: To count condition sets.

Prototype: (a) COUNT cset
(b) COUNT ALLCSETS

Action: The COUNT command performs condition set counts using the currently-assigned data input file.

With prototype (a), only the specified condition sets are counted.

With prototype (b), counts are performed of all the currently-defined condition sets.

The start of the actual count is signaled with the message "Processing Begins", while the end is signaled with "Processing Completed". Depending on the size of the data file being analyzed and the complexity of the condition sets being counted, the time from start to finish may last from several seconds to several minutes.

Immediately upon the completion of the count, a table is printed out displaying the count results. The sample table below, along with the description that follows, can be used to illustrate the method for interpreting these results.

	C1	C2	C3
Times Met	--	--	--
1 -	5	34	128
2 -	0	2	47
3 -	0	0	12
Drivers-	5	36	187
Tot Met-	5	38	258

For condition set one, five drivers met its requirements exactly once. Thirty-four drivers satisfied the second condition set one time, while two satisfied it twice, for a total of 36 drivers and 38 satisfactions. Condition set three was satisfied by 128 drivers exactly once, 47 drivers exactly twice, and 12 drivers three times, for a total of 187 drivers and 258 condition set satisfactions.

Example of the COUNT command

```
 MDIRA Command
→→ ?COUNT C1 C4

 Processing Begins
 Processing Completed

 Drivers read   =    166
 Incidents read =   2347

 ----------------------
                 C1    C4
 Times Met       --    --
      1  -        5    57
      2  -        0    14
      3  -        0     6
      4  -        0     3

 Drivers-         5    80

 Tot Met-         5   115


 MDIRA Command
?
```

DATA

Purpose: To assign input and output data files.

Prototype: (a) DATA INPUT=filename
(b) DATA INPUT=filename,TAPEn
(c) DATA OUTPUT=filename
(d) DATA OUTPUT=filename,TAPEn,file#

Action: The DATA command allows input and output data file assignments to be made. If the file to be assigned is located on magnetic tape, the tape must be mounted prior to the use of the command. All MDIRA analysis operations require an input file. An output file is needed only for operations that generate a new MDIRA data file (e.g., SUBSET).

With prototype (a), the specified file is made the program input file. A number of files are available for analysis. Issuing the command statement "DISPLAY FILES" will cause their names and locations to be displayed. Files specified by this first format that do not appear in this listing are assumed to be on disk.

With prototype (b), an input tape file not included in the "DISPLAY FILES" listing (see above) may be assigned. The specification "TAPEn" (where "n" is the number of a recognized MDIRA tape) indicates which tape the file resides on. It must immediately follow the file name, separated by a comma.

With prototype (c), the specified file is made program output file. The file is assumed to be on disk and, if not a temporary file, must exist at the time the command is issued. In addition, the file name must not duplicate any of the names already included in the "DISPLAY FILES" listing.

With prototype (d), the specified tape file is made the program output file. Again, the file name must not duplicate any of the names already included in the "DISPLAY FILES" listing. The specification "TAPEn" (where "n" is the number of a recognized MDIRA tape) indicates which tape the file will be written on. It must immediately follow the file name, separated by a comma. It, in turn, may be followed by the specification "file#" which determines the tape file number of the output file. A value of "0" (or zero) will cause the file to be placed at the end of the tape. The "file#" specification may be omitted altogether, in which case the value "0" will be assumed.

Note that the program will not allow input and output assignments to be made to files on the same tape at the same time. Note also that whenever an assignment is made to a disk file as, for example, input, and the same disk file is the currently-assigned output file, the output assignment will be cancelled by the program.

Examples of the DATA command

```
     MDIRA Command
→→  ?DATA INPUT=LID75

     Assigned: LID75

     MDIRA Command
→→  ?DATA OUTPUT=-TEMP

     Assigned: -TEMP

     MDIRA Command
    ?
```

DEFINE

Purpose: To define condition sets.

Prototype: (a) DEFINE cset
(b) DEFINE ALLCSETS
(c) DEFINE

Action: The DEFINE command allows MDIRA condition sets to be created and held in memory for the duration of the program run. One or more condition sets may be defined with one use of the command.

With prototype (a), prompts are made for each of the condition sets specified. An end-of-file can be issued at any time to terminate the prompting.

With prototype (b), continuous prompts are made for condition sets one through twenty-five. Again, an end-of-file issued at any prompt will cause the prompting to cease.

With prototype (c), a single prompt is made for the next available condition set (i.e., the lowest-numbered condition set that is currently undefined).

At each separate condition set prompt, either a simple or compound definition may be entered (see below). If any one definition line is too long, the line may be terminated with a dash ("-") and the definition continued on the next line. The total definition may not exceed 511 characters, however.

Successfully entered condition set definitions will replace any previously-defined condition sets of the same number, and will remain in effect until either explicitly cleared with the CLEAR command or implicitly cleared by another definition entry.

Simple condition set definitions consist of a group of variable value specifications, together with a possible NUMBER/RANGE value setting. There are currently twelve "standard" variables available in each MDIRA data file. Depending on the data file, other variables, ranging from variable 13 through 25, may also be available. Variable values are specified with the variable keyword. The keyword has the form "Vn=valuelist" where "n" is the variable number (1 through 25) and "valuelist" is one or more values the variable may take on. The value list may consist of one value (e.g., "V4=34900"), a range of values (e.g., "V3=30-40"), or any combination of the two ("V10=19,119,200-299"). Imbedded blanks are not allowed in the value list.

With the use of one or several variable keywords, the user defines the requirements of condition set satisfaction. For example, if a condition set were entered as "V6=1600-1999 V8=1", the condition set would only be satisfied by an incident where the driver was a teenage male. On this level, condition set satisfactions represent individual incidents that meet the requirements of the variable keywords.

A higher level of complexity (and power) may be attained, however, with the use of the NUMBER and RANGE keywords. Used together, these keywords specify the number of times that the variable values must be satisfied within a certain range of months for the condition set to be satisfied once. The values specified by the NUMBER and RANGE keywords must be single integer values greater than zero. The minimum acceptable abbreviation for NUMBER is "N", while the minimum for RANGE is "R". If, for example, a condition set were entered as "V6=1600-1999 V8=1 NUMBER=2 RANGE=6", the condition set would be satisfied once whenever two incidents were found within six months of each other where the driver was a teenage male. Because the NUMBER and RANGE keywords operate on the date of the incident, incidents with missing dates are not considered.

The figure below can be used to illustrate the operation of the NUMBER and RANGE keywords more clearly.

```
| X XX        |          X |            |X X X     X|
```

Consider each section of the figure to represent one year of a driver's driving history, with each "X" representing one incident meeting the requirements of the variable keyword settings (drunk driving convictions, for example). If the value of NUMBER were three and RANGE was set to twelve (i.e., one year), this driver would satisfy the condition set three times, once in the first year and twice in the fourth year.

Compound condition sets are made up of other condition sets, grouped together in "and/or" combinations. A single condition set "group" consisting of condition sets one and two, for example, would be entered as either "C1 C2" or "CSETS=1,2". This definition would require that both condition sets one <u>and</u> two be satisfied for the compound condition set to be satisfied once. Up to 24 previously-defined condition sets may be specified in a single group. Other compound condition sets may be included in the definition as long as the condition set number is less than the number of the condition set being defined.

Condition set groups may then be linked together with the special OR operator. For example, the specification "C1 C2 OR C1 C3" would define two groups with the satisfaction of either group satisfying the compound condition set once. Please note that parenthesis are not allowed in the specification. Notice, too, that there is <u>no</u> corresponding AND operator (an implied AND ties together all the condition sets of a single group). Up to 25 groups of condition sets, all linked together with OR operators, may be included in a single definition.

Examples of the DEFINE command

```
    MDIRA Command
→→ ?DEFINE C=1-2

    Enter Definition for Condition Set 1
→→ ?V3=31,36,41,46 V9=607

    Enter Definition for Condition Set 2
→→ ?V10=6 NUM=2 RANGE=12

    Defined: C1 C2

    MDIRA Command
→→ ?DEFINE

    Enter Definition for Condition Set 3
→→ ?C1 OR C2

    Defined: C3

    MDIRA Command
→→ ?DISPLAY C=1-3

    ---------------
    Condition Set 1   -  Simple Type

      V3 LICENSE CODE              = 31,36,41,46
      V9 INCIDENT CODE             = 607

    ---------------
    Condition Set 2   -  Simple Type

         NUMBER = 2          RANGE = 12

     V10 POINTS FROM INCIDENT      = 6

    ---------------
    Condition Set 3   -  Compound Type

      GROUP 1  = C1
      GROUP 2  = C2

    MDIRA Command
   ?
```

DISPLAY

Purpose: To display the status or setting of various items.

Prototype:
(a) DISPLAY cset
(b) DISPLAY ALLCSETS
(c) DISPLAY FILES
(d) DISPLAY INPUT
(e) DISPLAY OUTPUT

Action: The DISPLAY command displays the current setting or status of a number of MDIRA-related items. More than one item may be specified at one time (e.g., "DIS I O").

With prototype (a), the current definitions of the specified condition sets are displayed.

With prototype (b), the definitions of all the currently-defined condition sets (if any) are displayed.

With prototype (c), the names and locations of a number of input data files are displayed. These data files are available for analysis.

With prototype (d), the name of the currently-assigned input data file is displayed.

With prototype (e), the name of the currently-assigned output data file is displayed.

Examples of the DISPLAY command

```
 MDIRA Command
→→ ?DISPLAY C1 C2

 ---------------
 Condition Set 1   -  Simple Type

   V9 INCIDENT CODE            = 318,319,418

 ---------------
 Condition Set 2 is undefined

 MDIRA Command
→→ ?DIS FILES

  The following are currently defined files :

  --  File Name  --  Location  --  Description

   1%INC.0681        TAPE1       1979 1% Sample
   LID75             TAPE1       Lic Iss Date 75
   LID75/CON2        TAPE3       LID 75 / Conv

 MDIRA Command
→→ ?DIS INPUT

 Input: SK4P:6POINT.SUB

 MDIRA Command
?
```

EXPLAIN

Purpose: To provide on-line program documentation.

Prototype: (a) EXPLAIN
(b) EXPLAIN command

Action: The EXPLAIN command provides instant on-line documentation of the MDIRA program. (The HELP command is a synonym for EXPLAIN.)

With prototype (a), a brief listing of all the available MDIRA commands is printed out at the terminal.

With prototype (b), a brief description of the specified command is printed out (where "command" is a valid MDIRA command).

Example of the EXPLAIN command

```
 MDIRA Command
→ ?EXPLAIN EXPLAIN

 The EXPLAIN command provides instant on-line
 documentation of the MDIRA program.  EXPLAIN
 used alone produces a brief overview of all
 the available commands.  EXPLAIN followed by
 a command name produces a brief description
 of that command.

 For example:  EXPLAIN EXPLAIN

 MDIRA Command
?
```

MOUNT

Purpose: To mount MDIRA tapes.

