2005

Development of an alternative approach to estimate vehicle miles traveled using a classification procedure

Srividya Vadlamani
Louisiana State University and Agricultural and Mechanical College, svadla1@lsu.edu

Follow this and additional works at: https://digitalcommons.lsu.edu/gradschool_theses

Part of the Civil and Environmental Engineering Commons

Recommended Citation
Vadlamani, Srividya, "Development of an alternative approach to estimate vehicle miles traveled using a classification procedure" (2005). LSU Master's Theses. 2511.
https://digitalcommons.lsu.edu/gradschool_theses/2511

This Thesis is brought to you for free and open access by the Graduate School at LSU Digital Commons. It has been accepted for inclusion in LSU Master's Theses by an authorized graduate school editor of LSU Digital Commons. For more information, please contact gradetd@lsu.edu.
DEVELOPMENT OF AN ALTERNATIVE APPROACH TO ESTIMATE VEHICLE MILES TRAVELED USING A CLASSIFICATION PROCEDURE

A Thesis

Submitted to the Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering

in

The Department of Civil and Environmental Engineering

by

Srividya Vadlamani
B.E., Andhra University, 2002
May 2005
Dedicated to my dearest grandparents
Acknowledgements

I would like to thank my advisor Dr. Chester Wilmot for making this work possible. No words of appreciation are enough for his guidance, support, encouragement and patience. I really need to acknowledge the freedom he gave me in pursuing my research and for making me work on various aspects of transportation planning. His experience and observations helped me a lot in defining my research path. I have learned many things from him in the last few years both as a researcher and as a person.

I would like to thank Dr. John Metcalf and Dr. Sherif Ishak, for taking their time and agreeing to be a part of my committee. I would also like to thank Dr. Sherif Ishak and Dr. Ling Bing Wang for their encouragement and support. I would like to thank Mr. Thomas Roff of FHWA for supplying the data. I would like to thank Mr. Erik Williams of Tele Atlas, Dr. Xionping Zhang of Coastal Sciences, and Mr. Glenn Chustz and Mr. Neil Hohmann of LADOTD for their timely help and support.

I would like to thank my grandfather and best friend without whom none of this would have been possible. I would like to thank my dearest parents for everything, above all for making me believe in my dreams and for always being my inspiration to achieve them. Deeps thanks for all your love and support. Sheela, Swapna, and Karthik thanks a lot for everything and thanks for being there with me throughout.

Finally, I would like thank all my friends and colleagues at LSU, one and all for their encouragement and support, and for making my stay here, the Most Wonderful.
Table of Contents

Acknowledgements...........................................................................................................ii
List of Tables...................................................................................................................vii
List of Figures....................................................................................................................viii
Abstract............................................................................................................................ix

1. Introduction..................................................................................................................1
  1.1 Background..............................................................................................................1
  1.2 Problem Statement.................................................................................................2

2. Literature Review.......................................................................................................4
  2.1 Introduction ............................................................................................................4
  2.2 VMT Input Requirements for MOBILE6..............................................................4
    2.2.1 VMT by Vehicle Class.....................................................................................4
    2.2.2 VMT by Facility .............................................................................................4
    2.2.3 VMT by Hour .................................................................................................6
    2.2.4 VMT by Average Speed..................................................................................6
  2.3 Review of Current VMT Estimation Methods.......................................................6
    2.3.1 Non-Traffic Count Based Methods...............................................................9
    2.3.2 Traffic-Count Based Methods ....................................................................10
  2.4 Summary ................................................................................................................12

3. Objectives of the Study.............................................................................................14

4. HPMS Data...............................................................................................................15
  4.1 Obtaining the HPMS Traffic Count Data............................................................15
    4.1.1 General ........................................................................................................15
    4.1.2 Volume Group Assignments .......................................................................16
    4.1.3 Traffic Monitoring Guide ..........................................................................16
      4.1.3.1 Continuous ATRs .............................................................................17
      4.1.3.2 Short Counts ......................................................................................17
      4.1.3.3 Vehicle Classification Data Collection .............................................18
    4.1.4 Application of Traffic Count Procedures to HPMS .......................................18
  4.2 HPMS VMT Calculation Process .........................................................................20
  4.3 HPMS Data Items ................................................................................................21
    4.3.1 HPMS Data ................................................................................................21
      4.3.1.1 Universe Data ...................................................................................21
      4.3.1.2 Sample Data .....................................................................................22
      4.3.1.3 Donut Sample Data ..........................................................................22
      4.3.1.4 Summary Data ..................................................................................23
    4.3.2 HPMS Data Items .......................................................................................23
      4.3.2.1 Annual Average Daily Traffic (AADT) ............................................23
      4.3.2.2 Rural/ Urban Designation .................................................................23
List of Tables

Table 2(a) Complete MOBILE6 Vehicle Classification ........................................... 5
Table 2(b) Composite MOBILE6 Vehicle Classification ........................................... 6
Table 4-1 Rural/Urban Designation Codes in HPMS .............................................. 24
Table 4-2 FHWA Approved HPMS Functional Classification and Codes ............... 25
Table 4-3 NHS Codes for HPMS Data ................................................................. 26
Table 4-4 Type of Facility Codes and Description for HPMS data ....................... 27
Table 5-1 HPMS Vehicle Codes and Description .................................................. 32
Table 5-2 Vehicle Type Conversion from HPMS to MOBILE6 (%) ....................... 34
Table 6-1 HPMS and MOBILE6 Functional Classification ................................... 47
List of Figures

Figure 5-1 Tree Map Showing the Overview of the Classification Tree……………….30
Figure 5.2: Classification Tree in Answer Tree…………………………………………30
Figure 5.3: Schematic Process to Identify and Assign VMT…………………………37
Figure 6-1 Adding Route Events for HPMS Data from the NHPN Data……………….42
Figure 6-2 Route Event Theme Displaying the Louisiana HPMS Road Network……….43
Figure 6-3 Adding the Control Sections layer to the View……………………………44
Figure 6-4 Adding LA Highway Network Route Events to the State Roads Network….45
Figure 6-5 Map Showing the Added LA Highway Route Events………………………45
Figure 7-1 Created EBR Highway Network…………………………………………….54
Figure 7-2 Dataview Showing the Observed and Estimated AADT……………………55
Figure 7-3 T-Test Result Output for the Observed AADT and CART Estimated
AADT……………………………………………………………………………..56
Figure 7-4 T-Test Result Output for the Observed AADT and HPMS Method Estimated
AADT………………………………………………………………………………57
Figure 7-5 T-Test Result Output for the AADT Estimates Obtained Through HPMS
Method and CART Classification Method…………………………………………58
Figure 7-6 Average Percentage Difference between Observed and Estimated AADT….59
Figure 8-1 CART Classification System Add-in Generated DVMT Estimate………..61
Figure 8-2 The Info Toolbox created by CART Classification Add-in…………………..62
Figure 8-3 CART Classification system add-in created VMT Distribution………………63
Figure 8-4 Selecting VMT Distribution Options……………………………………….63
Figure 8-5 Choosing VMT Input Options………………………………………………64
Figure 8-6 Selecting Vehicle Type………………………………………………………65
Figure 8-7 Selecting System VMT………………………………………………………66
Abstract

In this study a Classification and Regression Tree (CART) classification system method was developed as an alternative to the HPMS conventional method of VMT estimation. Louisiana Highway Performance Monitoring System (HPMS) data from Federal Highway Administration (FHWA) and Louisiana highway network data from Louisiana Department of Transportation (LADOTD) were used to carry out the analysis. The CART classification was derived based on link attributes that effectively distinguished the homogeneous traffic volume sections. VMT estimation on individual links as well for the different systems was calculated using the CART classification system and the values were assigned on each link. The default MOBILE6 VMT values were also assigned on each link. The entire procedure was carried out using GIS Developer’s kit in TransCAD. The whole procedure was automated by creating a customized add-in program to TransCAD.

The resulted AADT estimates from the CART classification system equation were compared with the HPMS conventional method estimates and with the observed values on the HPMS sample sections. This comparison was done graphically and statistically by the paired sample T-test. The results indicated that the HPMS conventional method performs better in estimating AADT than the CART classification system method.

The study also included the demonstration of the CART classification system add-in in the East Baton Rouge non-attainment area. The developed add-in can also be used on any other traffic count data for any region, state or country. The GIS program developed during this study provides a framework, which can be built upon further and shared with other researchers in future.
1. Introduction

1.1 Background

Vehicle miles traveled (VMT) is a unit to measure the travel made by a private vehicle, such as an automobile, van, pickup truck, or motorcycle. True VMT is the total number of miles traveled by a given vehicle or fleet of vehicles in a given period of time (e.g., day, year) on the road network (Qiao and Yu, 2004).

Estimates of VMT are used extensively in transportation-related areas such as transportation planning, vehicle emissions estimation, energy consumption computation, traffic impact analysis, resources allocation, etc (Kumapley, 1994). For instance, VMT, combined with pollution rates per mile traveled, provide an estimate of the total amount of vehicle pollution in a given period of time. VMT estimates are also important primary inputs to the emission estimation model MOBILE, designed by the Environment Protection Agency (EPA) to estimate current and future vehicle emissions under different conditions. This model has been approved by the EPA, for all states in United States except California for the conformity determination analysis.

MOBILE6, the latest in a series of MOBILE models, requires detailed VMT estimates, which include VMT BY VEHICLE CLASS, VMT BY FACILITY, VMT BY HOUR, and SPEED VMT. VMT BY VEHICLE CLASS is a set of VMT fractions by each of 16 composite vehicle types; VMT BY FACILITY is the VMT distribution by facility type for each of the 96 scenarios (with 4 road types including freeway, arterial, local and ramp, and 24 hours from 6 AM to 5 AM); VMT BY HOUR is the hourly distribution of VMT for each of the 24 hours and all facility types (with the sum of the total 24 values adding up to 1); VMT BY SPEED is the VMT distribution across 14 pre-selected speed ranges for each of the 24 hours of the day for freeways and arterials. MOBILE6 includes national default values of VMT related variables, provided by the EPA. If input data is unavailable, users may use the national default values. However, EPA recommends the use of locality-specific VMT values in estimating emissions whenever possible.

Traditionally, the vehicle activity data are supplied by travel demand models, traffic count data, personal travel surveys or instrumented vehicle studies. Since, these methods are not developed for air quality modeling, the data sources are seldom accurate
enough for the detailed level emission modeling required in MOBILE6. Hence, developing accurate travel related input data for air quality models like MOBILE6 is a major challenge for the transportation planners involved.

EPA suggests the use of traffic ground counts or network-based transportation models in the estimation of vehicle miles traveled. EPA's guidance is based on the need to have a reliable VMT estimation methodology, applicable nationwide, that lends itself readily to annual use. The Highway Performance Monitoring System (HPMS) was chosen for its ability to fill that role. Network-based travel demand models, with their time-consuming and resource-intensive updating procedures, are not a practical alternative to HPMS as an annual VMT estimation method (Workshop on Transportation Air Quality Analysis, 1994).

The current research focuses on developing an alternate methodology to the conventional method of estimating VMT using the HPMS traffic count data and to apply that method to the non-attainment parish of East Baton Rouge in a practical demonstration of the process.

1.2 Problem Statement

VMT can be estimated from other traffic and/ or socioeconomic variables. The various methods, which have been developed, result in different estimates of VMT for the same road network. Furthermore, with the strict environmental monitoring and good transportation planning required under the 1990 Clean Air Act Amendments (CAAA) and the 1991 Intermodal Surface Transportation Efficiency Act (ISTEA), the accuracy, stability, and growth rates of VMT estimates are of utmost concern.

The EPA is required by Section 187 of the Clean Air Act Amendments of 1990 (CAAA), to provide guidance on the measurement of VMT. They recommend the use of systematic traffic ground counts, at least in the urbanized areas. This method is considered by EPA to be superior in terms of both practicality and effectiveness to other methods such as driver surveys, odometer data, registration counts, fuel sales, annually validated network models, etc.

In preparing this guidance, EPA encountered two views among interested parties regarding how ground counts should be used to estimate actual travel in a year just
completed. One view supported the approach that has already been institutionalized in the Federal Highway Administration's Highway Performance Monitoring System. In this approach, traffic counts taken at various points on an urban area's road network are directly expanded into an estimate of area-wide VMT using statistics on the number of roadway miles associated with each sampling point. The other view supported the use of network-based transportation models, which theoretically provide more detail on the location, sources, and purposes of travel. While network-based models begin with only indirect indications of VMT (for example, number of households, household locations, and household trips per day by purpose), their final results are generally validated via comparison to actual ground counts at selected sites, usually sites on major traffic corridors. However, annual updates of household and other input data and annual validation against traffic counts would be too resource-intensive to be practical. Therefore, EPA has chosen to specify the use of the HPMS approach in this guidance for purposes of tracking 1993 and later VMT (Section 187, VMT Forecasting and Tracking Guidance, 1990).