Prototype: (a) MOUNT TAPEn
(b) MOUNT TAPEn=pseudo-device name
(c) MOUNT TAPEn WRITE=external-label

Action: The MOUNT command automatically mounts MDIRA tapes. Up to 25 predefined tapes have been provided for, though only a limited number may be available at any one time. All tapes given as the location of input data files should be available (use "DISPLAY FILES" to determine these). More than one tape may be mounted with one use of the command (this is the recommended procedure, in fact, whenever more than one tape is to be mounted so that the operator receives all the mount requests at one time). The command has no effect on tapes that are already mounted.

With prototype (a), the tape specified is mounted. "TAPEn" refers to tape number "n", where "n" is an integer value from 1 to 25. When mounted, tapes are assigned the pseudo-device name "*TAPEn*".

With prototype (b), the tape specified is mounted with the pseudo-device name given. If, for example, more than one MDIRA tape was to be accessed during one program run, the first tape could be mounted with the command statement "MOUNT TAPE1=*T*" and, after analysis operations had been performed using the tape, the command statement "MOUNT TAPE2=*T*" would automatically dismount the first tape and put the second tape on the same drive. The specified pseudo-device name is totally up to user. A specification such as "MOUNT TAPE2=*TAPE1*" is, in fact, perfectly valid.

With prototype (c), the tape specified is mounted with its write-enable ring in. This action will allow new files to be written to the tape. "External-label" must match the tape's external label in order for the write specification to be successful. Though more than one tape may be mounted with one use of the MOUNT command, only those with external labels matching the WRITE keyword's value will be mounted with their rings in.

It should be noted that the MOUNT command need not be issued if the tape to be accessed is already mounted. This is true even if the tape was not mounted within the MDIRA program, or was mounted with a pseudo-device name other than "*TAPEn*".

Example of the MOUNT command

```
    MDIRA Command
→→ ?MOUNT TAPE1
    Tape mount in process
   #*TAPE1* (C5759C): Mounted on T905

    MDIRA Command
   ?
```

MTS

Purpose: To temporarily return to MTS.

Prototype: MTS

Action: The MTS command returns control to MTS without unloading the MDIRA program. If no commands that invoke the loader are issued while in MTS (e.g.,"$RUN"), the command "$RESTART" may be used to reenter the MDIRA program at the point where the MTS command was issued.

Example of the MTS command

```
     MDIRA Command
-->  ?MTS
     Use $RESTART to reenter MDIRA
-->  #F
     ADFILE        DOCUMENT.T   DSET          FINAL
     MDIRCDBKCRDS  MDIRDICT.1B  MDIRDICT.1C   MDIRX
     OCTPROGREP.T  WORDS        XEC.EX
-->  #RES

     MDIRA Command
    ?
```

POINTSUM

Purpose: To perform point summation.

Prototype: (a) POINTSUM
(b) POINTSUM VARIABLE=variable
(c) POINTSUM AGE=age-range
(d) POINTSUM MONTHS=months

Action: The POINTSUM command performs point summation for each driver, producing a new data file with a new variable containing the results of this summation.

An example of the way the summation takes place may be useful. If a driver's first incident had a point count of two (indicated by variable 10), then the new variable's value would be two for that incident record. If the second incident was a six point conviction, the new variable's value would be eight. And so forth. For non-point incidents (i.e., those with a value of 8 or 9 for variable 10) or those not meeting the AGE or MONTH specifications (see below), the new variable's value will be "99".

Prototypes (b) through (d) indicate that a number of optional keywords may be used to control the summation. Any or all may be omitted, in which case default values will be used.

The VARIABLE keyword is used to specify the variable number where the summation results will be put. It must be between 13 and 25. It cannot be more than one greater than the highest input variable number (i.e., if the input data file ranged from V1 to V12, the specification V=14 would be invalid). If omitted, the new variable number will be one greater than the highest input variable number.

The AGE keyword is used to specify an age range. If used, the summation is limited to incidents where the driver's age (variable 6) falls somewhere within the range specified. For example, the specification "AGE=1600-1699" would cause points to be summed for sixteen year-old drivers only. Note that age values are stored internally in a YYMM format. If omitted, all incidents are considered.

The MONTHS keyword is used to specify a single value that represents the number of months over which points will be summed. If MONTHS were twelve, for example, points would be summed for an entire year. At the end of the year, the summation would begin again (i.e., be reinitialized to zero). Incidents with missing incident dates (variable 11) are not included in the summation. If omitted, the summation takes place over the driver's entire driving career.

An output file, as well as an input file, must, of course, be already assigned before the summation can take place. If the output file is to be written on magnetic tape, the tape must be mounted with the specification "WRITE=YES" (achieved with the MOUNT command's WRITE keyword). The tape must, at the same time, be a predefined MDIRA tape. Note, too, that the input file and output file cannot be on the same tape.

Example of the POINTSUM command

```
     MDIRA Command
-->  ?POINTSUM

     New variable will be V13

     Processing Begins
     Processing Completed

     Drivers read   =  166   Drivers written   =  166
     Incidents read= 2347   Incidents written= 2347

     MDIRA Command
     ?
```

RECOUNT

Purpose: To reproduce the last COUNT command table.

Prototype: RECOUNT

Action: The RECOUNT command reproduces the table printed by the mostly recently-issued COUNT command. Because no actual file processing is done by this command, an input data file need not be currently assigned for the command to be used. The only requirement for the command's use is that there be a previous COUNT result available.

Example of the RECOUNT command

```
    MDIRA Command
--> ?RECOUNT

    ----------------------
                  C1     C4
    Times Met     --     --
        1  -       5     57
        2  -       0     14
        3  -       0      6
        4  -       0      3

    Drivers-       5     80

    Tot Met-       5    115


    MDIRA Command
   ?
```

RELEASE

Purpose: To release MDIRA tapes.

Prototype: (a) RELEASE TAPEn
(b) RELEASE

Action: The RELEASE command releases MDIRA tapes.

With prototype (a), the tape specified is released. "TAPEn" refers to tape number "n", where "n" is an integer value between 1 and 25.

With prototype (b), all currently-mounted MDIRA tapes (if any) are released.

Example of the RELEASE command

```
     MDIRA Command
→→  ?RELEASE
    #T905 released.

     MDIRA Command
    ?
```

STOP

Purpose: To terminate program operation.

Prototype: STOP

Action: The STOP command terminates the MDIRA program. If any MDIRA tapes remain mounted at the time the command is issued, a reminder is printed out at the terminal.

Example of the STOP command

```
     MDIRA Command
→→  ?STOP
    #Execution terminated
    #
```

SUBSET

Purpose: To subset driver incidents.

Prototype: (a) SUBSET cset
(b) SUBSET ALLCSETS

Action: The SUBSET command produces a subset of the current incident file based on the specified condition set(s). The incidents of all drivers who satisfy any of the condition sets at least once are copied into a new data file. This new data file may then be used as an input file for use with other MDIRA commands.

With prototype (a), the specified condition sets are used to determine the subset. Any that are satisfied at least once cause the driver to be included in the subset.

With prototype (b), all currently-defined condition sets are used to determine the subset.

It should be emphasized that the filtering action of the condition sets operates on a driver level, rather than on an incident level. For example, suppose a subset was made specifying a condition set that selected only HBD accidents. If a certain driver had twenty incidents with only one being an HBD accident, all twenty incidents would nevertheless be included in the output data file.

To repeat also, drivers need only satisfy one of the condition sets, not all of the condition sets, to be included in the subset.

An output file, as well as an input file, must, of course, be already assigned before the subset can take place. If the output file is to be written on magnetic tape, the tape must be mounted with the specification "WRITE=YES" (achieved with the MOUNT command's WRITE keyword). The tape must, at the same time, be a predefined MDIRA tape. Note, too, that the input file and output file cannot be on the same tape.

To provide some idea of the size of disk files produced by the SUBSET command, 300 incident records that include variables one through twelve occupy approximately five disk pages.

Example of the SUBSET command

```
     MDIRA Command
-->  ?SUBSET C1

     Processing Begins
     Processing Completed

     Drivers read   =  4439  Drivers written   =  166
     Incidents read= 20000  Incidents written= 2347

     MDIRA Command
     ?
```

TRANS

Purpose: To translate Julian and Gregorian dates.

Prototype: (a) TRANS Gregorian-date
(b) TRANS Julian-date

Action: The TRANS command performs date translations. Both the Original License Issue Date (variable 4) and the Incident Date (variable 11) are stored internally in a Julian format (sequential from March 1, 1900). The TRANS command simplifies their use when either need to be specified within a condition set.

With prototype (a), the Julian equivalent of the given Gregorian date is printed at the terminal. The date may be specified in either a MM/DD/YY or MM-DD-YY format.

With prototype (b), the Gregorian equivalent of the given Julian date is printed.

Examples of the TRANS command

```
    MDIRA Command
→→ ?TRANS 12-31-79

    Julian translation is: 29160

    MDIRA Command
→→ ?TRANS 29161

    Gregorian translation is: 01/01/80

    MDIRA Command
   ?
```

APPENDIX B

Driver Incident Codebook

APPENDIX B
Driver Incident Codebook

Variable Number	Variable Name	Field Width	Character Type	Mult Resp	Page Number
1	DRIVER NUMBER	6	Numeric		63
2	INCIDENT NUMBER	2	Numeric		63
3	LICENSE CODE	2	Numeric		64
4	ORIG LICENSE ISSUE DATE	5	Numeric		65
5	BIRTH DATE	6	Numeric		65
6	AGE ON INCIDENT DATE	4	Numeric		66
7	COUNTY OF RESIDENCE	2	Numeric		66
8	SEX	1	Numeric		68
9	INCIDENT CODE	4	Numeric		68
10	POINTS FROM INCIDENT	1	Numeric		77
11	INCIDENT DATE	5	Numeric		78
12	VEHICLE TYPE	2	Numeric		78

This codebook documents a data set built from a dump of the Michigan Driver Incident File of all drivers with an original license issue date in 1975. The dump was made on January 23, 1982. The data set was built on January 31 and includes all incidents up to the present date.