The HPMS method of VMT estimation involves the use of adjusted 24-hr traffic counts, referred to as annual average daily traffic (AADT); obtained on sample sections identified through a systematic stratified random sampling process. The sample section VMT is estimated as the product of the section AADT and road segment mileage. The sample section VMT is expanded using expansion factors to obtain the areawide and universal VMT estimates. The expansion factors are generated by volume group within each functional system and geographical area classification. However, no consideration is given to their specific link attributes such as lane width, shoulder width, speed limit or surrounding land use. The research performed in the present study is aimed at determining whether expanding the classification process to include such attributes, will provides better VMT estimates from HPMS data. However, HPMS information on traffic counts is unavailable for some functional classes, such as local roads in the state network i.e., the traffic data are not representative of the total system of roads. Despite this, HPMS data was selected as this study mainly focuses on developing a methodology that can provide better estimates of VMT than the HPMS conventional method.
2. Literature Review

2.1 Introduction

MOBILE6 is the newly released version of the MOBILE models, first produced in 1978. The EPA’s MOBILE6 User’s Guide and its various other technical documents are the primary source of literature for the users of the MOBILE6 model. Prior to reviewing the published literature, EPA’s Users Guide to MOBILE6 was reviewed with specific emphasis on detailed VMT input requirements for MOBILE6.

2.2 VMT Input Requirements for MOBILE6

The MOBILE6 User’s Guide gives a detailed explanation for each input requirement. The MOBILE model requires several traffic related inputs for different vehicle types. The EPA divides the total number of vehicles into 28 vehicle types based on their weight, type and fuel type. This classification is known as the complete MOBILE6 vehicle classification shown in Table 2(a). Another classification in which gasoline and diesel vehicles are combined has 16 vehicle types and is known as the MOBILE6 composite vehicle classification as shown in Table 2(b).

The MOBILE model requires several traffic related inputs, one of which is the vehicle miles traveled (VMT). The model requires detailed estimates of VMT such as VMT by vehicle class, VMT by hour, VMT by facility type, and VMT by speed. The national default values are available, but EPA suggests the use of local data wherever possible. The details of these input requirements are described below:

2.2.1 VMT by Vehicle Class

The VMT fraction is the fraction of total highway VMT accumulated by each of the 16 composite vehicle types shown in Table 2(b). The input must consist of 16 fractional values, which are the fraction of total highway VMT accumulated by each of 16 composite vehicle types. All values must be between 0 and 1, and the 16 values must sum up to 1.

2.2.2 VMT by Facility

VMT must also be identified by various roadway or facility type (4 types) for each of the 28 vehicle classes as shown in Table 2(a) for each of the 24 hours of the day. The facility types used in this input requirement are freeway, arterial, local and ramp. For each vehicle class, there must be four sets of 24 values, corresponding to the four
facility types: freeway, arterial, local, and ramp. The distributions for each hour must add up to 1.

**Table 2(a) Complete MOBILE6 Vehicle Classification**

<table>
<thead>
<tr>
<th>Vehicle Type</th>
<th>Individual Vehicle Type - Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDGV</td>
<td>Light-Duty Gasoline Vehicles (Passenger Cars)</td>
</tr>
<tr>
<td>LDGT1</td>
<td>Light-Duty Gasoline Trucks 1 (0-6,000 lbs. GVWR, 0-3750 lbs. LVW)</td>
</tr>
<tr>
<td>LDGT2</td>
<td>Light Duty Gasoline Trucks 2 (0-6,001 lbs. GVWR, 3751-5750 lbs. LVW)</td>
</tr>
<tr>
<td>LDGT3</td>
<td>Light Duty Gasoline Trucks 3 (6,001-8500 lbs. GVWR, 3751-5750 lbs. LVW)</td>
</tr>
<tr>
<td>LDGT4</td>
<td>Light Duty Gasoline Trucks 4 (6,001-8500 lbs. GVWR, 3751-5750 lbs. LVW)</td>
</tr>
<tr>
<td>HDGV2B</td>
<td>Class 2b Heavy Duty Gasoline Vehicles (8501-10,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV3</td>
<td>Class 3 Heavy Duty Gasoline Vehicles (10,001-14,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV4</td>
<td>Class 4 Heavy Duty Gasoline Vehicles (14,001-16,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV5</td>
<td>Class 5 Heavy Duty Gasoline Vehicles (16,001-19,500 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV6</td>
<td>Class 6 Heavy Duty Gasoline Vehicles (19,501-26,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV7</td>
<td>Class 7 Heavy Duty Gasoline Vehicles (26,001-33,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV8A</td>
<td>Class 8a Heavy Duty Gasoline Vehicles (33,001-60,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDGV8B</td>
<td>Class 8b Heavy Duty Gasoline Vehicles (&gt;60,000 lbs. GVWR)</td>
</tr>
<tr>
<td>LDDV</td>
<td>Light Duty Diesel Vehicles (Passenger Cars)</td>
</tr>
<tr>
<td>LDDT12</td>
<td>Light Duty Diesel Trucks 1 (0-6,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV2B</td>
<td>Class 2b Heavy Duty Diesel Vehicles (8501-10,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV3</td>
<td>Class 3 Heavy Duty Diesel Vehicles (10,001-14,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV4</td>
<td>Class 4 Heavy Duty Diesel Vehicles (14,001-16,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV5</td>
<td>Class 5 Heavy Duty Diesel Vehicles (16,001-19,500 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV6</td>
<td>Class 6 Heavy Duty Diesel Vehicles (19,501-26,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV7</td>
<td>Class 7 Heavy Duty Diesel Vehicles (26,001-33,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV8A</td>
<td>Class 8a Heavy Duty Diesel Vehicles (33,001-60,000 lbs. GVWR)</td>
</tr>
<tr>
<td>HDDV8B</td>
<td>Class 8b Heavy Duty Diesel Vehicles (&gt;60,000 lbs. GVWR)</td>
</tr>
<tr>
<td>MC</td>
<td>Motorcycles (Gasoline)</td>
</tr>
<tr>
<td>HDGB</td>
<td>Gasoline Busses (School, Transit and Urban)</td>
</tr>
<tr>
<td>HDDBT</td>
<td>Diesel Transit and Urban Busses</td>
</tr>
<tr>
<td>HDDBS</td>
<td>Diesel School Busses</td>
</tr>
<tr>
<td>LDDT34</td>
<td>Light Duty Diesel Trucks 1 (6,001-8500 lbs. GVWR)</td>
</tr>
</tbody>
</table>

*GVWR: Gross Vehicle Weight Rating: The maximum weight of the vehicle when it is fully loaded as specified by the manufacturer
*LVW: Loaded Vehicle weight (weight of vehicle sitting empty (or vehicle curb weight) plus 300lb)
*ALVW: Alternative Loaded Vehicle Weight (: An adjusted weight obtained by the numerical average of vehicle curb weight and gross vehicle weight)
2.2.3 VMT by Hour

This command requires information, which is an aggregation of the information of VMT by Facility. The fraction of total VMT that occurs each hour of the day is specified in this command. This command requires values that are independent of facility type. 24 VMT fractions, one value for each of the 24 hours of the day must be supplied, and the 24 values must sum to 1.

2.2.4 VMT by Speed

This input specifies VMT by average speed (pre selected ranges) on freeways and arterial roads for each of the 24 hours of the day. Within each of the facility type (freeways and arterials) and each hour of the day, travel in terms of VMT is divided into 14 speed bins. The 14 speed ranges are allocated as follows. The first bin is “idle” which is below 5 mph. From 5 mph to 65 mph there are 13 speed bins, at 5 mph intervals, making a total of 14 bins, with the last bin catering for speeds of 65 mph and above. The fourteen average speed fractions must add up to 1.

2.3 Review of Current VMT Estimation Methods

The national default values for the VMT fractions by functional class, speed and hour of the day were developed by aggregating the data obtained in different locations

<table>
<thead>
<tr>
<th>Number</th>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>LDV</td>
<td>Light-Duty Vehicles (Passenger Cars)</td>
</tr>
<tr>
<td>2</td>
<td>LDT1</td>
<td>Light-Duty Trucks 1 (0-6,000 lbs. GVWR, 0-3,750 lbs. LVW)</td>
</tr>
<tr>
<td>3</td>
<td>LDT2</td>
<td>Light-Duty Trucks 2 (0-6,000 lbs. GVWR, 3,751-5,750 lbs. LVW)</td>
</tr>
<tr>
<td>4</td>
<td>LDT3</td>
<td>Light-Duty Trucks 3 (6,001-8,500 lbs. GVWR, 0-5,750 lbs. ALVW+)</td>
</tr>
<tr>
<td>5</td>
<td>LDT4</td>
<td>Light-Duty Trucks 4 (6,001-8,500 lbs. GVWR, 5,751 lbs. and greater ALVW)</td>
</tr>
<tr>
<td>6</td>
<td>HDV2B</td>
<td>Class 2b Heavy-Duty Vehicles (6,501-10,000 lbs. GVWR)</td>
</tr>
<tr>
<td>7</td>
<td>HDV3</td>
<td>Class 3 Heavy-Duty Vehicles (10,001-14,000 lbs. GVWR)</td>
</tr>
<tr>
<td>8</td>
<td>HDV4</td>
<td>Class 4 Heavy-Duty Vehicles (14,001-16,000 lbs. GVWR)</td>
</tr>
<tr>
<td>9</td>
<td>HDV5</td>
<td>Class 5 Heavy-Duty Vehicles (16,001-19,500 lbs. GVWR)</td>
</tr>
<tr>
<td>10</td>
<td>HDV6</td>
<td>Class 6 Heavy-Duty Vehicles (19,501-26,000 lbs. GVWR)</td>
</tr>
<tr>
<td>11</td>
<td>HDV7</td>
<td>Class 7 Heavy-Duty Vehicles (26,001-33,000 lbs. GVWR)</td>
</tr>
<tr>
<td>12</td>
<td>HDV8A</td>
<td>Class 8a Heavy-Duty Vehicles (33,001-60,000 lbs. GVWR)</td>
</tr>
<tr>
<td>13</td>
<td>HDV8B</td>
<td>Class 8b Heavy-Duty Vehicles (&gt;60,000 lbs. GVWR)</td>
</tr>
<tr>
<td>14</td>
<td>HDBS</td>
<td>School Buses</td>
</tr>
<tr>
<td>15</td>
<td>HDBT</td>
<td>Transit and Urban Buses</td>
</tr>
<tr>
<td>16</td>
<td>MC</td>
<td>Motorcycles (All)</td>
</tr>
</tbody>
</table>
from travel demand studies and traffic count data sets. Since individual areas have different functional classification systems, the EPA had to merge these road classes into four functional classes required for the MOBILE6. The data obtained from five urban areas namely, Chicago, Houston, Charlotte, Ada county (Boise region) and New York were used to develop VMT default values for MOBILE6. The results for Chicago, Houston and Boise were obtained from travel demand models while, the results for Charlotte and New York were obtained from traffic counts. These cities were used as prototypes and the EPA developed an aggregation procedure to develop VMT by speed, facility class and hour of the day to obtain the national default values.

Most state and local transportation agencies have their own traffic count databases. Some of these counts may be collected as part of the HPMS. Traffic count data are generally maintained at hourly or daily resolution and in some cases they are available at 15-minute intervals (EPA, 2001a).

EPA suggests the following steps in the estimation of VMT by facility class (EPA, 2001a).

1. Calculate the sum of counts in each functional class
2. Determine the sample size in each functional class
3. Determine the average volume by dividing the total count by the sample size
4. Obtain miles of facility in each class (from existing databases of the network)
5. Calculate VMT by class as average volume multiplied by number of miles of facility.

MOBILE6 requires VMT for arterials and freeways be further disaggregated by speed. EPA suggests the use of the Bureau of Public Roads (BPR) method for this process. In this method, the volume/capacity relationship expressed in the BPR curves is used to estimate the speed on each link.

Another source of estimating vehicle activity by functional class, time of day and speed is the use of travel demand models. Land use data, network data and data of diurnal factors are the inputs to the four-step process in travel demand modeling. Land use forecasting models are used to allocate future population and employment growth based on accessibility measures and attractiveness of each zone. The highway network
represents only the major roadways and all inter-zonal travel is assumed to occur on this network. The diurnal factors are used to develop the traffic by hour of the day as a percent of total daily traffic.

These factors are estimated using traffic counts on travel surveys. The estimated factors from traffic counts are usually classified by facility/area type and orientation of link. The factors based on travel survey data are distinguished by trip purpose (TMIP, 1994).