Variable 1	DRIVER NUMBER	MD1: None	Field Width: 6
		MD2: None	Type: Numeric

ASSIGNED SEQUENTIALLY BY HSRI

Variable 2	INCIDENT NUMBER	MD1: None	Field Width: 2
		MD2: None	Type: Numeric

FREQ	Prcnt	INCIDENT NUMBER
39194	5.0	00. No incidents for driver
140202	17.8	01. Incident #1
108182	13.7	02. Incident #2
85913	10.9	03. Incident #3
69552	8.8	04. Incident #4
56516	7.2	05. Incident #5
46033	5.8	06. Incident #6
37762	4.8	07. Incident #7
31025	3.9	08. Incident #8
25735	3.3	09. Incident #9
21532	2.7	10. Incident #10
18092	2.3	11. Incident #11
15346	1.9	12. Incident #12
13069	1.7	13. Incident #13
11159	1.4	14. Incident #14
9601	1.2	15. Incident #15
8277	1.0	16. Incident #16
7161	0.9	17. Incident #17
6190	0.8	18. Incident #18
5381	0.7	19. Incident #19
4679	0.6	20. Incident #20
4049	0.5	21. Incident #21
3537	0.4	22. Incident #22
3056	0.4	23. Incident #23
2635	0.3	24. Incident #24
2263	0.3	25. Incident #25
1952	0.2	26. Incident #26
1689	0.2	27. Incident #27
1417	0.2	28. Incident #28
1200	0.2	29. Incident #29

FREQ	Prcnt	Var 2	INCIDENT NUMBER
1020	0.1	30.	Incident #30
868	0.1	31.	Incident #31
732	0.1	32.	Incident #32
636	0.1	33.	Incident #33
530	0.1	34.	Incident #34
445	0.1	35.	Incident #35
375	0.0	36.	Incident #36
330	0.0	37.	Incident #37
276	0.0	38.	Incident #38
240	0.0	39.	Incident #39
205	0.0	40.	Incident #40
166	0.0	41.	Incident #41
132	0.0	42.	Incident #42
108	0.0	43.	Incident #43
90	0.0	44.	Incident #44
71	0.0	45.	Incident #45
58	0.0	46.	Incident #46
45	0.0	47.	Incident #47
38	0.0	48.	Incident #48
33	0.0	49.	Incident #49
29	0.0	50.	Incident #50
27	0.0	51.	Incident #51
25	0.0	52.	Incident #52
21	0.0	53.	Incident #53
19	0.0	54.	Incident #54
15	0.0	55.	Incident #55
13	0.0	56.	Incident #56
11	0.0	57.	Incident #57
10	0.0	58.	Incident #58
8	0.0	59.	Incident #59
6	0.0	60.	Incident #60
6	0.0	61.	Incident #61
6	0.0	62.	Incident #62
5	0.0	63.	Incident #63
3	0.0	64.	Incident #64
3	0.0	65.	Incident #65
2	0.0	66.	Incident #66
1	0.0	67.	Incident #67
1	0.0	68.	Incident #68

Variable 3	LICENSE CODE	MD1: 99	Field Width: 2
		MD2: None	Type: Numeric

FREQ	Prcnt	LICENSE CODE
160	0.0	10. None on record
0	0.0	20. Special restricted
2	0.0	21. Minor restricted
29	0.0	31. Operator probationary

FREQ	Prcnt	Var 3 LICENSE CODE
16208	2.1	32. Operator original, not probationary
427078	54.1	33. Operator renewal
155877	19.8	34. Operator duplicate
25967	3.3	35. Operator corrected
0	0.0	36. Operator/cycle probationary
190	0.0	37. Operator/cycle original, not probationary
51596	6.5	38. Operator/cycle renewal
18000	2.3	39. Operator/cycle duplicate
19207	2.4	40. Operator/cycle corrected
0	0.0	41. Chauffeur probationary
10672	1.4	42. Chauffeur original, not probationary
31658	4.0	43. Chauffeur renewal
14239	1.8	44. Chauffeur duplicate
2208	0.3	45. Chauffeur corrected
0	0.0	46. Chauffeur/cycle probationary
2055	0.3	47. Chauffeur/cycle original, not probationary
7385	0.9	48. Chauffeur/cycle renewal
3307	0.4	49. Chauffeur/cycle duplicate
3170	0.4	50. Chauffeur/cycle corrected
0	0.0	99. Other or missing data

Variable 4	ORIG LICENSE ISSUE DATE	MD1: 0	Field Width: 5
		MD2: None	Type: Numeric

FREQ Prcnt ORIGINAL LICENSE ISSUE DATE (JULIAN FORMAT)

00000. Unknown
- .
99999.

Variable 5	BIRTH DATE	MD1: 0	Field Width: 6
		MD2: None	Type: Numeric

FREQ Prcnt BIRTH DATE (YYMMDD FORMAT)

000000. Unknown
- .
991231.

Variable 6	AGE ON INCIDENT DATE	MD1: 9999	Field Width: 4
		MD2: None	Type: Numeric

FREQ	Prcnt	AGE ON INCIDENT DATE (YYMM FORMAT)
33	0.0	0000.
		- .
46004	5.8	9999. Not applicable (no incident) or missing data

Variable 7	COUNTY OF RESIDENCE	MD1: 99	Field Width: 2
		MD2: None	Type: Numeric

FREQ	Prcnt	COUNTY OF RESIDENCE
459	0.1	01. Alcona
503	0.1	02. Alger
4606	0.6	03. Allegan
2522	0.3	04. Alpena
920	0.1	05. Antrim
1266	0.2	06. Arenac
399	0.1	07. Baraga
2665	0.3	08. Barry
9810	1.2	09. Bay
712	0.1	10. Benzie
13547	1.7	11. Berrien
2582	0.3	12. Branch
11880	1.5	13. Calhoun
2437	0.3	14. Cass
1377	0.2	15. Charlevoix
1138	0.1	16. Cheboygan
1776	0.2	17. Chippewa
1759	0.2	18. Clare
6459	0.8	19. Clinton
650	0.1	20. Crawford
2998	0.4	21. Delta
1403	0.2	22. Dickinson
7432	0.9	23. Eaton
1630	0.2	24. Emmet
38841	4.9	25. Genesee
1199	0.2	26. Gladwin
808	0.1	27. Gogebic
4750	0.6	28. Grand Traverse
2679	0.3	29. Gratiot
2430	0.3	30. Hillsdale
1364	0.2	31. Houghton
2244	0.3	32. Huron
24483	3.1	33. Ingham
3817	0.5	34. Ionia
2022	0.3	35. Iosco
707	0.1	36. Iron
3138	0.4	37. Isabella

FREQ	Prcnt	Var 7	COUNTY OF RESIDENCE
10988	1.4	38.	Jackson
20392	2.6	39.	Kalamazoo
923	0.1	40.	Kalkaska
45376	5.8	41.	Kent
519	0.1	42.	Keweenaw
318	0.0	43.	Lake
4873	0.6	44.	Lapeer
585	0.1	45.	Leelanau
5998	0.8	46.	Lenawee
6708	0.9	47.	Livingston
257	0.0	48.	Luce
402	0.1	49.	Mackinac
70619	9.0	50.	Macomb
1583	0.2	51.	Manistee
5694	0.7	52.	Marquette
1631	0.2	53.	Mason
2187	0.3	54.	Mecosta
1479	0.2	55.	Menominee
4703	0.6	56.	Midland
541	0.1	57.	Missaukee
8115	1.0	58.	Monroe
3516	0.4	59.	Montcalm
474	0.1	60.	Montmorency
12041	1.5	61.	Muskegon
2028	0.3	62.	Newaygo
81424	10.3	63.	Oakland
926	0.1	64.	Oceana
1071	0.1	65.	Ogemaw
523	0.1	66.	Ontonagon
1150	0.1	67.	Osceola
448	0.1	68.	Oscoda
1294	0.2	69.	Otsego
12183	1.5	70.	Ottawa
817	0.1	71.	Presque
1056	0.1	72.	Roscommon
18480	2.3	73.	Saginaw
12036	1.5	74.	St. Clair
3701	0.5	75.	St. Joseph
2720	0.3	76.	Sanilac
598	0.1	77.	Schoolcraft
4718	0.6	78.	Shiawassee
3774	0.5	79.	Tuscola
4400	0.6	80.	Van Buren
22059	2.8	81.	Washtenaw
217221	27.5	82.	Wayne
1705	0.2	83.	Wexford
15342	1.9	99.	Missing data

Variable 8	SEX	MD1: 9	Field Width: 1
		MD2: None	Type: Numeric

FREQ	Prcnt	SEX
596603	75.6	1. Male
192405	24.4	2. Female
0	0.0	9. Missing data

Variable 9	INCIDENT CODE	MD1: None	Field Width: 4
		MD2: None	Type: Numeric

FREQ	Prcnt	INCIDENT CODE (CONVICTION, ACCIDENT, OR ACTION)
39194	5.0	0000. No incidents (conviction, accident, or action) on file for driver
		CONVICTION RECORD INCIDENT
34235	4.3	0019. Energy speed (less than old posted speed)
3767	0.5	0119. Energy speed (after 4/1/81, between 60 and old posted speed)
1030	0.1	0214. Cycle-improper or no safety equipment
2587	0.3	0215. Drove without proper endorsement
18508	2.3	0216. Violation basic speed law
66	0.0	0217. Failed to drive minimum speed
83828	10.6	0218. Speed (no amount or up through 10 mph)
16205	2.1	0219. Energy speed (up through 10 mph)
490	0.1	0220. Drag racing (before 8/1/79)
264	0.0	0221. Failed to report accident
4948	0.6	0225. Disobeyed traffic control device
251	0.0	0226. Disobeyed policeman signal
509	0.1	0227. Improper crossing - divided highway
458	0.1	0228. Fleeing and eluding officer (before 8/1/79)
1	0.0	0229. Unlawful driving away auto w/o intent to steal (attempted)
11678	1.5	0230. Failed to yield
16	0.0	0232. Drove while impaired (attempted)
11	0.0	0234. Improper operation - emergency vehicle or school bus
37	0.0	0236. Interfered with fire apparatus
936	0.1	0237. Failed to stop leaving alley or private drive
1411	0.2	0239. Following too close
4529	0.6	0240. Failed to signal and/or observe
14792	1.9	0244. Prohibited turn
5594	0.7	0245. Improper turn
2993	0.4	0246. Drove wrong way on 1-way street
4121	0.5	0247. Improper lane use
3115	0.4	0248. Drove left of center
37	0.0	0249. Illegal entrance or exit - expressway
1499	0.2	0251. Improper use of lights

FREQ	Prcnt	Var 9	INCIDENT CODE
3	0.0	0255.	Negligent homicide (attempted)
1	0.0	0256.	Felonious driving (attempted)
9	0.0	0257.	Driving under influence liquor (attempted)
4	0.0	0259.	Failed to stop or identify after P.I. accident (attempted)
15	0.0	0260.	Failed to stop or identify after P.D. accident (attempted)
32	0.0	0261.	Reckless driving (attempted)
3	0.0	0262.	Fleeing or eluding officer (attempted)
183	0.0	0265.	Improper size, load, or towing
801	0.1	0266.	Obstructed vision or control
3	0.0	0267.	Drove moped on sidewalk
8	0.0	0268.	Unlawful rider on motorcycle/moped
3	0.0	0269.	Motorcycle/moped over two abreast
4056	0.5	0270.	Drove while license expired
19081	2.4	0271.	No valid license in possession
2	0.0	0272.	Disobeyed traffic signal (attempted)
8422	1.1	0274.	Drove while license suspended, etc.
191	0.0	0276.	Violation of instruction permit
291	0.0	0277.	Violation of license restrictions
995	0.1	0278.	Drove without corrective lenses
467	0.1	0281.	Violation of restricted license
14	0.0	0282.	Violation of financial responsibility law
39028	4.9	0318.	Speed (up through 15 mph)
1743	0.2	0319.	Energy speed (up through 15 mph)
2762	0.4	0322.	Careless driving (8/1/79 or later)
32912	4.2	0384.	Disobeyed traffic signal
465	0.1	0387.	Failed to stop for school bus
20598	2.6	0388.	Disobeyed stop sign
3581	0.5	0389.	Improper passing
22251	2.8	0418.	Speed (over 15 mph)
1101	0.1	0419.	Energy speed (over 15 mph)
85	0.0	0420.	Drag racing (8/1/79 or later)
7742	1.0	0422.	Careless driving (before 8/1/79)
6290	0.8	0423.	Drove while impaired
179	0.0	0428.	Fleeing and eluding officer (8/1/79 or later)
27	0.0	0602.	Negligent homicide
15	0.0	0603.	Felony - auto used
4	0.0	0604.	Felonious driving
20	0.0	0606.	Unlawful driving away auto
3142	0.4	0607.	Driving under influence liquor
28	0.0	0608.	Driving under influence controlled substance
94	0.0	0609.	Failed to stop or identify after P.I. accident
1279	0.2	0610.	Failed to stop or identify after P.D. accident
2724	0.3	0611.	Reckless driving
1	0.0	0652.	Manslaughter (attempted)
3	0.0	0654.	Unlawful driving away auto (attempted)