The following data are available from travel demand models:

1. Link specific traffic volumes. From these volumes, VMT can be calculated by multiplying link volumes by link length.
2. Average speed on each link. This can be obtained by dividing the link length by the travel time on the corresponding link.
3. Zone to zone trips by purpose and trip ends by TAZ and trip purpose.

Another means of estimating vehicle activity by functional class, time of day and speed is to use link volume predictions from travel demand models. The procedure recommended by the EPA to estimate VMT from travel demand model estimate is ((EPA, 2001a):

1. Distribute link level volumes by hour of the day
2. Calculate hourly VMT by multiplying link distribution by hourly volume.
3. Find the v/c ratio, using link specific capacities.
4. Apply the BPR curve, using link specific free flow speed or look up tables, to arrive at hourly-congested speeds.

Kumapley and Fricker (1996) classified VMT estimation methods into two broad divisions: the non-traffic-count-based methods and the traffic-count based methods. Both methods have their advantages and also shortcomings. Non-traffic-count based methods normally serve as rough estimation since the accuracy of the needed information is normally not guaranteed. Traffic-count-based methods are the commonly used ones, which mostly depend on the samples from the road network. However, it is normal practice to apply the sampled traffic counts directly onto other links within the same road class, and then estimate VMT.
2.3.1 Non-Traffic Count Based Methods

Besides travel demand model methods, VMT can be estimated by non-traffic count methods such as the fuel sales based method, the odometer recording based method, the Nationwide Personal Transportation Survey (NPTS) data based method, travel simulation models based method, etc.

Erlbaum (1989) proposed the fuel sales based VMT estimation method, which involves the use of information on retail gasoline and diesel fuel sales in price per gallon of fuel, and estimates of fleet fuel efficiency. This method was first used in 1957 and was the basis for early VMT submittals to FHWA. The accuracy of VMT estimation by this method is dependent on the accuracy of retail fuel sales data and the fleet fuel efficiency figure used. Kumapley and Fricker (1996) suggested using this method with caution, with its estimates being considered as rough or preliminary estimates.

By the odometer recording based VMT estimation, VMT may be obtained by summing the odometer recordings for all vehicles provided the odometer readings are available and there are no errors for all vehicles. However, sampling and extrapolating are necessary. So, Kumapley (1994) states that this method is too resource intensive and the possible error sources include the odometer calibration errors, reporting errors, secondary-party readings or transcription errors, odometer rollovers, vehicle drop-out caused by accidents or aged vehicles, and out-of-area travel likely to be considered in area travel. These error sources limit the use of this method.

Maring et al. (1974) and Greene (1987) developed cross-classification models based on the NPTS data to predict VMT using demographic predictions of the population of licensed drivers with respect to defined age groups and gender. They relied on the constancy of some demographic patterns, such as the distribution of population and licensed drivers and average annual miles traveled, to predict highway vehicle travel as a function of demographic forecasts. In Maring’s (1974) and Greene’s (1987) studies, NPTS data were used to determine annual VMT per driver, distribution of licensed drivers, and trends in VMT per driver.

Maring’s (1974) results indicate that information from the NPTS has the potential to generate stable and accurate VMT forecasts. Kumapley and Fricker (1996) state that on the basis of the results of Maring’s and Greene’s research, a short-term cross-
classification VMT forecasting model was developed for Indiana Department of Transportation (INDOT) using the information from the 1990 NPTS, which has data specific to Indiana. The INDOT model generates statewide VMT forecasts in Indiana to supplement INDOT traffic count–based VMT estimates.

Colman and Aronson (1992) used travel demand models to estimate VMT. Because of the good availability and low cost, there is a tremendous increase of the use of travel demand models such as EMME/2, QRS II, TRANSPLAN, TMODEL, UTPS, and MINUTP, in urban areas. After the forecasted link traffic volumes are obtained for a network on the basis of relevant socioeconomic parameters and the link centerline mileage is known, daily VMT can be estimated for the study area modeled. This method depends highly on the accuracy of the modeling and calibration during the entire four steps of traffic forecasting in geographic aggregation, trip generation, trip distribution, modal split, and traffic assignment.

2.3.2 Traffic-Count Based Methods

Traffic count-based VMT estimation methods use actual count data. The Environment Protection Agency (EPA) prefers the FHWA traffic count-based Highway Performance Monitoring System (HPMS) method for estimating VMT.

The HPMS method of VMT estimation involves the use of adjusted 24-hr traffic counts (referred to as annual average daily traffic AADT) obtained on sample sections identified through a systematic stratified random sampling process. Detailed HPMS VMT calculation is included in the HPMS field manual. The sample section VMT is estimated as the product of the section AADT and road segment mileage. The sample section VMT is expanded using expansion factors to obtain the area-wide and universal VMT estimates.

Bhat and Nair (2001), developed a fractional split model that predicts the VMT mix on links as a function of the functional roadway classification of the link, the physical attributes of the link, the operating conditions on the link, and the attributes of the traffic analysis zone in which the link lies. They did an empirical analysis to estimate a model for the Dallas-Fort Worth region in Texas. Five sets of independent variables were included in the model to predict the VMT mix on links. These are: a) link specific functional classification, b) link physical attributes, c) link free speed variables, d) degree
of urbanization of zone, e) zonal land use characteristics. The results of the proposed model and those of the default model in use by the Metropolitan Planning Organizations were compared with the actual observed VMT distribution values. The model evaluation results indicated that proposed model provided better predictions of VMT distribution values on links than the default model.

Qiao and Yu (2004) developed an alternative approach that incorporates both traffic count and link attributes through an exponential regression model in estimating volume and VMT. This method was an alternative approach to the Traffic Count Method suggested by the EPA, in which the average traffic counts in a certain road functional class are applied to all other links with the same class where the differences among link attributes were ignored. In this method, the traffic counts on the other links were applied by incorporating link attributes. They did an empirical analysis to estimate a model for the Southwest Houston area in Texas. A total of four link attributes were selected in the mode; to predict VMT on other (unknown) links. They were: a) link length b) mean speed c) night speed limit d) land use type. To validate the model they conducted three scenarios and compared the estimated results. Scenario 1 used the proposed experimental regression model to estimate link volume and link VMT. In Scenario 1 the real link volume was only used for comparison purpose. Scenario 2 estimated VMT on each link based on the EPA Traffic Count Method (TCM). Scenario 3 calculated the VMT on all links by multiplying link volumes and their corresponding lengths, then adding them to get the entire VMT. They named this Scenario as the Full Size VMT. To better compare the results among the three scenarios, they calculated the relative errors of the estimates by model and by TCM and for comparison they used the Full Size VMT. The model evaluation results indicated that proposed model provides best estimation factors among all methods of VMT estimation methods.

The Indiana Department of Transportation (INDOT) does not use the Highway Performance Monitoring System (HPMS) data submittal software for VMT estimation. Like the HPMS estimation method, the INDOT procedure is traffic count–based and follows the method outlined in the HPMS manual. However, because traffic data are a critical factor affecting the accuracy of VMT estimated using traffic count data, INDOT uses its own extensive inventory data base, which holds about 620,000 records of traffic

11
data for estimating VMT. The INDOT estimates are therefore expected to be more accurate than estimates that would be obtained from the HPMS, which is based on only about 4,000 records for Indiana.

The INDOT VMT estimation procedure, which is implemented in a spreadsheet, calculates by functional system the vehicle miles that are traveled in a 24-hr period. The sample section VMT estimates are then obtained by multiplying the sample section AADT to the section length. The sample section VMT estimates are then multiplied by expansion factors to transform or extrapolate the data to represent the entire universe and area-wide functional subsystems for rural, small urban and urbanized areas.

The roadway data input to emissions calculations for Pennsylvania uses information from the Roadway Management System (RMS) maintained by the Pennsylvania Department of Transportation (PENNDOT)’s Bureau of Planning and Research. PENNDOT obtains this information from periodic visual and electronic traffic counts. RMS data is dynamic since it is continually reviewed and updated from new traffic counts and field visits conducted by PENNDOT. Information on roadways included in the National Highway System is reviewed at least annually, while information on other roadways is reviewed at least biennially. However, according to the EPA recommended practice, baseline inventory VMT computed from the RMS highway segment volumes should be adjusted to be consistent with Highway Performance Monitoring System (HPMS) VMT totals. The HPMS VMT reported for Pennsylvania is a subsystem of the RMS established to meet the data reporting requirements of the Federal Highway Administration (FHWA) and to serve as PENNDOT’s official source of highway information. Adjustment factors are calculated which adjust the 1990 RMS VMT to be consistent with 1990 HPMS totals. In addition, the base year RMS download (in this case 1999) is also adjusted to the reported HPMS totals for that year. These factors are developed for each county, urban/rural code and functional class combination and are also applied to all future year runs.

2.4 Summary

EPA provides extensive documentation, which is helpful for understanding the VMT input requirements to MOBILE6, and the methodologies developed by the EPA in preparing the national default values. EPA used different data sets from different regions
and aggregated the results to obtain the national default values. Travel demand models, traffic counts, and travel surveys are some of the primary data sources in preparing the national default values. However, these sources are not exclusively developed for the purpose of air quality modeling and are not accurate in supplying the detailed vehicle activity data required for MOBILE6. EPA suggests that agencies preparing emission inventories should use local data in MOBILE6 and encourages innovative methods and modeling efforts for supplying vehicle activity data to MOBILE6.

Several alternative methods were developed in supplying accurate vehicle miles traveled data to MOBILE6. The different methods which have been developed may result in varying estimates of VMT for the same road network. According to Kumapley and Fricker (1996), VMT estimation methods can be classified into two broad divisions: the non-traffic-count-based methods and the traffic-count based methods. Both methods have their advantages and drawbacks. Non-traffic-count based methods normally serve as rough estimation since the accuracy of the information required is normally not guaranteed. Traffic-count-based methods are commonly used, which mostly depend on the samples from the road network. Some authors modeled VMT by vehicle type as a function of functional classifications, operating conditions on link and zonal characteristics.

The main drawbacks in using the traditional data sources can be described as follows: Speed estimates of travel demand models are inaccurate. Hence the VMT by facility class values obtained by these models may be erroneous. The VMT estimates by the travel survey data are subject to error, as they depend on reposted information and modal estimates of travel by the respondents. EPA suggests the use of traffic ground counts or network-based transportation models in the estimation of actual vehicle miles traveled. However, network-based travel demand models, with their time-consuming and resource-intensive procedures, are not a practical alternative to HPMS as an annual VMT estimation method. Hence the use of HPMS traffic count data is an attractive means of estimating VMT.
3. Objectives of the Study

The main objectives of this study are to:

1. Develop a comprehensive classification to estimate VMT using the HPMS data for Louisiana.

2. Test whether VMT estimates obtained using the Classification and Regression Tree (CART) classification above provides more accurate estimates than the conventional method used in HPMS Software.

3. Develop an add-in to TransCAD that will use the Classification system above to (a) estimate system VMT, and (b) estimate individual link VMT.

4. Demonstrate the procedure by applying the methodology to the East Baton Rouge non-attainment parish.
4. HPMS Data

The HPMS is a national system for collecting, maintaining, and reporting data detailing, “...the extent, condition, performance, use, and operating characteristics of the Nation's highways,” (HPMS Field Manual, 2000). Obtaining comprehensive, high quality data for HPMS reporting is critical to the financial stability of every state’s department of transportation because HPMS data are used for planning and apportioning various federal highway funds, including Transportation Equity for the 21st Century funds. Hence, HPMS provides a source of consistently formatted, high quality data for vehicle classification. Additionally, the U.S. EPA recognizes the HPMS as a quality source of local fleet data (EPA, 2002).

While a variety of different types of data are collected as part of the HPMS system, the data relevant to this research is traffic count data. Moreover, there are specific provisions of the HPMS system for traffic counting stations in and around ozone non-attainment areas (HPMS Field Manual, 2000). Such stations are commonly referred to as donut stations in HPMS parlance. Two types of traffic counts are performed for HPMS: total traffic counts and classified counts. For the purposes of fleet characterization, the classified counts are of primary interest.

As discussed earlier, the HPMS method of VMT estimation involves the use of adjusted 24-hr traffic counts, referred to as annual average daily traffic (AADT) obtained on sample sections identified through a systematic stratified random sampling process. The sample section VMT is estimated as the product of the section AADT and road segment mileage. The sample section VMT is expanded using expansion factors to obtain the areawide and universal VMT estimates. FHWA in the HPMS Field Manual (2002), specified as to how the State departments of transportation should obtain HPMS traffic count data and other data items along with the counts and how to estimate VMT by conventional method. The details are discussed below:

4.1 Obtaining the HPMS Traffic Count Data

4.1.1 General

State maintenance of an adequate traffic counting program is a primary HPMS concern. A State’s count program should cover all Interstate, principal arterial, other
National Highway System (NHS), and HPMS sample sections on a 3-year maximum cycle. It should cover all roads, not just State-owned roads, and should include counts on those systems made on the State’s behalf by MPOs, cities, or counties. The program should provide for a sufficient number of automatic traffic recorder (ATR) stations and classification count stations to permit factoring of 48-hour tube-type counts to estimates of annual average daily traffic (AADT). The State should also have an underlying short count program that assures that traffic counts on all roads on all systems are obtained over a longer term (6-year maximum) cycle for basic traffic monitoring purposes.