FREQ	Prcnt	Var 9 INCIDENT CODE
0	0.0	0818. Speed (undetermined points)
223	0.0	0890. Unlawful use or display of license
1	0.0	0891. False info or fraud in obtaining license
12	0.0	0892. Allowed intoxicated person to drive
1053	0.1	0896. No proof of insurance
9	0.0	0897. Displayed or permitted display of suspended or revoked license
		ACCIDENT RECORD INCIDENTS
193	0.0	1561. No conv, HBD unk, single veh, prop damage, no V
175	0.0	1562. No conv, HBD unk, single veh, injury, no V
3	0.0	1563. No conv, HBD unk, single veh, fatal, no V
3121	0.4	1564. No conv, HBD unk, multiple veh, prop damage, no V
979	0.1	1565. No conv, HBD unk, multiple veh, injury, no V
6	0.0	1566. No conv, HBD unk, multiple veh, fatal, no V
35	0.0	1569. No conv, HBD unk, veh count and/or type unk, no V
6032	0.8	1571. No conv, non HBD, single veh, prop damage, no V
2991	0.4	1572. No conv, non HBD, single veh, injury, no V
73	0.0	1573. No conv, non HBD, single veh, fatal, no V
41108	5.2	1574. No conv, non HBD, multiple veh, prop damage, no V
16556	2.1	1575. No conv, non HBD, multiple veh, injury, no V
132	0.0	1576. No conv, non HBD, multiple veh, fatal, no V
422	0.1	1579. No conv, non HBD, veh count and/or type unk, no V
674	0.1	1581. No conv, HBD, single veh, prop damage, no V
590	0.1	1582. No conv, HBD, single veh, injury, no V
19	0.0	1583. No conv, HBD, single veh, fatal, no V
1338	0.2	1584. No conv, HBD, multiple veh, prop damage, no V
955	0.1	1585. No conv, HBD, multiple veh, injury, no V
30	0.0	1586. No conv, HBD, multiple veh, fatal, no V
31	0.0	1589. No conv, HBD, veh count and/or type unk, no V
61	0.0	1611. No conv, HBD unk, single veh, prop damage, V1
3	0.0	1613. No conv, HBD unk, single veh, prop damage, V3
4	0.0	1615. No conv, HBD unk, single veh, prop damage, V5
2	0.0	1616. No conv, HBD unk, single veh, prop damage, V6
22	0.0	1618. No conv, HBD unk, single veh, prop damage, V8
26	0.0	1619. No conv, HBD unk, single veh, prop damage, V11
55	0.0	1621. No conv, HBD unk, single veh, injury, V1
6	0.0	1623. No conv, HBD unk, single veh, injury, V3
2	0.0	1625. No conv, HBD unk, single veh, injury, V5
1	0.0	1626. No conv, HBD unk, single veh, injury, V6
1	0.0	1627. No conv, HBD unk, single veh, injury, V7
17	0.0	1628. No conv, HBD unk, single veh, injury, V8
11	0.0	1629. No conv, HBD unk, single veh, injury, V11
3	0.0	1631. No conv, HBD unk, single veh, fatal, V1
1	0.0	1635. No conv, HBD unk, single veh, fatal, V5

FREQ	Prcnt	Var 9 INCIDENT CODE
1	0.0	1639. No conv, HBD unk, single veh, fatal, V11
200	0.0	1641. No conv, HBD unk, multiple veh, prop damage, V1
220	0.0	1643. No conv, HBD unk, multiple veh, prop damage, V3
1	0.0	1644. No conv, HBD unk, multiple veh, prop damage, V4
113	0.0	1645. No conv, HBD unk, multiple veh, prop damage, V5
63	0.0	1646. No conv, HBD unk, multiple veh, prop damage, V6
76	0.0	1647. No conv, HBD unk, multiple veh, prop damage, V7
229	0.0	1648. No conv, HBD unk, multiple veh, prop damage, V8
186	0.0	1649. No conv, HBD unk, multiple veh, prop damage, V11
74	0.0	1651. No conv, HBD unk, multiple veh, injury, V1
1	0.0	1652. No conv, HBD unk, multiple veh, injury, V2
98	0.0	1653. No conv, HBD unk, multiple veh, injury, V3
37	0.0	1655. No conv, HBD unk, multiple veh, injury, V5
10	0.0	1656. No conv, HBD unk, multiple veh, injury, V6
3	0.0	1657. No conv, HBD unk, multiple veh, injury, V7
72	0.0	1658. No conv, HBD unk, multiple veh, injury, V8
24	0.0	1659. No conv, HBD unk, multiple veh, injury, V11
2	0.0	1661. No conv, HBD unk, multiple veh, fatal, V1
2	0.0	1663. No conv, HBD unk, multiple veh, fatal, V3
1	0.0	1665. No conv, HBD unk, multiple veh, fatal, V5
5	0.0	1691. No conv, HBD unk, veh count and/or type unk, V1
1	0.0	1693. No conv, HBD unk, veh count and/or type unk, V3
2	0.0	1696. No conv, HBD unk, veh count and/or type unk, V6
1	0.0	1698. No conv, HBD unk, veh count and/or type unk, V8
7	0.0	1699. No conv, HBD unk, veh count and/or type unk, V11
3439	0.4	1711. No conv, non HBD, single veh, prop damage, V1
9	0.0	1712. No conv, non HBD, single veh, prop damage, V2
69	0.0	1713. No conv, non HBD, single veh, prop damage, V3
129	0.0	1715. No conv, non HBD, single veh, prop damage, V5
62	0.0	1716. No conv, non HBD, single veh, prop damage, V6
35	0.0	1717. No conv, non HBD, single veh, prop damage, V7
853	0.1	1718. No conv, non HBD, single veh, prop damage, V8
598	0.1	1719. No conv, non HBD, single veh, prop damage, V11
1820	0.2	1721. No conv, non HBD, single veh, injury, V1
3	0.0	1722. No conv, non HBD, single veh, injury, V2
173	0.0	1723. No conv, non HBD, single veh, injury, V3
2	0.0	1724. No conv, non HBD, single veh, injury, V4
119	0.0	1725. No conv, non HBD, single veh, injury, V5
39	0.0	1726. No conv, non HBD, single veh, injury, V6
19	0.0	1727. No conv, non HBD, single veh, injury, V7
570	0.1	1728. No conv, non HBD, single veh, injury, V8
371	0.0	1729. No conv, non HBD, single veh, injury, V11
5	0.0	1731. No conv, non HBD, single veh, fatal, V1
1	0.0	1738. No conv, non HBD, single veh, fatal, V8

FREQ	Prcnt	Var 9 INCIDENT CODE
2	0.0	1739. No conv, non HBD, single veh, fatal, V11
1	0.0	1740. No conv, non HBD, multiple veh, prop damage, V?
6198	0.8	1741. No conv, non HBD, multiple veh, prop damage, V1
23	0.0	1742. No conv, non HBD, multiple veh, prop damage, V2
7076	0.9	1743. No conv, non HBD, multiple veh, prop damage, V3
15	0.0	1744. No conv, non HBD, multiple veh, prop damage, V4
2688	0.3	1745. No conv, non HBD, multiple veh, prop damage, V5
1568	0.2	1746. No conv, non HBD, multiple veh, prop damage, V6
1434	0.2	1747. No conv, non HBD, multiple veh, prop damage, V7
5674	0.7	1748. No conv, non HBD, multiple veh, prop damage, V8
3810	0.5	1749. No conv, non HBD, multiple veh, prop damage, V11
2252	0.3	1751. No conv, non HBD, multiple veh, injury, V1
9	0.0	1752. No conv, non HBD, multiple veh, injury, V2
2844	0.4	1753. No conv, non HBD, multiple veh, injury, V3
6	0.0	1754. No conv, non HBD, multiple veh, injury, V4
763	0.1	1755. No conv, non HBD, multiple veh, injury, V5
428	0.1	1756. No conv, non HBD, multiple veh, injury, V6
68	0.0	1757. No conv, non HBD, multiple veh, injury, V7
2062	0.3	1758. No conv, non HBD, multiple veh, injury, V8
815	0.1	1759. No conv, non HBD, multiple veh, injury, V11
11	0.0	1761. No conv, non HBD, multiple veh, fatal, V1
17	0.0	1763. No conv, non HBD, multiple veh, fatal, V3
17	0.0	1765. No conv, non HBD, multiple veh, fatal, V5
2	0.0	1766. No conv, non HBD, multiple veh, fatal, V6
4	0.0	1768. No conv, non HBD, multiple veh, fatal, V8
2	0.0	1769. No conv, non HBD, multiple veh, fatal, V11
173	0.0	1791. No conv, non HBD, veh count and/or type unk, V1
81	0.0	1793. No conv, non HBD, veh count and/or type unk, V3
27	0.0	1795. No conv, non HBD, veh count and/or type unk, V5
20	0.0	1796. No conv, non HBD, veh count and/or type unk, V6
5	0.0	1797. No conv, non HBD, veh count and/or type unk, V7
30	0.0	1798. No conv, non HBD, veh count and/or type unk, V8
80	0.0	1799. No conv, non HBD, veh count and/or type unk, V11
1043	0.1	1811. No conv, HBD, single veh, prop damage, V1
6	0.0	1812. No conv, HBD, single veh, prop damage, V2
18	0.0	1813. No conv, HBD, single veh, prop damage, V3
87	0.0	1815. No conv, HBD, single veh, prop damage, V5
29	0.0	1816. No conv, HBD, single veh, prop damage, V6
12	0.0	1817. No conv, HBD, single veh, prop damage, V7
417	0.1	1818. No conv, HBD, single veh, prop damage, V8
166	0.0	1819. No conv, HBD, single veh, prop damage, V11
1060	0.1	1821. No conv, HBD, single veh, injury, V1
1	0.0	1822. No conv, HBD, single veh, injury, V2
21	0.0	1823. No conv, HBD, single veh, injury, V3
1	0.0	1824. No conv, HBD, single veh, injury, V4