4.1.2 Volume Group Assignments

The State’s comprehensive traffic count program can be used to develop traffic volume group assignments for all road sections if the program has been formulated to adequately monitor both high and low volume roads, including those off the State system. To facilitate this process, count station locations should be selected to represent expected AADT volume group breakpoints for the volume ranges of both the standard and nonattainment area samples. This may require locating count stations at one per 5 to 10 miles in rural areas and more closely in urban areas; for homogeneous traffic sections, more than one section may be represented by a single traffic count station (HPMS Field Manual, 2002). Selection of count station locations should be based on previous count experience, recent land developments, and the existence of uncounted sections along the routes.

Generally, traffic counts in addition to those taken for the HPMS are needed to establish the assignment of road sections to their respective volume groups in both rural and urban areas. A well-designed comprehensive count program that includes off-State system roads should provide the needed additional counts. Traffic mapping techniques also can be applied to maximize the use of HPMS universe and sample counts, other coverage counts, and counts taken for project planning or operational purposes in assigning road sections to volume groups.

4.1.3 Traffic Monitoring Guide

A detailed discussion of recommended procedures for developing reliable estimates of travel characteristics, including AADT, is contained in the *Traffic
Monitoring Guide (TMG, 2001). However, a general discussion of some elements of a typical traffic volume count program and their applicability to the HPMS, as described in the HPMS Field Manual (2002) is as follows:

4.1.3.1 Continuous ATRs

Automatic Traffic Recorders (ATRs) are used to provide continuous traffic count coverage at selected locations. ATR data are also used to develop seasonal or monthly, day-of-week, and growth factors which are then used to adjust short coverage counts to AADT. Analytical procedures to determine the appropriate level of effort and to develop the needed traffic estimates are described in the TMG (Traffic Monitoring Guide, 2001).

Continuous count data are essential for converting coverage counts to AADT. The State’s documentation of its continuous count program should discretely identify the number of continuous counters on the rural and urban portions of the Principal Arterial System (PAS)/NHS system. Whenever possible, the State should have at least one continuous counter on each major PAS/NHS highway route. At a minimum, each continuous counter should have at least two full days of data for each day of the week, for each month (HPMS Field Manual, 2002).

4.1.3.2 Short Counts

Although short counts cover shorter periods, the TMG (2001) recommends using 48-hour counts (two full 24-hour days) for all HPMS universe, standard sample, and donut area sample sections whether on or off the State-owned highway system. Short counts should be randomly scheduled geographically throughout the State and temporally throughout the calendar year to ensure adequate representation and to minimize bias. Where axle correction factors are needed to adjust raw counts, they should be derived from facility specific vehicle classification data obtained on the same route or on a similar route with similar traffic in the same area. Factors that purported to account for suspected machine error in high traffic volume situations should not be applied to HPMS traffic counts or traffic count programs used for HPMS purposes, such as volume group assignment. In high volume situations, such as controlled access facilities, it is more
appropriate to use ramp counts in conjunction with strategic mainline counting, or other appropriate technology, than to use short counts and adjustment factors.

4.1.3.3 Vehicle Classification Data Collection

Data reported in the HPMS should reflect the use of statistically valid data collection procedures employing automatic vehicle classification equipment. More details on vehicle classification data reporting requirements are outlined in HPMS Field Manual (2002). Axle corrections based on vehicle classification data should be applied to all counts where the counting device uses axle sensors. State documentation of the vehicle classification activity should illustrate that:

a) Classification data are representative of specific functional systems.
b) Each season of the year is represented in the development of axle correction factors.
c) Classification sessions are long enough to account for the changes in vehicle mix from day to day. The TMG recommends that vehicle classification sessions be at least 48-hours. Data for less than 24 continuous hours is not appropriate.
d) The total volume of vehicles observed is at least equal to that for an average day.
e) Classification counts are well distributed among rural and urban locations.
f) Classification counts are collected, at a minimum, over a 3-year cycle, one-third of the counts per year.
g) There are sufficient classification categories to represent vehicles with two to seven axles.

4.1.4 Application of Traffic Count Procedures to HPMS

Traffic count data reported for all Interstate, other principal arterial and other NHS sections provide most of the travel data used for apportionment and other purposes. The HPMS standard sample design provides an appropriate statistical base for the development of traffic estimates for each sampled section and of system wide travel for the sampled systems (HPMS Field Manual, 2002). By incorporating vehicle classification and truck weight information, the TMG structure provides the capability of
estimating classified VMT and Equivalent Single Axle Loads (ESALs) from the HPMS data.

One-third of the Interstate, other principal arterial, and NHS road sections should be counted each year. In addition, one-third of the HPMS standard sample sections on each functional system should be counted each year. The sections to be counted should be randomly selected. Samples should be selected from each sample stratum (volume group), although minor adjustments may be necessary for strata with numbers of sections not divisible by three or having less than three samples. A single count may be used for several sections between adjacent interchanges on controlled access facilities (HPMS Field Manual, 2002).

The development of section AADT estimates from count data must include the use of short count and other appropriate adjustment factors. AADTs reported to the HPMS for standard sample and non-sample PAS/NHS sections not counted during the current year must be updated to current AADT estimates by use of appropriate growth factors.

Estimates of Daily Vehicle-Miles of Travel (DVMT) can be developed by direct computation for the Interstate, other principal arterials, and other NHS sections and by expansion of the HPMS standard sample on a functional system basis for other systems. This is done by multiplying the standard sample section AADT by the section length and by the standard sample expansion factor and summing the result to the HPMS stratification level desired (functional system, total rural, etc.); the HPMS software will perform these calculations by functional system. Since HPMS standard sample expansion procedures are based on the ratio of universe to sample mileage, mileage totals at any stratification level should be exact.

A comprehensive count program, good count practices, a well-distributed HPMS standard sample, and appropriate AADT estimation techniques will result in highly reliable DVMT estimates (HPMS Field Manual, 2002). The same procedures can be used in preparing and reporting count based AADT data for donut area sample sections; a unique expansion factor for each applicable NAAQS nonattainment area is used to prepare DVMT estimates (HPMS Field Manual, 2002).
4.2 HPMS VMT Calculation Process

This section deals with the conventional method of estimating VMT from the HPMS traffic count data. The detailed HPMS VMT calculation, including the calculation of expansion factors, can be found in the HPMS field manual (2002); a brief overview of the process is provided in this section.

The VMT calculation process for sample sections is carried out by SELSUMT, a subprogram of the HPMS submittal software that calculates by functional system for each sample section the vehicle miles that are traveled in a 24-hr period (referred to as daily VMT, or DVMT). For areawide VMT estimates, expansion factors are used to transform or extrapolate the sample section data to areawide VMT estimates to represent areas such as the National Ambient Air Quality Standards (NAAQS) nonattainment areas for the respective functional classification system and geographical area. The expansion factors are obtained using the following equation:

\[
EF_{ij} = \frac{(TMAVG)_{ij}}{(TMSVG)_{ij}}
\]

where,

- \(EF_{ij}\) = expansion factor for geographical area \(l\) for volume group \(i\) sampled in functional class \(j\),
- \((TMAVG)_{ij}\) = total mileage in geographical area \(l\) for volume group \(i\) sampled in functional class \(j\),
- \((TMSVG)_{ij}\) = total mileage in all sections in geographical area \(l\) for volume group \(i\) in functional class \(j\)

The expansion factors are unity for functional classes such as the Interstates that are sampled 100 percent. The computerized procedure incorporated in the HPMS software for calculating expansion factors is the SELEXPF option, which generates a summary table of expansion factors by volume group within each functional system and geographical area classification.

For universe VMT estimates, expansion factors are employed to transform or extrapolate the sample section data and VMT estimates to represent each functional class
and geographical area. The HPMS statewide VMT estimation is calculated using the following equation:

$$DVMT_s = \sum_l \sum_i \sum_j \sum_k DVMT_{ijk} \times EF_{lij}$$

where,

- $DVMT_s$ = statewide VMT estimate for geographical area $l$ and functional class $j$,
- $DVMT_{ijk}$ = VMT for sample section $k$ in geographical area $l$ for volume group $i$ of functional class $j$, and
- $EF_{lij}$ = expansion factor for geographical area $l$ for volume group $i$ of functional class $j$

The units of $DVMT$ are daily miles traveled, in thousands. $DVMT$ is estimated separately for rural, small urban and urbanized areas for each road functional class.

### 4.3 HPMS Data Items

#### 4.3.1 HPMS Data

The following are the details of HPMS data as specified in the HPMS Field Manual (2002):

##### 4.3.1.1 Universe Data

Universe data is the data representing total system length including National Highway System length not yet built or open to traffic. These data consist of a complete inventory of length (kilometers or miles) by functional system, jurisdiction, geographic location, (rural, small urban, urbanized, and NAAQS nonattainment areas) and other selected characteristics. Universe data fully reflect all open-to-traffic public roads in the State and contain basic information for planned, unbuilt future NHS.

The FHWA requires certain basic inventory information to be reported for all open-to-traffic, public road systems in the universe portion of the HPMS data set. There are 46 data items in this basic inventory. Limited universe data for facilities on planned National Highway System (NHS) roadways that are not yet built or open to traffic also are to be reported. Universe data must be reported on a section-by-section basis for all rural arterials, urban principal arterials, the NHS and all standard sample and donut area supplemental sample sections; universe data may be reported for the remaining functional systems on a grouped length basis.
4.3.1.2 Sample Data

Standard sample data consists of additional inventory, condition, use, pavement, operational, and improvement data that complement the universe data for those sections of roadway that have been selected as standard samples. When expanded through use of an appropriate expansion factor, the data represents the entire universe from which the sample was drawn, permitting evaluation of highway system performance. The sample sections form nominally "fixed" panels of road segments that are monitored on an established cyclical basis. Samples can be added or deleted from the sample panels as the need arises.

Panels of roadway sections are established using a statistically designed sampling plan based on the random selection of road segments at predetermined precision levels. The sample is stratified by area, by functional system, and by traffic volume group. Sample selection is done randomly within each stratum i.e., within each volume group for each arterial and major collector functional highway system in rural, and for each arterial and collector functional system in small urban and urbanized areas of the State.

Unique sampling is required for each urbanized area having >200,000 population and smaller urbanized areas that are NAAQS nonattainment areas. Rural and small urban areas (5,000 to 49,999 population) are sampled collectively statewide.

4.3.1.3 Donut Sample Data

An additional sample is required for the donut area portions of NAAQS nonattainment areas that use the HPMS as the basis of VMT estimates for air quality travel tracking and conformity purposes. The donut area supplementary sample consists of sections in the rural and small urban minor arterial, rural major collector, and small urban collector functional systems that are located outside of urbanized area(s), but within the nonattainment area boundary. The donut area supplementary sample is used to enhance the existing standard sample to achieve higher confidence levels for travel estimates. The purpose of the donut area supplementary sample is limited to the development of travel estimates. The combination of existing standard samples in the donut area plus the donut area supplementary samples makes up the donut area sample. In addition to the universe data items, donut area supplementary sample sections need only have sample data Items 47 and 48 coded.
4.3.1.4 Summary Data

These data consist of annual summary reports for certain data not included in the HPMS universe and sample data set for the minor collector and local functional systems. Summary data must be coded manually onto the several summary screens contained in the HPMS submittal software. These additional data are derived from State and local sources such as statewide highway databases, management systems, Intelligent Transportation Systems (ITS) and traffic monitoring systems, and data made available from local governments and MPOs.

As discussed earlier, the HPMS database contains information on 46 data items for the universe sections and 98 data items for the sample sections related to the nation’s highway system. In this research, we are interested in identifying the variables that can effectively distinguish the homogeneous traffic volume sections. Appendix A contains the list of all the data items collected as a part of the HPMS data. However, only the details of those variables that can probably distinguish the traffic volume sections effectively are discussed below:

As discussed earlier, the target variable for the classification analysis will be Annual Average Daily Traffic (AADT), and the potential predictor variables will be rural/urban designation, functional system, National Highway System (NHS), type of facility, speed limit, number of through lanes, shoulder width, and lane width.

4.3.2 HPMS Data Items Used in CART Classification System

4.3.2.1 Annual Average Daily Traffic (AADT)

The AADT is the average daily traffic volume on a facility in both directions of travel unless the street is one-way or part of a couplet. AADT is adjusted for weekday and seasonal factors so that it represents an average daily value that represents all days of the reporting year. Growth factors must be applied if the AADT is not derived from current year counts.

4.3.2.2 Rural/Urban Designation

Rural (R) Areas are those areas outside the boundaries of small urban and urbanized areas. The Bureau of Census defines rural areas as having a population of less than 5,000. Small Urban (S) areas are defined by the Bureau of Census as having a population of 5,000 to 49,999.
Urbanized (U) areas are defined as having a population of 50,000 to 199,999. HPMS codes for all rural/urban areas are shown in Table 4-1.

**Table 4-1 Rural/Urban Designation Codes in HPMS**

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Rural Area</td>
</tr>
<tr>
<td>2</td>
<td>Small Urban Area (Population 5,000 to 49,999)</td>
</tr>
<tr>
<td>3</td>
<td>Small Urbanized Area (Population 50,000 to 199,999)</td>
</tr>
<tr>
<td>4</td>
<td>Large Urbanized Area (Population 200,000 or More)</td>
</tr>
</tbody>
</table>

(Source: HPMS Workbook, 2002)

**4.3.2.3 Functional System**

The functional systems required by Title 23, U.S.C. legislation, have been chosen as the base for the HPMS and include the following systems:

**(a) Principal Arterials (PA)**

PA roads can be classified as roads that serve corridor movements having trip length and travel density characteristics of statewide or interstate travel. PA serves intra-area travel (such as between major inner-city communities, between central business districts and residential areas or between major suburban centers).

**Rural:** Rural PA routes provide an integrated network without stub connections except under certain conditions such coastal city or international boundary connections.

**Urban:** Urban PA routes carry the major portion of trips entering and leaving urban areas as well as the majority of through movements bypassing the central city.

**(b) Minor Arterials (MA)**

**Rural:** Rural MA roads are roads that link cities, larger towns and other important traffic generators such as resort areas into a built-in arterial network of arterial highways. This network provides intrastate and inter-county service that connects to rural collector or local roads.

**Urban:** Roads classified under Urban MA streets and roads interconnect and expand the PA system thus providing service for moderate distance and lower mobility travel. This MA system distributes travel to smaller geographic areas than the PA system and may
carry local bus routes. Thus, this system provides continuity within the community but does not go through neighborhoods.

(c) Collectors

**Rural:** Rural Collector routes are routes that generally serve travel of primarily within the county rather than statewide. The predominant travel distances are shorter than on arterial routes.

**Rural Major Collectors (MJC):** These routes provide service to county seats and larger towns that are not served by arterials. Also, these routes link other important intra-county traffic generators such as consolidated schools, shipping points, county parks and important mining and agricultural areas.

**Rural Minor Collectors (MNC):** These routes serve the chief intra-county corridors and developed areas within a reasonable distance of a major collector road. Also, these routes link the important local traffic generators in the rural areas.

**Urban Collectors (COL):** These routes provide land access service and traffic in the urban areas.

(d) Local Streets and Roads

Technically, the local streets and roads are not functionally classified.

**Rural Local (LOC):** These routes provide access to adjacent land and provide service to short distance travel.

**Urban Local (LOC):** These routes provide access to neighboring land or to functionally classified routes. These routes offer the lowest level of mobility and usually contain no bus routes. Also, through traffic movement is usually discouraged. HPMS contains codes for functional classification is shown in Table 4-2.

**Table 4-2 FHWA Approved HPMS Functional Classification and Codes**

<table>
<thead>
<tr>
<th>Rural Functional System Codes</th>
<th>Urban Functional System Codes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Principal Arterial Interstate (PAI)</td>
<td>11 Principal Arterial Interstate (PAI)</td>
</tr>
<tr>
<td>2 Other Principal Arterial (OPA)</td>
<td>12 Principal Arterial Other Fwys &amp; Exp (OFE)</td>
</tr>
<tr>
<td>6 Minor Arterial (MA)</td>
<td>14 Other Principal Arterial (OPA)</td>
</tr>
<tr>
<td>7 Major Collector (MJC)</td>
<td>16 Minor Arterial (MA)</td>
</tr>
<tr>
<td>8 Minor Collector (MNC)</td>
<td>17 Collector (COL)</td>
</tr>
<tr>
<td>9 Local (LOC)</td>
<td>19 Local (LOC)</td>
</tr>
</tbody>
</table>

(Source: FHWA Approved HPMS Workbook, 2002)
4.3.2.4 National Highway System (NHS)

The National Highway System is a federal-aid highway system initiated by (Intermodal Surface Transportation Equity Act (ISTEA) legislation and further outlined in the NHS Designation Act (NHSDA) of 1995. Due to an increased need for Intermodal Transportation analysis, the HPMS must identify NHS Intermodal connectors and distinguish the type of terminals they serve. HPMS data contains NHS codes as illustrated in Table 4-3.

4.3.2.5 Type of Facility

Type of facility is used to determine whether a segment is on a one- or two-way roadway or structure. It is used in investment requirements modeling to calculate capacity and estimate roadway deficiencies and improvement needs, in the cost allocation pavement model and in the NHS database. The HPMS codes for type of facility are as shown in Table 4-4.

Table 4-3 NHS Codes for HPMS Data

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>This Section Is Not On The NHS</td>
</tr>
<tr>
<td>1</td>
<td>This Section Is On The NHS But Is Not An NHS Intermodal Connector</td>
</tr>
<tr>
<td>2</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Airport</td>
</tr>
<tr>
<td>3</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Port Facility</td>
</tr>
<tr>
<td>4</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Amtrak Station</td>
</tr>
<tr>
<td>5</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Rail/Truck Terminal</td>
</tr>
<tr>
<td>6</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Intercity Bus Terminal</td>
</tr>
<tr>
<td>7</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Public Transit Or Multi-Modal Passenger Terminal</td>
</tr>
<tr>
<td>8</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Pipeline Terminal</td>
</tr>
<tr>
<td>9</td>
<td>This Section Is An NHS Intermodal Connector. Type Of Connector: Major Ferry Terminal</td>
</tr>
</tbody>
</table>

(Source: HPMS Workbook, 2002)
Table 4-4 Type of Facility Codes and Description for HPMS data

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>One-Way Roadway</td>
</tr>
<tr>
<td>2</td>
<td>Two-Way Roadway</td>
</tr>
<tr>
<td>3</td>
<td>One-Way Structure (Bridge, Tunnel, Causeway, etc.)</td>
</tr>
<tr>
<td>4</td>
<td>Two-Way Structure (Bridge, Tunnel, Causeway, etc.)</td>
</tr>
</tbody>
</table>

(Source: HPMS Workbook, 2002)

4.3.2.6 Shoulder Width

(a) Right (outside) Shoulder Width: It is the predominant width and does not include width for a right shoulder if it is a parking or bike lane.

(b) Left (median) Shoulder Width – It is the predominant width and occurs only where a divided highway and median exist. It does not include width for a left shoulder if it is part of a continuous left turn lane.

4.3.2.7 Speed Limit

This is the posted speed limit on sample sections and is used in investment modeling to estimate running speed and for other analyses, including delay estimation. If more than one speed limit exists, the predominant one is considered.

4.3.2.8 Number of Through Lanes

Number of through lanes provides basic inventory information on the amount of public road in use. It is used for apportionment, administrative, legislative, analytical and national highway database purposes. This does not include auxiliary lanes (collector/distributor lanes, weaving lanes, frontage road lanes, parking and turning lanes, acceleration/deceleration lanes, toll collection lanes and truck climbing lanes.

4.3.2.9 Lane Width

This is a measure of existing lane width on sample roadway sections. It is used for investment modeling to calculate capacity, estimate needed improvements, and compute a safety index, for cost allocation pavement models, and for other policy analysis and national highway database purposes.
5. Methodology

5.1 Introduction

The main part of this research was to develop a methodology that can estimate VMT by a comprehensive classification process based on link attributes. This classification was done using classification software SPSS Answer Tree 1.0. A GIS platform has been used to implement this methodology. The proposed methodology is explained in a detailed step-by-step process in the following sections.

5.2 Classification Procedure Using SPSS Answer Tree 1.0

Answer Tree is a computer learning system that creates classification systems displayed in decision trees. It is used to generate the classification rules from existing data. It provides four algorithms for performing classification and segmentation analysis. They are CHAID, Exhaustive CHAID, Classification and Regression Tree (C&RT/CART) and QUEST. The CART algorithm was used for performing classification in this analysis.

The classification procedure in this research was required to identify those features that could effectively distinguish the homogeneous traffic volume sections. CART is an exploratory data analysis method that is used to study the relationships between a dependent measure and a number of possible predictor variables that themselves may interact.

5.2.1 Data

The Louisiana HPMS traffic count data for the year 2003 was obtained from the Federal Highway Administration (FHWA). These data were representative of the type of data the EPA recommends to be used to estimate the VMT. The Louisiana 2003 HPMS data contained information related to 46 data items for the universe data and 98 data items for the standard sample data. The standard sample included data related to annual average daily traffic (AADT), rural/urban designation, functional system, national highway system (NHS), type of facility, speed limit, number of through lanes, shoulder width, lane width, etc., for all the HPMS sampled sections in Louisiana.

5.2.2 Identifying the CART Variables

The Louisiana 2003 HPMS standard sample data was selected for the classification as it contained a richer assortment of data items. Because the classification
procedure is entirely mechanical in nature and cannot anticipate the consequence of division of data at one level with that at another level, classification results are not necessarily globally optimal and can be sensitive to small differences in item values. In order to overcome this tendency to obtain solutions that are unique rather than representative, a process called “cross-validation” was used, in which data are divided into several subsets to produce multiple results (Wilmot and Shivananjappa, 2003). This process uses all subsets combined except one on each run, and rotates the excluded subset among all subsets until each subset has been excluded once. A tree is generated on each run and the average, or the most frequent classification scheme, is considered the appropriate classification so the cross-validation process consisted of 10 runs. In this study the data was split into 10 datasets.

To grow a classification tree in SPSS Answer Tree 1.0, the model must first be defined by selecting the target and predictor variables, and the classification procedure. In this case the target variable was AADT (defined as continuous) and the predictor variables were rural/urban designation (ordinal), functional system (ordinal), NHS (ordinal), type of facility (ordinal), speed limit (continuous), number of through lanes (continuous), left shoulder width (continuous), right shoulder width (continuous), lane width (continuous). The classification procedure chosen was the CART method.

After defining the model, the Growing Criteria for the tree must be specified. In this case, under the minimum number of cases of the stopping rules tab it was specified as 10 for parent node and 1 for child node. The minimum change in impurity was specified as 1.0. This produced a classification tree with 28 terminal nodes; an overview of the classification tree can be seen in the Tree Map as shown in Figure 5-1. The nodes display the mean, standard deviation, and the number of data records it could split and the improvement i.e., the measure of decrease in impurity for each predictor in each node with the use of each variable as shown in Figure 5-2. The automatically grown tree was then analyzed by examining the risk estimate summary tree and finding the proportion of variance captured by the classification procedure. The risk estimate, generally, is simply the within-node variance. The total variance equals the within-node (error) variance plus the between-node (explained) variance. The within-node variance and the total variance (the risk estimate for the tree with only one node are noted).
Figure 5-1 Tree Map Showing the Overview of the Classification Tree

Figure 5.2: Classification Tree in Answer Tree
Then, the proportion of the variance due to error is the within-node variance divided by the total variance. Thus, the proportion of variance explained by the model is \(100\% - (\frac{\text{within-node variance}}{\text{total variance}})\)%.

This was done to check how well the tree does at identifying those features that effectively distinguish the homogeneous traffic volume sections. The results of the evaluation are discussed in Chapter 7.

### 5.3 Deriving the CART Equation

After conducting the ten consecutive runs of the CART process, the variable splits were examined to identify the variables that consistently played an important role in distinguishing homogeneous traffic volume sections. These variables were identified and the VMT function was then established with the variables obtained. The details of the derived equation are given in Chapter 7. The derived VMT function was used to estimate VMT for the sampled as well as for the unsampled links in the network.

### 5.4 HPMS to MOBILE6

VMT estimates from the HPMS data had to be converted into the MOBILE6 input format as HPMS classifies vehicles by length and number of axles, while MOBILE6 classifies vehicles based on weight and fuel type. However, the Louisiana 2003 HPMS standard sample data did not contain any information regarding vehicle classification, speed, and hour of the day, and MOBILE6 requires detailed VMT inputs with respect this information. Hence, the national default MOBILE6 detailed VMT values for VMT by hour, speed, vehicle class, and facility type were used for all the links in the Louisiana network. However, if the data contained information related to vehicle classification, speed, and hour of day, then the procedure described below can be used to convert the VMT estimates obtained from the HPMS data into the detailed VMT inputs required by MOBILE6.