FREQ	Prcnt	Var 9 INCIDENT CODE
105	0.0	1825. No conv, HBD, single veh, injury, V5
11	0.0	1826. No conv, HBD, single veh, injury, V6
2	0.0	1827. No conv, HBD, single veh, injury, V7
423	0.1	1828. No conv, HBD, single veh, injury, V8
184	0.0	1829. No conv, HBD, single veh, injury, V11
22	0.0	1831. No conv, HBD, single veh, fatal, V1
1	0.0	1833. No conv, HBD, single veh, fatal, V3
2	0.0	1835. No conv, HBD, single veh, fatal, V5
3	0.0	1838. No conv, HBD, single veh, fatal, V8
1	0.0	1839. No conv, HBD, single veh, fatal, V11
435	0.1	1841. No conv, HBD, multiple veh, prop damage, V1
3	0.0	1842. No conv, HBD, multiple veh, prop damage, V2
263	0.0	1843. No conv, HBD, multiple veh, prop damage, V3
3	0.0	1844. No conv, HBD, multiple veh, prop damage, V4
283	0.0	1845. No conv, HBD, multiple veh, prop damage, V5
96	0.0	1846. No conv, HBD, multiple veh, prop damage, V6
114	0.0	1847. No conv, HBD, multiple veh, prop damage, V7
465	0.1	1848. No conv, HBD, multiple veh, prop damage, V8
247	0.0	1849. No conv, HBD, multiple veh, prop damage, V11
271	0.0	1851. No conv, HBD, multiple veh, injury, V1
1	0.0	1852. No conv, HBD, multiple veh, injury, V2
259	0.0	1853. No conv, HBD, multiple veh, injury, V3
2	0.0	1854. No conv, HBD, multiple veh, injury, V4
197	0.0	1855. No conv, HBD, multiple veh, injury, V5
36	0.0	1856. No conv, HBD, multiple veh, injury, V6
7	0.0	1857. No conv, HBD, multiple veh, injury, V7
320	0.0	1858. No conv, HBD, multiple veh, injury, V8
95	0.0	1859. No conv, HBD, multiple veh, injury, V11
9	0.0	1861. No conv, HBD, multiple veh, fatal, V1
8	0.0	1863. No conv, HBD, multiple veh, fatal, V3
19	0.0	1865. No conv, HBD, multiple veh, fatal, V5
2	0.0	1868. No conv, HBD, multiple veh, fatal, V8
23	0.0	1891. No conv, HBD, veh count and/or type unk, V1
10	0.0	1893. No conv, HBD, veh count and/or type unk, V3
6	0.0	1895. No conv, HBD, veh count and/or type unk, V5
3	0.0	1896. No conv, HBD, veh count and/or type unk, V6
4	0.0	1898. No conv, HBD, veh count and/or type unk, V8
13	0.0	1899. No conv, HBD, veh count and/or type unk, V11
2	0.0	1997. No conviction, unknown accident record
14	0.0	2561. 1+ conv, HBD unk, single veh, prop damage, no V
11	0.0	2562. 1+ conv, HBD unk, single veh, injury, no V
84	0.0	2564. 1+ conv, HBD unk, multiple veh, prop damage, no V
42	0.0	2565. 1+ conv, HBD unk, multiple veh, injury, no V
3	0.0	2569. 1+ conv, HBD unk, veh count and/or type unk, no V

FREQ	Prcnt	Var 9 INCIDENT CODE
157	0.0	2571. 1+ conv, non HBD, single veh, prop damage, no V
105	0.0	2572. 1+ conv, non HBD, single veh, injury, no V
3	0.0	2573. 1+ conv, non HBD, single veh, fatal, no V
856	0.1	2574. 1+ conv, non HBD, multiple veh, prop damage, no V
475	0.1	2575. 1+ conv, non HBD, multiple veh, injury, no V
5	0.0	2576. 1+ conv, non HBD, multiple veh, fatal, no V
23	0.0	2579. 1+ conv, non HBD, veh count and/or type unk, no V
248	0.0	2581. 1+ conv, HBD, single veh, prop damage, no V
212	0.0	2582. 1+ conv, HBD, single veh, injury, no V
2	0.0	2583. 1+ conv, HBD, single veh, fatal, no V
294	0.0	2584. 1+ conv, HBD, multiple veh, prop damage, no V
200	0.0	2585. 1+ conv, HBD, multiple veh, injury, no V
2	0.0	2586. 1+ conv, HBD, multiple veh, fatal, no V
5	0.0	2589. 1+ conv, HBD, veh count and/or type unk, no V
15	0.0	2611. 1+ conv, HBD unk, single veh, prop damage, V1
1	0.0	2613. 1+ conv, HBD unk, single veh, prop damage, V3
2	0.0	2615. 1+ conv, HBD unk, single veh, prop damage, V5
6	0.0	2618. 1+ conv, HBD unk, single veh, prop damage, V8
6	0.0	2619. 1+ conv, HBD unk, single veh, prop damage, V11
15	0.0	2621. 1+ conv, HBD unk, single veh, injury, V1
2	0.0	2623. 1+ conv, HBD unk, single veh, injury, V3
4	0.0	2628. 1+ conv, HBD unk, single veh, injury, V8
4	0.0	2629. 1+ conv, HBD unk, single veh, injury, V11
52	0.0	2641. 1+ conv, HBD unk, multiple veh, prop damage, V1
59	0.0	2643. 1+ conv, HBD unk, multiple veh, prop damage, V3
26	0.0	2645. 1+ conv, HBD unk, multiple veh, prop damage, V5
8	0.0	2646. 1+ conv, HBD unk, multiple veh, prop damage, V6
5	0.0	2647. 1+ conv, HBD unk, multiple veh, prop damage, V7
36	0.0	2648. 1+ conv, HBD unk, multiple veh, prop damage, V8
41	0.0	2649. 1+ conv, HBD unk, multiple veh, prop damage, V11
29	0.0	2651. 1+ conv, HBD unk, multiple veh, injury, V1
40	0.0	2653. 1+ conv, HBD unk, multiple veh, injury, V3
14	0.0	2655. 1+ conv, HBD unk, multiple veh, injury, V5
5	0.0	2656. 1+ conv, HBD unk, multiple veh, injury, V6
23	0.0	2658. 1+ conv, HBD unk, multiple veh, injury, V8
8	0.0	2659. 1+ conv, HBD unk, multiple veh, injury, V11
2	0.0	2691. 1+ conv, HBD unk, veh count and/or type unk, V1
2	0.0	2693. 1+ conv, HBD unk, veh count and/or type unk, V3
1	0.0	2699. 1+ conv, HBD unk, veh count and/or type unk, V11
1130	0.1	2711. 1+ conv, non HBD, single veh, prop damage, V1
2	0.0	2712. 1+ conv, non HBD, single veh, prop damage, V2
31	0.0	2713. 1+ conv, non HBD, single veh, prop damage, V3

FREQ	Prcnt	Var 9 INCIDENT CODE
51	0.0	2715. 1+ conv, non HBD, single veh, prop damage, V5
20	0.0	2716. 1+ conv, non HBD, single veh, prop damage, V6
3	0.0	2717. 1+ conv, non HBD, single veh, prop damage, V7
220	0.0	2718. 1+ conv, non HBD, single veh, prop damage, V8
153	0.0	2719. 1+ conv, non HBD, single veh, prop damage, V11
639	0.1	2721. 1+ conv, non HBD, single veh, injury, V1
1	0.0	2722. 1+ conv, non HBD, single veh, injury, V2
99	0.0	2723. 1+ conv, non HBD, single veh, injury, V3
1	0.0	2724. 1+ conv, non HBD, single veh, injury, V4
43	0.0	2725. 1+ conv, non HBD, single veh, injury, V5
6	0.0	2726. 1+ conv, non HBD, single veh, injury, V6
2	0.0	2727. 1+ conv, non HBD, single veh, injury, V7
123	0.0	2728. 1+ conv, non HBD, single veh, injury, V8
101	0.0	2729. 1+ conv, non HBD, single veh, injury, V11
6	0.0	2731. 1+ conv, non HBD, single veh, fatal, V1
1	0.0	2733. 1+ conv, non HBD, single veh, fatal, V3
1	0.0	2739. 1+ conv, non HBD, single veh, fatal, V11
3638	0.5	2741. 1+ conv, non HBD, multiple veh, prop damage, V1
17	0.0	2742. 1+ conv, non HBD, multiple veh, prop damage, V2
6456	0.8	2743. 1+ conv, non HBD, multiple veh, prop damage, V3
15	0.0	2744. 1+ conv, non HBD, multiple veh, prop damage, V4
1655	0.2	2745. 1+ conv, non HBD, multiple veh, prop damage, V5
1076	0.1	2746. 1+ conv, non HBD, multiple veh, prop damage, V6
222	0.0	2747. 1+ conv, non HBD, multiple veh, prop damage, V7
3052	0.4	2748. 1+ conv, non HBD, multiple veh, prop damage, V8
1251	0.2	2749. 1+ conv, non HBD, multiple veh, prop damage, V11
1830	0.2	2751. 1+ conv, non HBD, multiple veh, injury, V1
1	0.0	2752. 1+ conv, non HBD, multiple veh, injury, V2
3604	0.5	2753. 1+ conv, non HBD, multiple veh, injury, V3
4	0.0	2754. 1+ conv, non HBD, multiple veh, injury, V4
642	0.1	2755. 1+ conv, non HBD, multiple veh, injury, V5
383	0.0	2756. 1+ conv, non HBD, multiple veh, injury, V6
25	0.0	2757. 1+ conv, non HBD, multiple veh, injury, V7
1702	0.2	2758. 1+ conv, non HBD, multiple veh, injury, V8
282	0.0	2759. 1+ conv, non HBD, multiple veh, injury, V11
8	0.0	2763. 1+ conv, non HBD, multiple veh, fatal, V3
6	0.0	2765. 1+ conv, non HBD, multiple veh, fatal, V5
1	0.0	2769. 1+ conv, non HBD, multiple veh, fatal, V11
104	0.0	2791. 1+ conv, non HBD, veh count and/or type unk, V1
100	0.0	2793. 1+ conv, non HBD, veh count and/or type unk, V3
29	0.0	2795. 1+ conv, non HBD, veh count and/or type unk, V5
14	0.0	2796. 1+ conv, non HBD, veh count and/or type unk, V6
1	0.0	2797. 1+ conv, non HBD, veh count and/or type unk, V7
12	0.0	2798. 1+ conv, non HBD, veh count and/or type unk, V8
35	0.0	2799. 1+ conv, non HBD, veh count and/or type unk, V11