The HPMS system divides vehicles into 14 “bins” or classes, the first thirteen are listed in Table 5-1, and the fourteenth is an “unknown” bin where vehicle class cannot be determined. The classification is based on vehicle length and number of axles per vehicle.

The HPMS vehicle types can be translated into equivalent MOBILE6 vehicle types using conversion factors developed by the University Transportation Center for Alabama. The procedure involves using the factors in Table 5-2 to convert HPMS
vehicle counts (13 classes) into MOBILE6 vehicle counts (30 classes). The EPA suggested the 30 vehicle classes in 1999 and the conversion factors were made based on that classification.

Table 5-1 HPMS Vehicle Codes and Descriptions

<table>
<thead>
<tr>
<th>Vehicle Bin Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Motorcycles (Optional):</strong> All two- or three-wheeled motorized vehicles. Typical vehicles in this category have saddle type seats and are steered by handle bars rather than a wheel. This category includes motorcycles, motor scooters, mopeds, motor-powered bicycles, and three-wheeled motorcycles. This vehicle type may be reported at the option of the State, but should not be reported with any other vehicle type.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Passenger Cars:</strong> All sedans, coupes, and station wagons manufactured primarily for the purpose of carrying passengers and including those passenger cars pulling recreational or other light trailers. Vehicles registered as passenger cars that are pick-ups, panels, vans, etc. (described as vehicle type “3”) should be reported as vehicle type “3”.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Other Two-Axle, Four-Tire, Single-Unit Vehicles:</strong> All two-axle, four-tire vehicles, other than passenger cars. Included in this classification are pickups, panels, vans and other vehicles such as campers, motor homes, ambulances, hearses, and carryalls. Other two-axle, four-tire single unit vehicles pulling recreational or other light trailers are included in this classification.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Buses:</strong> All vehicles manufactured as traditional passenger-carrying buses with two-axles, six-tires and three or more axles. This category includes only traditional buses (including school buses) functioning as passenger-carrying vehicles. All two-axle, four-tire minibuses should be classified as other two-axle, four-tire, single-unit vehicles (type ”3”). Modified buses should be considered as trucks and be appropriately classified.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Two-Axle, Six-Tire, Single-Unit Trucks:</strong> All vehicles on a single frame including trucks, camping and recreational vehicles, motor homes, etc., having two axles and dual rear wheels.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Three-Axle, Single-Trailer Trucks:</strong> All vehicles on a single frame including trucks, camping and recreational vehicles, motor homes, etc., having three axles.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Four-or-More Axle, Single-Trailer Trucks:</strong> All vehicles on a single frame with four or more axles.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Four-or-Less Axle, Single-Trailer Trucks:</strong> All vehicles with four or less axles consisting of two units, one of which is a tractor or straight truck power-unit.</td>
</tr>
<tr>
<td>9</td>
<td><strong>Five-Axle, Single-Trailer Trucks:</strong> All five-axle vehicles consisting of two units, one of which is a tractor or straight truck power-unit.</td>
</tr>
</tbody>
</table>

1 Table Continued Below
10 **Six-or-More Axle, Single-Trailer Trucks**: All vehicles with six or more axles consisting of two units, one of which is a tractor or straight truck power-unit.

11 **Five-or-Less Axle, Multi-Trailer Trucks**: All vehicles with five or less axles consisting of three or more units, one of which is a tractor or straight truck power-unit.

12 **Six-Axle, Multi-Trailer Trucks**: All six-axle vehicles consisting of three or more units, one of which is a tractor or straight truck power-unit.

13 **Seven-or-More Axle, Multi-Trailer Trucks**: All vehicles with seven or more axles consisting of three or more units, one of which is a tractor or straight truck power-unit.

14 **Miss Count**: Vehicles that could not be accurately assigned to the 13 defined classes.

(Source: Federal Highway Administration, FHWA 2001)

However, in the MOBILE6 User’s Guide (EPA, 2002) vehicles are classified into 28 classes by combining LDDT1 and LDDT2 in the 30-vehicle classification as one class and LDDT3 and LDDT4 as another. EPA classified these as four different classes in the year 1999. Hence, care should be taken while translating the vehicles types to be classified into 28 classes i.e., the complete MOBILE6 classification and into 16 vehicle classes as required by the EPA as detailed VMT estimates.

The conversion factors provide a quick method for state and local analysts to convert their own vehicle classification data (obtained through HPMS) to MOBILE6 vehicle classes (UTCA, 2003). For example, if an HPMS station counted 100 “2-axle, 4-tire, single unit trucks,” the conversion factors from Table 5-2 can be used to apportion those 100 vehicles as: 20.87 LDGT1’s, 69.75 LDGT2’s, 2.733 LDGT3’s, 1.257 LDGT5’s, 1.56 HDGV2B’s, 0.052 HDGV3’s, 0.017 HDGV5’s, 0.057 HDGV5’s, 0.123 HDGV6’s, 0.050 HDGV7’s, 0.005 HDGB’s, 2.992 (2.050+0.952) LDDT3 &5’s, 0.127 HDDV2B’s, 0.038 HDDV3’s, 0.036 HDDV5’s, 0.017 HDDV5’s, 0.073 HDDV6’s, 0.105 HDDV7’s, 0.063 HDDV8A’s, 0.155 HDDV8B’s, 0.006 HDDBT’s, and 0.052 HDDBS’. Since the fractional counts are not possible, the translated counts are ultimately rounded.

A software package “HPMS2MOBILE” developed by the UTCA (2003) can be used to automate the translation process (application of Table 5-2). However, because this software classifies the vehicles into 30 classes, necessary steps should be taken in classifying the classes into 28 vehicle classes. After converting the vehicle types, the
HPMS VMT estimates need to be converted further into VMT by facility class, VMT by vehicle type, VMT by hour, and VMT by speed.

Table 5-2 Vehicle Type Conversion from HPMS to MOBILE6 (%)

<table>
<thead>
<tr>
<th>MOBILE6 Vehicle Type</th>
<th>Motorcycles</th>
<th>Passenger cars</th>
<th>2 Axle, 4 Tire Single Unit</th>
<th>Buses</th>
<th>2 Axle, 6 Tire Single Unit</th>
<th>3 Axle Single Units</th>
<th>4+ Axle Single Unit</th>
<th>1/4 Axle single trailer</th>
<th>5/4 Axle single trailer</th>
<th>6 Axle single trailer</th>
<th>4/5 Axle Multi trailer</th>
<th>6 Axle Multi trailer</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC</td>
<td>100.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LDGJ</td>
<td>0.000</td>
<td>98.800</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LDGJ1</td>
<td>0.000</td>
<td>0.000</td>
<td>20.870</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.014</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LDGJ2</td>
<td>0.000</td>
<td>0.000</td>
<td>69.750</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.045</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LDGJ3</td>
<td>0.000</td>
<td>0.000</td>
<td>2.733</td>
<td>13.770</td>
<td>6.765</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.014</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>LDGJ4</td>
<td>0.000</td>
<td>0.000</td>
<td>1.257</td>
<td>6.330</td>
<td>3.125</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV2</td>
<td>0.000</td>
<td>0.000</td>
<td>1.460</td>
<td>0.000</td>
<td>41.870</td>
<td>11.670</td>
<td>3.775</td>
<td>2.444</td>
<td>3.534</td>
<td>2.079</td>
<td>2.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV3</td>
<td>0.000</td>
<td>0.000</td>
<td>0.052</td>
<td>0.000</td>
<td>1.497</td>
<td>0.424</td>
<td>0.138</td>
<td>0.151</td>
<td>0.048</td>
<td>0.010</td>
<td>0.009</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV4</td>
<td>0.000</td>
<td>0.000</td>
<td>0.017</td>
<td>0.000</td>
<td>0.476</td>
<td>0.136</td>
<td>0.043</td>
<td>0.048</td>
<td>0.010</td>
<td>0.009</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.057</td>
<td>0.000</td>
<td>1.827</td>
<td>0.463</td>
<td>0.147</td>
<td>0.166</td>
<td>0.033</td>
<td>0.031</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV6</td>
<td>0.000</td>
<td>0.000</td>
<td>0.123</td>
<td>0.000</td>
<td>3.113</td>
<td>1.011</td>
<td>0.318</td>
<td>0.388</td>
<td>0.070</td>
<td>0.067</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV7</td>
<td>0.000</td>
<td>0.000</td>
<td>0.050</td>
<td>0.000</td>
<td>1.438</td>
<td>0.410</td>
<td>0.130</td>
<td>0.146</td>
<td>0.029</td>
<td>0.027</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV8</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.004</td>
<td>0.002</td>
<td>0.120</td>
<td>0.146</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV10</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV11</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV12</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV13</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV14</td>
<td>0.000</td>
<td>0.000</td>
<td>0.020</td>
<td>0.000</td>
<td>0.549</td>
<td>0.129</td>
<td>0.024</td>
<td>0.007</td>
<td>0.014</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV15</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.704</td>
<td>0.178</td>
<td>0.052</td>
<td>0.006</td>
<td>0.006</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV16</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.704</td>
<td>0.178</td>
<td>0.052</td>
<td>0.006</td>
<td>0.006</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV17</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.704</td>
<td>0.178</td>
<td>0.052</td>
<td>0.006</td>
<td>0.006</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>HDOV18</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.704</td>
<td>0.178</td>
<td>0.052</td>
<td>0.006</td>
<td>0.006</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Note: "0.000" in Table 6 indicates that HPMS vehicle type has no matching with the corresponding MOBILE6 vehicle type.

(Source: UTCA, 2003)

5.4.1 VMT by Facility

The HPMS data contains functional classification as shown in Table 4-2. However, this functional classification must be changed into the four facility classes freeway, arterial, local and ramp required by MOBILE6. The EPA suggests that the rural and urban interstates in the HPMS functional classification constitute the freeway facility class in the MOBILE6 system, the arterials and collectors in the HPMS system can be grouped together as to form arterials in the MOBILE6 system and the local roads as local.

Since the HPMS data does not contain ramp information MOBILE6 default values should be used. Then, one of the HPMS vehicle bin codes (Table 5-1) will be assigned to every record based on their vehicle classification. Finally, a file with 14
vehicle classes will be obtained in which each vehicle class is subdivided into 4 facility classes. Under each functional class, 24 VMT fractions for each hour of the day are thus obtained. Now, the 14 HPMS vehicle classes are to be converted into the required MOBILE6 28 vehicle classes using the conversion factors.

5.4.2 VMT by Vehicle Class

This can be done by adding a field called ‘composite vehicle class’ to the original HPMS data file. All the records will then have to be assigned with one of the 16 composite vehicle types shown in Table 2(b). Thus, VMT under each vehicle type is obtained and the required 16 VMT fractions representing each composite vehicle class is computed.

5.4.3 VMT by Hour

The data file obtained from VMT by facility type can be used here in which the data was split by hour of the day. The only difference is that, this command requires the VMT to be classified in hour of the day only. 24 VMT fractions representing each of the hour of the day is required to be computed. EPA (MOBILE6 User’s Guide, 2001) specifies that if the extensive data for VMT by functional class command is unavailable, this command can be used.

5.4.4 VMT by Speed

The HPMS data file, obtained in VMT by facility type can be used here with some modifications. Along with the speed bins, two of the four facility classes (arterials and freeways) are required to classify VMT using this command. Firstly, all the records with arterials and freeways functional classification have to be selected and transferred into another file. The speed values in the HPMS data can then be subdivided into 14 speed bins at 5mph intervals, from 0 to 65 mph. Speed values that fall under 60-65mph or more are assigned with the last speed bin. A separate field called ‘speed bins’ is to be added to the data file and the speed bin numbers corresponding to the speed will have to be assigned for each records. Now, the VMT that comes under arterials can be distributed using the hour of day values. Under each hour of the day, 14 VMT fractions for 14 speed bins will finally be obtained. The same procedure is to be adopted for the freeways.
5.5 Developing the CART System Add-in to TransCAD

5.5.1 Overall Outline

The above sections provide technical information for obtaining VMT estimates using CART and obtaining other HPMS detailed VMT estimates for input to MOBILE6. However, conducting these procedures manually is time consuming. Hence, a procedure was developed as an add-in to TransCAD in which the CART classification system identified earlier is used to (a) estimate system VMT, and (b) estimate individual link VMT. That is, a system was developed in which the user can retrieve information on any particular link, e.g: 1) link volume at a particular hour, speed VMT on that link, 2) link volume by hour, speed, facility type for links outside urbanized area and within the “donut” area. The overall procedure of developing the CART system add-in to TransCAD is explained in terms of a flow diagram in Figure 5-2. Each step is described in the sections that follow.

5.5.2 Creating the HPMS Road Network

Figure 5.3 shows that the first step is to create the highway network for the available HPMS data. The HPMS data contains Linear Referencing System Identification (LRS Id) which is required by the FHWA for linking with the Federal Geographical Information System (GIS). This system allows users to reference HPMS information to the map location of the road section. Based on the available information a network will be created using GIS.