FREQ	Prcnt	Var 9 INCIDENT CODE
923	0.1	2811. 1+ conv, HBD, single veh, prop damage, V1
4	0.0	2812. 1+ conv, HBD, single veh, prop damage, V2
18	0.0	2813. 1+ conv, HBD, single veh, prop damage, V3
4	0.0	2814. 1+ conv, HBD, single veh, prop damage, V4
84	0.0	2815. 1+ conv, HBD, single veh, prop damage, V5
23	0.0	2816. 1+ conv, HBD, single veh, prop damage, V6
12	0.0	2817. 1+ conv, HBD, single veh, prop damage, V7
313	0.0	2818. 1+ conv, HBD, single veh, prop damage, V8
156	0.0	2819. 1+ conv, HBD, single veh, prop damage, V11
913	0.1	2821. 1+ conv, HBD, single veh, injury, V1
3	0.0	2822. 1+ conv, HBD, single veh, injury, V2
32	0.0	2823. 1+ conv, HBD, single veh, injury, V3
1	0.0	2824. 1+ conv, HBD, single veh, injury, V4
74	0.0	2825. 1+ conv, HBD, single veh, injury, V5
6	0.0	2826. 1+ conv, HBD, single veh, injury, V6
1	0.0	2827. 1+ conv, HBD, single veh, injury, V7
319	0.0	2828. 1+ conv, HBD, single veh, injury, V8
130	0.0	2829. 1+ conv, HBD, single veh, injury, V11
5	0.0	2831. 1+ conv, HBD, single veh, fatal, V1
1	0.0	2833. 1+ conv, HBD, single veh, fatal, V3
1	0.0	2835. 1+ conv, HBD, single veh, fatal, V5
2	0.0	2838. 1+ conv, HBD, single veh, fatal, V8
533	0.1	2841. 1+ conv, HBD, multiple veh, prop damage, V1
2	0.0	2842. 1+ conv, HBD, multiple veh, prop damage, V2
393	0.0	2843. 1+ conv, HBD, multiple veh, prop damage, V3
6	0.0	2844. 1+ conv, HBD, multiple veh, prop damage, V4
416	0.1	2845. 1+ conv, HBD, multiple veh, prop damage, V5
104	0.0	2846. 1+ conv, HBD, multiple veh, prop damage, V6
45	0.0	2847. 1+ conv, HBD, multiple veh, prop damage, V7
534	0.1	2848. 1+ conv, HBD, multiple veh, prop damage, V8
272	0.0	2849. 1+ conv, HBD, multiple veh, prop damage, V11
412	0.1	2851. 1+ conv, HBD, multiple veh, injury, V1
2	0.0	2852. 1+ conv, HBD, multiple veh, injury, V2
538	0.1	2853. 1+ conv, HBD, multiple veh, injury, V3
10	0.0	2854. 1+ conv, HBD, multiple veh, injury, V4
330	0.0	2855. 1+ conv, HBD, multiple veh, injury, V5
45	0.0	2856. 1+ conv, HBD, multiple veh, injury, V6
6	0.0	2857. 1+ conv, HBD, multiple veh, injury, V7
432	0.1	2858. 1+ conv, HBD, multiple veh, injury, V8
120	0.0	2859. 1+ conv, HBD, multiple veh, injury, V11
1	0.0	2861. 1+ conv, HBD, multiple veh, fatal, V1
2	0.0	2863. 1+ conv, HBD, multiple veh, fatal, V3
2	0.0	2865. 1+ conv, HBD, multiple veh, fatal, V5
1	0.0	2868. 1+ conv, HBD, multiple veh, fatal, V8
35	0.0	2891. 1+ conv, HBD, veh count and/or type unk, V1

FREQ	Prcnt	Var 9 INCIDENT CODE
7	0.0	2893. 1+ conv, HBD, veh count and/or type unk, V3
4	0.0	2895. 1+ conv, HBD, veh count and/or type unk, V5
3	0.0	2896. 1+ conv, HBD, veh count and/or type unk, V6
1	0.0	2898. 1+ conv, HBD, veh count and/or type unk, V8
18	0.0	2899. 1+ conv, HBD, veh count and/or type unk, V11
		ACTION RECORD INCIDENTS
47425	6.0	3901. 4-7 point warning letter
19779	2.5	3902. 8-11 point warning letter
330	0.0	3903. VLR warning letter
7	0.0	3904. PROB warning letter
16669	2.1	3905. RFS (reex info/susp)
3761	0.5	3906. NSS (no show/int susp)
2294	0.3	3907. INTS (internal susp)
929	0.1	3908. ADDS (additional susp)
3417	0.4	3909. RFR (reex info/revok)
1191	0.2	3910. CAR (fn coa/int rev)
79	0.0	3911. INTR (internal rev)
1566	0.2	3912. RFLR (reex inf/l res)
249	0.0	3913. INLR (internal l res)
5930	0.8	3914. RFNA (reex inf/talk)
1017	0.1	3915. MINT (misc internal)
53	0.0	3916. RFFA (reex inf/fatal)
1102	0.1	3917. RFRR (reex inf/rev l)
74461	9.4	3998. Other action record

Variable 10	POINTS FROM INCIDENT	MD1: 9	Field Width: 1
		MD2: None	Type: Numeric

FREQ	Prcnt	POINTS FROM INCIDENT
42640	5.4	0. 0 points
3731	0.5	1. 1 point
207133	26.3	2. 2 points
100568	12.7	3. 3 points
37445	4.7	4. 4 points
7161	0.9	6. 6 points
1194	0.2	8. Unassigned points
389136	49.3	9. Not a conviction incident

Variable 11	INCIDENT DATE	MD1:	0	Field Width:	5
		MD2:	None	Type:	Numeric

FREQ Prcnt   INCIDENT DATE (JULIAN FORMAT)

00000. Unknown
- .
99999.

Variable 12	VEHICLE TYPE	MD1:	99	Field Width:	2
		MD2:	None	Type:	Numeric

FREQ	Prcnt	VEHICLE TYPE FROM CONVICTION RECORD
789008	100.0	99. Missing data

APPENDIX C

File Build Summary

INCIDENT FILE BUILD SUMMARY

The Driver Incident data set is produced by an HSRI build program from the Michigan Master Driver Record (MDR) data tape provided by the Department of State. The MDR tape includes records that document, among other things, license type, address, conviction, accident, and action information for every Michigan driver. To generate each of the variables in the incident file, the program looks at the appropriate positions in these MDR records and, depending on the variable, manipulates or recombines the values occurring there to produce the code values that appear in the file. The following is a variable-by-variable description of the process.

Variable 1: DRIVER NUMBER

The DRIVER NUMBER is assigned sequentially for each driver Header Record on the MDR Master Tape by the HSRI file build program.

Variable 2: INCIDENT NUMBER

The INCIDENT NUMBER is the sequential ordering, for each driver, of the driver's incidents. The incidents encountered on the MDR Master Tape are not necessarily found in chronological order, so before assigning incident numbers and writing them out for a particular driver, the program sorts the incidents into ascending order by INCIDENT DATE. This action, though, does not necessarily guarantee that the incidents are finally in chronological order as incidents with missing INCIDENT DATES will come first in the ordering.

Variable 3: LICENSE CODE

In deriving the LICENSE CODE, the build program first assigns the value of 99 to a variable called LICODE. On the MDR Master Tape's Header Record, the program then looks at "License Issued" (starting position 20) and, depending on its value, assigns to LICODE the value of:

10 for "0" (No License)
20 for "3" (Special Restricted)
21 for "4" (Minor Restricted)
30 for "1" (Operator)
35 for "5" (Operator with Cycle Endorsement)
40 for "2" (Chauffeur)
45 for "6" (Chauffeur with Cycle Endorsement)

For LICODE values between 30 and 45, the program then checks "Probation Code" (starting position 44) and, depending on its value, assigns to a variable LICADD the value of:

1 for "P" (Probationary License)

If this assignment can be made, it then checks "License Type" (starting position 22) and, depending on its value, assigns to LICADD the value of:

2 for "O" (Original)
3 for "R" (Renewal)
4 for "D" (Duplicate)
5 for "C" (Corrected)

Finally, the program adds LICADD to LICODE. The resulting value of LICODE is the LICENSE CODE value in the incident file.

Variable 4: ORIG LICENSE ISSUE DATE

The build program derives the ORIG LICENSE ISSUE DATE from the "Original License Date" (starting position 38) on the Header Record by converting the existing MMDDYY format into a Julian date. Missing dates on the Header Record are assigned a value of zero.

Variable 5: BIRTH DATE

The build program takes the BIRTH DATE from the MDR Header Record's "Birthdate" (starting position 53) where it has a format of MMDDYY. The program modifies it, though, by assigning the year to a variable called BYEAR, the month to

BMONTH, and the day to BDAY and, after multiplying BYEAR by 10000 and BMONTH by 100, adding them together. The resulting format of BIRTH DATE is YYMMDD. Missing dates are assigned a value of zero.

Variable 6: AGE ON INCIDENT DATE

The AGE ON INCIDENT DATE is calculated for each incident encountered, whether conviction, accident, or action. Depending on the incident, then, the program takes the "Arrest Date" (starting position 13) on the Conviction Record, the "Accident Date" (starting position 7) on the Accident Record, or the "Occurrence Date" (starting position 7) on the Action Record (or, if an untranslatable "Occurrence Date" is encountered, the "From Date" (starting position 22) on the Action Record) and, from the MMDDYY format that occurs in each instance, assigns values to an IMONTH, IDAY, and IYEAR variable. Then, using the variables generated in the derivation of BIRTH DATE (BMONTH, BDAY, and BYEAR), the program performs the following calculations:

```
C
C     Accomodate drivers born before turn of century.
C
      IF (IYEAR .LT. BYEAR) IYEAR = IYEAR + 100
C
C     Is incident date beyond birth date within year?
C
      IF (IMONTH .GT. BMONTH) GOTO 50
      IF (IMONTH .EQ. BMONTH .AND. IDAY .GE. BDAY) GOTO 100
C
C     If not, adjust incident year and month accordingly.
C
      IYEAR  = IYEAR - 1
      IMONTH = IMONTH + 12
C
C     Adjust month if not beyond birth day within month.
C
   50 IF (IDAY .LT. BDAY) IMONTH = IMONTH - 1
C
C     Calculate age from incident date and birth date.
C
  100 AGE = (IYEAR - BYEAR)*100 + (IMONTH - BMONTH)
C
```

The resulting value of AGE (with a format of YYMM) is the AGE ON INCIDENT DATE in the incident file.

Variable 7: COUNTY OF RESIDENCE

The build program determines the COUNTY OF RESIDENCE by taking the "Zip Code" (starting position 161) on the MDR Header Record and, using an array of the 1175 Michigan ZIP Codes, translates it into a two-digit value that represents one of Michigan's 83 counties. If the Zip Code is missing on the Header Record or does not subsequently correspond to a Michigan county, the value in the incident file will be 99.

Variable 8: SEX

The build program takes SEX directly from the MDR Header Record's "Sex" (starting position 52), assigning it a value of:

1 for "M" (Male)
2 for "F" (Female)
9 for " " (Missing Data)

Variable 9: INCIDENT CODE

The INCIDENT CODE is arrived at in one of three ways, depending on whether the incident is a conviction on a traffic offense, an accident involvement, or a driver improvement action. Each driver incident encountered becomes, of course, a distinct record in the incident file. The codes for the three different types of incidents are derived as follows:

- Conviction INCIDENT CODE

When a conviction occurrence is encountered on the MDR Conviction Record, the program first determines if it occurred before a cut-off date (prompted for as the program begins and entered by the user) and then computes its incident number. The program then calculates the age of the driver at the time of the conviction (see Variable 6) and the date (see Variable 11).

The program begins processing the conviction incident by translating the characters that define "Offense Code" (starting position 50) into a binary number. The program then uses this value to properly index a value position in an array OFFPTS that becomes the POINTS associated with this offense. OFFPTS is a "two by ninety-nine" array that holds valid offense code point values as defined in the document "Offense Code Breakdown" provided by the Department of State Data Processing Center and dated April 1, 1981. Besides the point values, OFFPTS also contains a flag for each offense code that indicates whether or not point adjustment is necessary. OFFPTS(X,1), for example, holds the point value (as of 4/1/81) of offense code X, while OFFPTS(X,2) contains the flag that indicates, when not zero, that the current point value is so much more (or less) than what it was before a specified cutoff date. Offense code positions that have no associated points or that are undefined have point values of eight, as do offense codes 18 and 19 (which, if not subsequently replaced, will indicate undetermined points).

The point values for offense codes 18 ("regular" speed convictions) and 19 ("energy" speed convictions) are determined next. The first three positions of "Speed" (starting position 52) are translated into the variable ACTSPD (actual speed), while the last two become the variable POSSPD (posted speed). (For regular speed offenses, blanks occurring in these positions are first converted into zeros.) For energy speed offenses, if ACTSPD is less than 60, POINTS becomes zero. If ACTSPD is between 60 and POSSPD, POINTS becomes zero if the incident occurred before 4/1/81, and one if on that date or after. Otherwise, for both regular and energy speed offenses, the following assignments are made:

```
XTRSPD = ACTSPD - POSSPD
IF (XTRSPD .LE. 10) POINTS = 2
IF (XTRSPD .GT. 10) POINTS = 3
IF (XTRSPD .GT. 15) POINTS = 4
```

The point-readjustment flag is then checked and, for those offenses that have had differing point values in the past, a readjustment is made if the incident occurred before a specified date (at this time, the only date considered is 8/1/79).

Variable 9 is finally assembled by taking the translated offense code (what was literally found on the conviction record) and placing the value of POINTS in front of it. The resulting value represents the conviction INCIDENT CODE.

Variable 11 is also generated at this time by simply assigning it the value of POINTS. If, however, the "Same Incident/Late Recd/Bond Forfeiture" flag in position 57 of the Conviction Record is anything but blank or two, POINTS FROM INCIDENT is set equal to zero.

Finally, the program takes whatever value is found for "Type of Vehicle" (starting position 58) on the convicition record and assigns it to the output incident positions that correspond to variable 12, VEHICLE TYPE. If "Type of Vehicle" is blank, the assigned value is 99. The value will also be 99 for non-conviction incidents.

- Accident INCIDENT CODE

When an accident case is encountered on the MDR Accident Record, the program first determines if it occurred before the specified cut-off date and then computes its incident number. The program then calculates, as it did with convictions, the age of the driver at the time of the accident (see Variable 6) and the date (see Variable 11).

With accidents, in deriving the INCIDENT CODE and

POINTS FROM INCIDENT, the program first assigns a value of 19979 to the five incident file positions that represent these two variables. It should be noted that, with all accident occurrences, the POINTS FROM INCIDENT will be 9.

The program then examines the six positions of "Counts (Vehicles, Injured, Killed)" (starting position 13) on the Accident Record. First, by looking at the first two positions that indicate vehicle count, the program assigns a value of one to a variable SEVER if the count is one, or assigns it a value of four if the count is greater than one. Then, by looking at the next two positions that indicate injury count, the program assigns a value of one to a variable DAMTYP if the count is greater than zero. And thirdly, by looking at the last two positions that indicate fatal count, the program assigns a value of two to DAMTYP if the count is greater than zero. Finally, the program adds DAMTYP to SEVER to arrive at a new value for SEVER. If none of the above conditions have been met, however, SEVER is assigned a value of nine. As a result, the values for SEVER represent as follows:

1 - a single vehicle, property damage accident
2 - a single vehicle, injury accident
3 - a single vehicle, fatal accident
4 - a multiple vehicle, property damage accident
5 - a multiple vehicle, injury accident
6 - a multiple vehicle, fatal accident
9 - an unknown vehicle-count or type of accident

After this, the program takes "Coded Items" (starting position 54) on the Accident Record and scans through its eight positions for any occurrence of "X3", "X4", or "X5". These items describe the role that drinking had in the accident. Depending on the value found, the program assigns to a variable HBD the value:

8 for "X3" (Had Been Drinking)
7 for "X4" (Had Not Been Drinking)
6 for "X5" (Not Stated)

The program then scans "Coded Items" once more for violation (or "V") codes that indicate, for the accident, any suspected driver-fault that contributed to the accident, whether or not an actual conviction was obtained. If any V's are found, a variable VIOL is set to zero. Because the meanings of the violation codes changed in 1978, the program then checks the occurrence year (position 11) to see whether or not the incident occurred before 1978. The meanings of the pre-1978 violation codes are as follows:

V1 - Speed Too Fast
V2 - Failed to Yield Right of Way
V3 - Drove Left of Center
V4 - Improper Overtaking
V5 - Passed Stop Sign
V6 - Disregarded Traffic Signal
V7 - Followed Too Closely
V8 - Make Improper Turn
V9 - Improper or No Signal
V11 - Other Improper Driving

The meanings of the current violation values are as follows:

V1 - Speed Too Fast
V2 - Speed Too Slow
V3 - Failed to Yield Right-of-Way, Disregard of Traffic Control
V4 - Drove Wrong Way
V5 - Drove Left of Center, Improper Overtaking and Passing, Improper Lane Usage
V6 - Improper Turn, Improper or No Signal
V7 - Improper Backing, Unsafe Start
V8 - Following Too Closely, Unable to Stop Within Assurred Clear Distance Ahead, Failed to Use Due Care and Caution

If no incident date is found, the value of VIOL will remain zero. If the incident occurred before 1978, the program performs the following translations:

V1 remains V1
V2 becomes V3
V3 becomes V5
V4 becomes V5
V5 becomes V3
V6 becomes V3
V7 becomes V8
V8 becomes V6
V9 becomes V6

V11 becomes V9

The program then looks through "Coded Items" in an ordered fashion and translates the first V code found by rank directly into the variable VIOL (e.g., V1 causes VIOL to have the value of one). The ranking, which determines the translation priority if more than one V code is found, is as follows: V4, V5, V3, V8, V1, V6, V2, V7, V9.

For accidents in which a V code was found, the program inserts the value of HBD into the second position of the INCIDENT CODE in the incident file, the value of SEVER into the third position, and the value of VIOL into the fourth. For accidents in which no driver violation was found, the program inserts five into the INCIDENT CODE's second position, the value of HBD into the third position, and the value of SEVER into the fourth.

By means of a later subroutine that writes out all of a driver's accumulated incidents, the first position of the INCIDENT CODE is assigned the value "2" if it is determined that at least one conviction occurred on the same day as the accident. If no convictions are associated with the accident, the value of the first position remains "1".

The resulting values in the incident file represent the final accident INCIDENT CODE, as well as POINTS FROM INCIDENT (with a value of nine).

- Action INCIDENT CODE

When a driver improvement action is encountered on the MDR Action Record, the program first determines, as it did with convictions and accidents, if it occurred before the specified cut-off date and, if so, computes the incident number. Here, too, the program also calculates the age of the driver at the time of the action (see Variable 6) and the date (see Variable 11).

Initially, the program sets the first position of the

INCIDENT CODE in the incident file to "3" and the position representing POINTS FROM INCIDENT to "9". The program then assigns to a variable ACTINC the value "998". ACTINC will correspond to the last three positions of the INCIDENT CODE. In this case, if not subsequently replaced, INCIDENT CODE "3998" will represent "Other Action Record".

The program then examines the four-position "Action-Type Code" (starting position 13) on the Action Record. It first takes the value in the first two positions (indicating the action) and assigns it to a variable called ACTCD. It then takes the value in the last two positions (indicating the type) and assigns it to a variable TYPE. Finally, the program performs a series of logical checks of ACTCD and TYPE in combination with the presence of certain values of "Reason Codes" (starting position 34), "Occurrence Date" (starting position 7), "From Date" (starting position 22), and "Thru Date" (starting position 28) on the Action Record. Depending on the results of these checks, ACTINC is assigned a value of:

```
"901"  if ACTCD = 01;  TYPE = 11.
"902"  if ACTCD = 11;  TYPE = 11;
          Reason Code = "D".
"903"  if ACTCD = 11;  TYPE = 11;
          Reason Code = "E" or is blank.
"904"  if ACTCD = 11;  TYPE = 11;
          Reason Code = "H".
"905"  if ACTCD = 30, 31 or 36;  TYPE = 11;
          Reason Code ≠ "X", "Y", "B", "C", 16,
                        46-59, 78, 95 or 97;
          Occurrence Date > 000000.
"906"  if ACTCD = 30;  TYPE = 11;
          Reason Code = "X".
"907"  if ACTCD = 30;  TYPE = 11;
          Reason Code ≠ "X";
          Occurrence Date = 000000;
          From Date > 000000.
"908"  if ACTCD = 30;  TYPE = 21.
"909"  if ACTCD = 40;  TYPE = 11;
          Reason Code ≠ "X", "Y", "B", "C", 16,
                        46-59, 78, 95 or 97;
          Occurrence Date > 000000.
"910"  if ACTCD = 40;  TYPE = 11;
          Reason Code = "X" or "Y".
```

```
"911"  if ACTCD = 40;  TYPE = 11;
          Reason Code = "O", "P" or is blank;
          Occurrence Date = 000000;
          From Date > 000000.
"912"  if ACTCD = 60;  TYPE = 11;
          Reason Code ≠ 13;
          Occurrence Date > 000000;
          From Date > 000000;
          Thru Date > 000000.
"913"  if ACTCD = 60;  TYPE = 11;
          Reason Code ≠ 13;
          Occurrence Date = 000000;
          From Date > 000000.
"914"  if ACTCD = 03;  TYPE = 11;
          Reason Code ≠ "X" (unless it is "X20"),
                        40-42, 44 or "Y".
"915"  if ACTCD = 03;  TYPE = 11;
          Reason Code = "X".
"916"  if ACTCD = 07;  TYPE = 51.
"917"  if ACTCD = 40;  TYPE = 51;
          Occurrence Date > 000000.
       or ACTCD = 60;  TYPE = 11;
          Reason Code = 13.
```

The resulting values in the incident file represent the final action INCIDENT CODE, as well as POINTS FROM INCIDENT (with a value of "9").

Variable 10: POINTS FROM INCIDENT

The build program determines the POINTS FROM INCIDENT while deriving the value for the conviction INCIDENT CODE (Variable 9).

Variable 11: INCIDENT DATE

The INCIDENT DATE is calculated for each incident encountered, whether conviction, accident, or action. Depending on the incident, then, the program takes the "Arrest Date" (starting position 13) on the Conviction Record, the "Accident Date" (starting position 7) on the Accident Record, or the "Occurrence Date" (starting position 7) on the Action Record (or, if an untranslatable "Occurrence Date" is encountered, the "From Date" (starting position 22) on the Action Record) and converts the existing

MMDDYY format into a Julian date. Missing dates are assigned a value of zero.