5.5.3 Identifying Individual Links

The second step is to identify the individual links in the network. Note that the HPMS contains information on sampled sections only and the network established in the previous step contains speed limit, lane width information for sampled sections only. So, another highway network layer that contains all the links in the network needs to be added. This is achieved by adding a highway layer to the highway network that contains the required link attributes for the unsampled links.

5.5.4 Incorporating the Derived CART Function

The third step is to incorporate the CART system function for estimating VMT in TransCAD. This involves writing a program in GISDK in Caliper Script to create a CART function in TransCAD. The CART function involves using those variables
previously determined to most effectively distinguish homogeneous traffic volume sections, to estimate the VMT on individual links.

5.5.5 Assigning VMT Estimates on Individual Links

The fourth step is to assign the CART estimated VMT and detailed MOBILE6 VMT estimates on individual links. First, the VMT estimates using the CART will be obtained and assigned for each of the individual links. Then, the national default detailed MOBILE6 VMT input values such as values by VMT by hour, vehicle class, facility type and speed will be assigned to every individual link.

Figure 5.3: Schematic Process to Identify and Assign VMT
5.5.6 Identifying the System Boundaries

The fifth step involves identifying the required systems. The zones and the boundaries that distinguish the urbanized, donut and the total areas will be identified. FHWA classifies donut area as the area outside of the FHWA-approved adjusted Census boundary of one or more urbanized areas but within the boundary of an NAAQS nonattainment area. An FHWA-approved adjusted urbanized area includes the Census urbanized area plus transportation centers, shopping centers, major places of employment, satellite communities, and other major trip generators near the edge of the urbanized area, including those expected to be in place in the near future. The maps corresponding to these boundaries were obtained from the Bureau of Transportation Statistics (BTS) website www.bts.gov.

5.5.7 Total System VMT

The sixth step involves identifying the total system and estimating the VMT for the whole area. The obtained estimates will be assigned further.

5.5.8 Urbanized System VMT

The seventh step involves estimating and assigning VMT for the urbanized area. For all the links which are within the urbanized area, the VMT estimates will be done and will be assigned to the whole system.

5.5.9 Donut System VMT

The eighth step involves estimating VMT and assigning it to the links in the “donut” area. After identifying the area within the non-attainment boundaries but outside the urbanized area, the links in this area will also be identified. Then the VMT estimates for this system will be made and will then be assigned.
6. Building the Procedure on a GIS Platform

6.1 Introduction

Geographic Information Systems (GISs) are a system computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information. In other words, GIS combines layers of spatial information about a place to give us a better understanding of that place. TransCAD is a transportation/GIS software package and is the only package that fully integrates GIS with planning modeling and logistics applications.

TransCAD has a built-in programming language called GISDK (GIS Developer’s Kit), which lets the users customize and extend the capabilities of TransCAD in almost any way they like. GISDK lets the users automate the task they perform on a regular basis, so that one can accomplish it by clicking a single button or choosing a single menu item. GISDK was used to automate the above process to establish the CART system that can estimate the system VMT and individual link VMT. To incorporate the procedure described in the previous chapter into a GIS, the manner in which the input to the process is prepared, and the procedure is executed, must be explained. The following sections deals with how the procedure is implemented on a GIS platform.

6.2 Obtaining and Presenting the Data

6.2.1 HPMS Data

As discussed earlier, the HPMS is a national system for collecting, maintaining, and reporting data detailing, “…the extent, condition, performance, use, and operating characteristics of the Nation’s highways,” (HPMS Field Manual, 2000). In the past, the HPMS data was analyzed and viewed as tables, charts, and graphs. However, with the advent of Geographic Information Systems (GIS), a new spatial dimension was introduced into HPMS analysis. The GIS advances the HPMS’ state-of-the-art and enables the FHWA, the States, and others to analyze HPMS data for rural arterials, urban principal arterials, and other NHS roadways within a spatial context. The ability to integrate data through GIS results in enhanced analysis and presentation of the HPMS data State and Nation wide.
There are a variety of methods used to locate objects in the real world; however, the most commonly used among State Departments of Transportation (DOTs) are Linear Referencing Systems (LRS). Linear referencing is a natural and convenient means to associate attributes or events to locations or portions of a linear feature. The major advantage of linear referencing is its capability of locating attributes and events along a linear feature with only one parameter (usually known as measure) instead of two (such as latitude/longitude or x/y in Cartesian space). Sections of a linear feature can be referenced and created dynamically by indicating the start and end locations along the feature without explicitly storing them.

The FHWA requires each state to use common HPMS Linear Referencing System components in order to mount the LRS on the National Highway Planning Network (NHPN) for GIS processing. The NHPN is the database that contains the geographic or spatial locations of the Nation's principal highways. The NHPN, is a digital database representing the National Highway System and the remaining rural arterials and urban principal arterials. The NHPN was developed based on 1:100,000 Digital Line Graphs (DLGs) from the U.S. Geological Survey and augmented by State-supplied information describing roads and streets not represented in the DLGs. In this context, the NHPN represents highway geometry, and the HPMS is its attributes. The two databases are related to each other via LRS information. The LRS, in effect, gives each database addresses that can be identified in the real world. Through this tie, any location in the NHPN should have a corresponding HPMS record. The two data bases are being brought together through a process called dynamic segmentation, which in many ways resembles a simple database relate. To perform a relate, common fields must exist in all identified databases. In this case, the common fields used to link HPMS to the NHPN are:

- County FIPS
- Inventory Route Number
- Inventory Subroute Number
- Kilometerpoint/Milepoint (KMPT/MPT)

6.3 Creating the Louisiana HPMS Highway Network

For creating the highway network of Louisiana with the HPMS data the dynamic segmentation process should be done. Dynamic segmentation is the process of
transforming linearly referenced data (commonly called events) stored in a table into a feature that can be displayed on a map. In order to use dynamic segmentation, either “measured polyline shapefile” or a route-system in Arc/INFO coverage should be created for the NHPN data.

The NHPN data showing all the interstates, principal arterials, and rural minor arterials is available from the website NHPN (http://www.fhwa.dot.gov/planning/nhpn/). The data was provided in the form of an ArcView Shape file. Each record in the data describes a road section on the network in terms of has a linear referencing system key (LRSKEY), begin mile post (BEGINMP), and end mile post (ENDMP). The LRSKEY field is a 15 digit number which is a concatenation of the two fields LRS ID (a 12 digit linear reference system identification number) and COUNTY FIPS (a 3 digit county code number). The BEGMP field contains the beginning reference point for the route segment in miles. The BEGMP field contains three (3)-implied decimals. The ENDMPT field contains the end reference point for the route segment in miles. The ENDMPT field contains three (3)-implied decimals.

A coverage to the NHPN route-system was created using GIS software Arc/INFO. This software was chosen ahead of other GIS programs because it was relatively easy to create a ‘measured polyline’ network from a ‘polyline’ network by creating coverage in Arc/INFO. The procedure as to how the coverage was created in Arc/INFO is discussed in Appendix B.

The route-system for the Louisiana NHPN network was thus created by creating the coverage. Now, the coverage route-system was loaded into the GIS software ArcView 3.2. In this case, the route system was the first feature class within the coverage folder and was loaded by selecting the folder in the “Add Theme” dialog.

Prior to the loading the Louisiana HPMS data, a field LRSKEY field was added to the HPMS datasheet. FHWA’s definition of the LRSKEY field values in the NHPN data is the field created by combining 2 data items from the HPMS: the 12 character LRS Identification field (HPMS Data Item 10) plus the 3 character County FIPS Code (HPMS Data Item 4)." Then, the HPMS data was loaded as table in the Arcview. However, while importing HPMS ASCII data into an ArcView table, the County FIPS Code field (named "County_cod" in the data) was created as a number type field with a width of 1,
instead of a string type with a width of 3 as suggested by the FHWA. Hence, at first, the data was reselected by querying for just the HPMS records that had a valid LRS_ID and the correct LRSKEY values were created.

Now HPMS data was added as route events to the NHPN network. This adding of route events can be seen in Figure 6-1. This created a new event theme added to the view as shown below in Figure 6-2. The created map was then loaded into TransCAD.

Figure 6-1 Adding Route Events for HPMS Data from the NHPN Data
6.4 Identifying Individual Links

The next step was to identify the individual links in the network. The HPMS data contains information about all the link attributes on only sampled sections and created network contains both universe and sample sections. However, since a few attributes such as speed limit, and lane width are not available in the Universe HPMS data, it is required to identify those links and assign the other link attributes. This was done using the Louisiana highway network data, obtained from the Louisiana Department of Transportation and Development (LADOTD). This data contained information related to speed limit, lane width, shoulder width, etc. Also, supporting databases for the Louisiana State highway roads were obtained from the LADOTD to create a map for the Louisiana highway network data. The network was created as done earlier using the process of dynamic segmentation. This process was done using GIS software ArcGIS 9, as the data was compatible for creating layer (*.lyr) files.

Then, from the folder the Access database named “STLStateRoads” was selected. This Access database contained a table named “Controls”. Using Create Layer command
a road network layer with a supporting route system for the Control Sections data was created.

Now the Louisiana highway network route events were added to the created Control Sections layer as shown in the Figure 6-3. This creates a layer displaying the Louisiana highway network in the view as shown in the Figure 6-5. The created Louisiana highway network was loaded into TransCAD.

The Louisiana HPMS network was the overlaid on the created Louisiana highway network. The unsampled links were then identified and the corresponding link attributes speed limit and lane width were assigned.

Figure 6-3 Adding the Control Sections layer to the View
Figure 6-4 Adding LA Highway Network Route Events to the State Roads Network

Figure 6-5 Map Showing the Added LA Highway Route Events
6.5 Incorporating the Derived CART Function

After deriving the CART classification system function for estimating VMT, a code was written in GISDK, the CART Classification system function was incorporated into TransCAD. This add-in, estimates VMT based on those CART variables that effectively distinguished the homogeneous traffic volume sections on individual links automatically. The details about the created function and GISDK code for add-in are discussed in Chapter 7 and Appendix C.

6.6 Assigning VMT Estimates on Individual Links

6.6.1 CART Classification System Assigned VMT Estimates

After creating the network, identifying the individual links and estimating individual link VMT, the CART VMT estimates and the national default detailed MOBILE6 VMT estimates were assigned on individual links. A code was written in GISDK that can estimate VMT by CART classification system directly, when the user inputs the data that has CART specified link attributes in it.

6.6.2 Assigning Default MOBILE6 VMT Estimates

As discussed earlier in Chapter 5, the HPMS data did not have any information related to vehicle classification, speed and hour of the day, however it had information related to functional class. So with the functional class data, the national default detailed MOBILE6 VMT estimates were assigned to the individual links.

The HPMS data set contains functional classification as shown in Table 4-2. This functional classification is now changed into the required MOBILE6 facility types; the four types (freeway, arterial, local and ramp). The EPA suggests that rural and the urban interstates in the HPMS functional classification constitute the freeways, the arterials and collectors to be grouped as arterials, and the local roads as local. This can be done by assigning one of the four MOBILE6 functional class numbers in the functional class field. However, since the HPMS data does not contain ramp information MOBILE6 default values can be used wherever possible.

First, a field “Facility_Type” was added. The field “Facility_Type” represents the MOBILE6 Facility Type classification classified by the EPA in the MOBILE6 User’s guide. The rural and urban interstates and expressways i.e., “1”, “11”, & “12” functional classification of the HPMS data (Table 4-2) were classified as interstates, and were
numbered “1” in the “Facility_Type” field. Similarly, as specified by the EPA(2001), the urban and rural arterials and collectors i.e., “2”, “6”, “7”, “14”, “16”, & “17” were grouped and classified as arterials, and were numbered “2” in the “Facility_Type” field. This classification is done as shown in Table 6-1.

<table>
<thead>
<tr>
<th>HPMS Functional Classification</th>
<th>MOBILE6 Facility Type Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 11, 12</td>
<td>1</td>
</tr>
<tr>
<td>2, 6, 7, 14, 16, 17</td>
<td>2</td>
</tr>
</tbody>
</table>

6.6.2.1 VMT by Facility

Firstly, fields were added to the dataview based on facility class, hour of the day and vehicle classification. All the records were assigned default MOBILE6 VMT values by the facility class, by 24 hours of the day, and by 28 vehicle classification. For example for a record with facility class “1”, for a vehicle class “LDGV” the following fields were created: “VMTbyFC/LDGV/Hr1”, “VMTbyFC/LDGV/Hr2”, “VMTbyFC/LDGV/Hr3”, and so on by 24 hours of the day for all the 28 vehicle classes as shown in (Table 3(a)).