Variable 12: VEHICLE TYPE

The build program determines VEHICLE TYPE while deriving the value for the conviction INCIDENT CODE (Variable 9).

APPENDIX D

Master Driver Tape Layout

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

" → " used by the build program.

TAPE FILE LAYOUT

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
FILE ID: MDRMAST
MULTI-FILE ID:
CHAR/REC: 48-210
FIXED - REC/BLOCK:
VARIABLE - CHAR/BLOCK: 7791-8000

DATE 11/1/81

X 9 TRACK
__ 7 TRACK

MODE		DENSITY - BPI		PARITY	
X EBCDIC	BCL	1600	X 6250	ODD	
EBCDIC	BCL	556	800	ODD	EVEN

RECORD NAME: Header Record
SIZE: 210

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N	CLASS A	CODED FIELD	FORMAT
Record Count & Identification	1	5	X				
Indicator	6	1	X			X	
Driver License Number	7	13	X				
→ License Issued (used to be called ...)	20	1		X		X	
License Endorsement	21	1		X		X	
→ License Type (used to be called ...)	22	1			X	X	
License Extension	23	1			X	X	
License Duration	24	1		X			
License Restriction	25	1	X			X	
Issue Date	26	6		X			MMDDYY
Exam Station (Branch Number)	32	4	X			X	
→ Expiration Year	36	2		X			YY
→ Original License Date	38	6		X			MMDDYY
→ Probation Code	44	1			X	X	
End of Probation Date	45	6		X			MMDDYY
Probation Pre-Notice Code	51	1		X		X	
→ Sex	52	1			X	X	
→ Birthdate	53	6		X			MMDDYY
(Continued)							

This document paid for with State funds

EBCDIC: 8 BIT CHAR, 6 CHAR/WORD
BCL: 6 BIT CHAR, 8 CHAR/WORD

CODED FIELD: X - STANDARD CODE
Y - SPECIAL CODE (attac

DP-41

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Header Record (Continued)

DATE 11/1/81
SIZE:

FIELD NAME	START POS	LENGTH	CLASS AN	N	A	CODED FIELD	FORMAT	C VA
Name (First, Middle, Last)	59	36	X					
Name Char Count	95	2		X				
Street	97	36	X					
Street Char Count	133	2		X				
City	135	19	X					
City Code (If present)	154	3		X		X		
City Char Count (If no city code)	157	2		X				
State	159	2		X		X		
Zip Code	161	5		X				
County	166	2		X		X		
Microfilm #1	168	5	X					
Microfilm #2 (Backup)	173	5	X					
School Number	178	3	X			X		
Print Batches	181	4	X					
Posting Date (Julian)	185	5		X			YYDDD	
Last Conviction Date (Julian)	190	5		X			YYDDD	
Address Change Date (Julian)	195	5		X			YYDDD	
Last Activity Code	200	1	X			X		
Registration Notice Sent	201	1		X		X		
Renewal Select Code (Print Tape)	202	1			X	X		
Out-of-state State Code	203	2			X	X		
Filler	205	6			X			Bl

CODED FIELD: X = STANDARD CODE
Y = SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Dummy Header

PAGE 3 of 11
DATE 11/1/81
SIZE: 210

FIELD NAME	START POS	LENGTH	CLASS AN	N	A	CODED FIELD	FORMAT	CO[...] VAL[...]
Record Count & Identification	1	5	X					021
Indicator	6	1	X					
Old Driver License Number	7	13	X					
Filler	20	32			X			Bla
Sex	52	1			X			Jor
Old Birthdate	53	6		X			MMDDYY	
Old Name (First, Middle, Last)	59	36	X					
Old Name Char Count	95	2		X				
New Name (First, Middle, Last)	97	36	X					
New Name Char Count	133	2		X				
New Driver License Number	135	13	X					
New Birthdate	148	6		X			MMDDYY	
Filler	154	14			X			Bla
Microfilm #	168	5	X					
Filler	173	8			X			Bla
Print Batches	181	4	X					
Posting Date (Julian)	185	5		X			YYDDD	
Filler	190	21			X			Bla

CODED FIELD: X - STANDARD CODE
Y - SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

T A P E F I L E L A Y O U T (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Address History

PAGE 4 of 11
DATE 11/1/81
SIZE: 48

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N	CLASS A	CODED FIELD	FORMAT	CC VAL
Record Count & Identification	1	5	X					004
Indicator	6	1	X					
Change Date 1	7	6		X			MMDDYY	
Microfilm #1	13	5	X					
Change Date 2	18	6		X			MMDDYY	
Microfilm #2	24	5	X					
Change Date 3	29	6		X			MMDDYY	
Microfilm #3	35	5	X					
Posting Date (Julian)	40	5		X			YYDDD	
Filler	45	4			X			Bla

CODED FIELD: X = STANDARD CODE
Y = SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Previous Name or Alias

PAGE 5 of 11
DATE 11/1/81
SIZE: 78

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N	CLASS A	CODED FIELD	FORMAT	CON VALU
Record Count & Identification	1	5	X					0078I
Indicator	6	1	X					
Previous Name or Alias (Fir, Mid, Last)	7	36	X					
Name Char Count	43	2		X				
Old Drivers License Number	45	13	X					
Old Birthdate	58	6		X			MMDDYY	
Microfilm #	64	5	X					
Posting Date (Julian)	69	5		X			YYDDD	
Filler	74	5			X			Blank

CODED FIELD: X - STANDARD CODE
Y - SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

T A P E   F I L E   L A Y O U T (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Special Restriction

DATE 11/1/81
SIZE: 54

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N	CLASS A	CODED FIELD	FORMAT	CO VAL
Record Count & Identification	1	5	X					005
Indicator	6	1	X					
Restriction Codes (6 2-posit. codes)	7	12	X			X		
Restriction Description	19	28	X					
Description Char Count	47	2		X				
Posting Date (Julian)	49	5		X			YYDDD	
Filler	54	1			X			Blar

CODED FIELD: X = STANDARD CODE
Y = SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

X

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Conviction

PAGE 7 of 11
DATE 11/1/81
SIZE: 66

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N	CLASS A	CODED FIELD	FORMAT
Record Count & Identification	1	5	X				
Indicator (Purge)	6	1	X				
→ Conviction Date	7	6		X			MMDDYY
→ Arrest Date	13	6		X			MMDDYY
Microfilm #	19	5	X				
Court	24	20	X				
Court Code (If present)	44	3	X			X	
Court Char Count (If no court code)	47	2		X			
← Type of Court Code	49	1			X	X	
→ Offense Code	50	2		X		X	
→ Speed (going for 3/limit for 2)	52	5	X				
→ Same Incident/Late Recd/Bond Forfeiture	57	1	X			X	
→ Type of Vehicle	58	2	X			X	
Posting Date (Julian)	60	5		X			YYDDD
Print Tape New Indicator	65	1		X			
Filler	66	1			X		B

CODED FIELD: X - STANDARD CODE
Y - SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: FAC/FCJ/FCPV

DATE 11/1/81
SIZE 84

FIELD NAME	START POS	LENGTH	CLASS AN	N	A	CODED FIELD	FORMAT	VA
Record Count & Identification	1	5	X					OC
Indicator	6	1	X					
Suspension Date	7	6		X			MMDDYY	
Arrest Date	13	6		X			MMDDYY	
Court Date or Court File Number *	19	7	X					
Suspension Microfilm #	26	5	X					
Termination Microfilm #	31	5	X					
Court	36	20	X					
Court Code (If present)	56	3	X			X		
Court Char Count (If no court code)	59	2		X				
*Type of Court Code	61	1			X	X		
Offense Code	62	2		X		X		
Speed (Going for 3/Limit for 2)	64	5	X					
Termination Date	69	6		X			MMDDYY	
FAC or FCJ Code	75	1		X		X		
Posting Date (Julian)	76	5		X			YYDDD	
Filler	81	4			X			Bl

* Court Date will have "*" in position 1, followed by MMDDYY;
if no "*", field will be 7 position court file number.

CODED FIELD: X = STANDARD CODE
Y = SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: L. Pixley
RECORD NAME: Accident

PAGE 9 of 11
DATE 11/1/81
SIZE: 66

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N.A	CODED FIELD	FORMAT	CO: VAL
Record Count & Identification	1	5	X				0066
Indicator	6	1	X				
→ Accident Date	7	6		X		MMDDYY	
→ Counts (Vehicles, Injured, Killed)	13	6		X			
Report Number	19	6		X			
Police Department	25	24	X				
Department Code (If present)	49	3	X		X		
Department Char Count (If no dept code)	52	2		X			
→ Coded Items	54	8	X		X		
Posting Date (Julian)	62	5		X		YYDDD	

CODED FIELD: X = STANDARD CODE
Y = SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Action

PAGE 10 of 11
DATE 11/1/81
SIZE: 72

FIELD NAME	START POS	LENGTH	CLASS AN	CLASS N	CLASS A	CODED FIELD	FORMAT	CON VALU
Record Count & Identification	1	5	X					0072
Indicator (Purge)	6	1	X					
Occurrance Date (A-date)	7	6		X			MMDDYY	
Action-Type Code	13	4		X		X		
Microfilm #	17	5	X					
From Date (B-date)	22	6		X			MMDDYY	
Thru Date (C-date)	28	6		X			MMDDYY	
Reason Codes	34	8	X			X		
E-field (FR Case, App #, Crt Cd, or Misc*)	42	12	X			X		
Lifted Date (D-date)	54	6		X			MMDDYY	
Original Action Date (F-date)	60	6		X			MMDDYY	
Posting Date (Julian)	66	5		X			YYDDD	
Print Tape New Indicator	71	1		X				
Filler	72	1			X			Blank

* On Referrals and Warning Letters:
pos. 1-3 contain class code if 12+ points; 11-12 contain points at time of action.

On DI re-exams:
pos. 1-2 contain county; 3-6 contain analyst; 7-8 contain alcohol referral; 9 contains passed or failed road test; 10 contains points considered.

On DLAD & CC hearings:
pos. 1-4 contain county & hearing officer or judge, 5-6 contain previous action typ 7-8 contain previous action reason; 9-10 contain second previous action reason

CODED FIELD: X - STANDARD CODE
Y - SPECIAL CODE (attached)

DP-41A
Revised 3/79

MICHIGAN DEPARTMENT OF STATE
DATA PROCESSING CENTER

TAPE FILE LAYOUT (continued)

LAYOUT NUMBER: DR120T
FILE NAME: MDR Master
PREPARED BY: J. Pixley
RECORD NAME: Action Description

DATE 11/1/81

SIZE: 48

FIELD NAME	START POS	LENGTH	CLASS AN	N	A	CODED FIELD	FORMAT	CO VA
Record Count & Identification	1	5	X					004
Indicator	6	1	X					
Action Description *	7	32	X					
Description Char Count	39	2		X				
Count (# of Action Desc Records)	41	2		X				
Posting Date (Julian)	43	5		X			YYDDD	
Filler	48	1			X			Bla

* Description Record 02 may contain up to 16 two-position numeric reason codes.

CODED FIELD: X = STANDARD CODE
Y = SPECIAL CODE (attached)

DP-41A
Revised 3/79