6.6.2.2 VMT by Vehicle Class

The fields were similarly added in the dataview as described earlier but by only 16 composite vehicle types (Table 3(b)). Every record was assigned by MOBILE6 VMT default values with the 16 vehicle types such as “VMT by LDV”, “VMT by LDT1”, “VMT by LDT2”, and so on.

6.6.2.3 VMT by Hour

The fields were added by VMT by hour for 24 hours of the day such as “VMT by Hr1”, “VMT by Hr2”, etc. and the default MOBILE6 values were assigned for all records.

6.6.2.4 VMT by Speed

The HPMS data file, obtained in VMT by facility class can be used here with some modifications. Along with the speed bins, two of the four facility classes (arterials and freeways) are required to classify VMT using this command. After assigning the “Speed Bin” field and the corresponding speed bin numbers for all the records, the VMT
that comes under for facility classes “arterials” and “interstates” were distributed using the hour of day values. Under each hour of the day, 24 MOBILE6 default VMT fractions for the corresponding speed bin were assigned. For example, if for a particular record the facility type was “1” and the corresponding speed bin was “12”, then under each hour of the day 24 MOBILE6 default VMT fractions were assigned under the fields “SpeedVMT/Hr1”, “SpeedVMT/Hr2”, “SpeedVMT/Hr3” and so on.

6.7 Identifying the System Boundaries

This step involved identifying the required systems and their boundaries. The systems were classified as total System, urbanized system and the donut system.

The total zones and boundaries that distinguish the urbanized, donut and the total areas were obtained.

6.8 Total System VMT

After obtaining the boundaries for the three systems, a code was written in GISDK such that if the user clicks a button for total system VMT, the program estimates the total system VMT and displays the value.

6.9 Urbanized System VMT

This step involved estimating and assigning VMT for the urbanized area. For all the links which are within the urbanized area, the VMT estimates were done and a code was written in GISDK, such that when the user prompts for urban VMT, the code estimates VMT for all the links in the urbanized area and displays the value.

6.10 Donut System VMT

This step involved estimating VMT and assigning it to the links in the “donut” area. After identifying the area within the non-attainment boundaries but outside the adjusted-census urbanized area, the links in this area were identified. Then, a code was written that identifies links in the donut area and returns the estimated donut VMT.

6.11 Program Execution in GIS

6.11.1 Execution Platform

The Geographic Information System Developer’s Kit (GISDK) provides a tool kit that can be used to customize and extend the capabilities of TransCAD in any desired way. GISDK is a collection of software tools that comes with TransCAD which makes automation of repetitive tasks in TransCAD possible. It also helps in creating user-
designed add-ins, integrating other programs, building custom applications or access TransCAD as an automated server to add maps or transportation analysis function to the user created programs. GISDK also makes it possible to call TransCAD functionality from other programming environments, such as Visual Basic, to provide transportation analysis or mapping services to an application, such as finding a route or displaying a map around a location (GISDK programmer’s Guide for TransCAD, 2001).

GISDK uses a compiled programming language called Caliper Script which is the primary constituent of GISDK. This is an easy-to-learn programming language that provides a way to interact with the TransCAD program and data. The code written in other languages such as C, FORTRAN, Visual Basic or any other language including another application’s macro language (e.g. Excel) and can be mixed with GISDK programs written in Caliper Script, allowing for compatibility with existing software (GISDK programmer’s Guide for TransCAD, 2001). There are over one thousand GISDK functions in TransCAD, all of which can be called using Caliper Script. These functions help in managing and analyzing maps efficiently.

Caliper Script has three components: a compiler, a debugger and a toolbox for interacting with the compiler and debugger. The GISDK compiler takes the caliper script code and creates a User Interface (UI) database that can be run in the TransCAD platform. The compiler checks and reports various types of errors in the code including errors in syntax, invalid and mismatched statements and creates an error file listing each error and its location. The GISDK debugger runs the Caliper Script code in testing mode in order to make sure that there are no errors, such as errors in logic or variable handling. If the code does not execute properly, the debugger allows the user to set breakpoints or to step through the code one line at a time, to make sure that the code is executing as expected. The GISDK Toolbox not only has buttons to run the compiler and the debugger but also has tools that makes interacting with the program faster and easier (GISDK programmer’s Guide for TransCAD, 2001).

The hardware requirements for GISDK are minimal, if TransCAD can be run, then GISDK can be run. GISDK does not have its own editor. Therefore, the GISDK source code can be created, edited and displayed in any text editor such as notepad. The source code files are known as resource files and always have an .rsc extension GISDK
programmer’s Guide for TransCAD, 2001). The editor used for writing the code for the CART classification system add-in is ConTEXT.

6.11.2 Implementation of the Program

The HPMS network file along with the zone and system boundaries are added as layers and a map is created to further proceed with the implementation of the methodology identified in the previous chapter.

In the first step, a dialog box was created, which asks whether the user would like to start the CART classification system. GISDK has a function Dbox, which is used for creating a dialog box that can be used with the Macro to get the desired display. The user can select the map or file on which the user wants to run the CART Classification Scheme by clicking the button “Yes”.

In the second step, CART classification system equation was created to estimate the VMT on individual links. GISDK has functions called GetFirstRecord() and GetNextRecord(), which gets the first record and the next corresponding records. This step of the code checks each and every record for the CART classification specified link attributes and if the condition is satisfied it returns the AADT value and displays it in the field for every record. This process is repeated until all the records are checked for the condition and the value is displayed. This code also checks for all the CART classification specified attributes and returns the total DVMT value in a particular area. The entire GISDK code is shown in Appendix C.

In the third step, a tool box was created for the HPMS network layer. GISDK has functions called Tool and Button, which creates a toolbox as a dialog box. A toolbox appears on the map, until the user closes it. This toolbox named “Info” is used to identify the VMT distribution i.e., VMT on individual links as well as system VMT. GISDK has a function called ClickCoord(), which returns the coordinate of the clicked point and gets the link id of the clicked link. This is used for identifying the link the user clicks.

In the fourth step, “VMT Distribution” dialog box was created. When the user clicks the “Info” button on the tool box and clicks on any link, a dialog box appears prompting the user, whether the user wants to see the “Individual Default MOBILE6 VMT values” or “System VMT”. If the user expects to see “Individual Default MOBILE6 VMT values”, a dialog box appears asking the user to choose among the four
options: “VMT by Hour”, “VMT by FC”, “VMT Mix”, and “VMT by Speed”. When the user clicks any of these three options “VMT by Hour”, “VMT Mix”, and “VMT by Speed”, a scroll box appears displaying the default MOBILE6 VMT values of the corresponding option. However, when the user clicks the “VMT by FC” options, another dialog box prompts asking the user to select one of the 28 vehicle types, and when the user clicks the vehicle type, it displays the default MOBILE6 VMT by facility class values by 24 hours of the day for that particular vehicle type.

GISDK has functions *LocateNearestRecord()*, *LocateRecord()*, *GetRecordValues()*, and *ShowArray()*, which when the user clicks on a particular link, it finds the coordinate, locates the nearest record to that clicked point, and then locates the id of that particular link/record. Then, the function *GetRecordValues()* gets the record values of that particular link and the *ShowArray()* displays the retrieves the user specified attributes of that particular link.

If the user wants to see the “System VMT”, a dialog box appears asking the user to select one of the three options: “Urban”, “Donut”, and “Total”. If the user selects “Urban”, the code calculates DVMT value for all the links in the urbanized area, sums up the value and displays it as a message. If the user selects “Donut”, the code checks whether the road link is within the nonattainment area but outside the adjusted-census urbanized boundary, then calculates DVMT value for all the links in the donut area, sums up the value and displays it as a message. If the user selects “Total”, the code calculates DVMT value for all the links in the entire area, sums up the value and displays it as a message.
7. Analysis and Results

7.1 CART Classification System Equation

7.1.1 Evaluating the CART Classification Tree

As discussed earlier, the CART analysis was done on Louisiana standard sample HPMS data 2003 using the software SPSS Answer Tree 1.0. Using the CART algorithm, a tree was grown to identify the variables that effectively distinguish the homogenous traffic volume sections. The automatically grown tree was then analyzed by examining the risk estimate summary tree and finding the proportion of variance to check how well the tree does at identifying those variables that effectively distinguish the homogeneous traffic volume sections. The risk estimate here is simply the within-node variance and the total variance equals the within-node (error) variance plus the between-node (explained) variance. The within-node variance in this case was 9.03796e+007, while the total variance was 4.54353e+008 (the risk estimate for the tree with only one node). The proportion of variance due to error was (9.03796e+007/4.54353e+008) = 0.1989. Thus, the proportion of variance explained by the model was found to be 100% - 19.89% = 80.11%. This shows that there was still some residual variance, but the amount that can be accounted for using the model was enough to convince that the most important variables were captured.

7.1.2 Deriving the CART Classification System Equation

After evaluating the tree, the variable splits were examined to identify the variables that most effectively distinguish the homogeneous traffic volume sections. By examining the variable splits carefully, it was found that the following variables can effectively distinguish the homogeneous traffic volume sections. They are: rural/urban designation, functional system, type of facility, speed limit, number of through lanes, and lane width. A discrete function was derived with the above variables for estimating VMT on unsampled links. The derived equation was:

\[
DVMT = \sum_{i} \sum_{j} \sum_{k} \sum_{l} \sum_{m} \sum_{n} AADT_{ijklmn} \times TSI_{ijklmn}
\]

where, \(DVMT\) = Total VMT estimate

\(i\) = rural/urban designation

\(j\) = functional class

52
\( k = \) type of facility  
\( l = \) number of through lanes  
\( m = \) lane width  
\( n = \) speed limit  
\( AADT_{ijklmn} = \) AADT for sample section ‘o’ with link attributes \( i,j,k,l,m,n \)  
\( TSL_{ijklmn} = \) total section length for all the links with attributes \( i,j,k,l,m,n \)

7.2 VMT Estimation

The main objective of this research was to compare the VMT estimates obtained by CART classification system with that obtained by the HPMS conventional method. This comparison was done for the East Baton Rouge parish. A highway network was created for the East Baton Rouge parish using TransCAD.

7.2.1 East Baton Rouge Highway Network

The Louisiana highway network containing all the link attribute information for the universe and sample sections was opened. Using, Select by Condition command, all the links in the East Baton Rouge (EBR) Parish were selected and then all the records in the selection set were exported as a Standard Geographic file. Thus, the highway network for the EBR parish was created. This highway network file contained missing speed limit and lane width information on few links. These were links were the missing information could be filled by using Inference. This created a network shown in Figure 7-1, which had all the information for estimating VMT using CART classification system.

7.2.2 VMT Estimation by CART Classification System and HPMS Conventional Method

HPMS conventional method and CART classification method estimates are estimates that are made statewide and based on the type of area i.e., rural, small urban, urban or large urban area. Therefore, the values compared should be the statewide estimates and not just parish wide. However, the statewide estimates for each classification can be used and assigned to the same classification in a parish. That means, for example if statewide average AADT value for a CART classification of rural/urban ‘1’, functional system ‘1’, type of facility ‘2’, number of through lanes ‘4’, lanewidth ‘12’ and speed limit ‘55’ is 25500. Then, the parish wide average AADT value for the same CART classification must be assigned the same.
Figure 7-1 Created EBR Highway Network

Similarly, for HPMS conventional method, for example, if the statewide average AADT value for a rural area of volume group ‘1’, and functional classification ‘2’ is 38900, then the parish wide average AADT value for the same classification should be assigned the same.

Thus, two fields “AADT_CART” and “HPMS CO_MTHD” were added to the dataview of the EBR HPMS highway network layer. The two fields were calibrated using the procedure stated above, and the AADT values were assigned to all the known and the unknown links.
Figure 7-2 Dataview Showing the Observed and Estimated AADT

Figure 7-2 shows the dataview with AADT estimates by CART classification method and HPMS conventional method. The AADT on each link was then multiplied by the section length of that link to obtain the VMT estimates on each link.

7.3 Testing the Resulted AADT Estimates

The obtained AADT estimates were then tested by comparing the AADT estimates obtained through CART classification system and HPMS conventional method with the observed AADT i.e., with the AADT on sampled sections. For this paired sample T-test was done for testing the statistical similarity of the following and then by observing their root mean square error (RMSE).

- Observed AADT and AADT estimates obtained through CART classification system
- Observed AADT and AADT estimates obtained through HPMS conventional method
- AADT estimates obtained through CART and HPMS conventional method

The paired sample T-test compares the means of two variables. It computes the difference between the two variables for each case and tests to see if the average
difference is significantly different from zero. Here, the null hypothesis that there is no significant difference between the two variables was tested against the alternative hypothesis that there is significant difference between the two variables. The null hypothesis is rejected if the calculated T-value is greater than the critical T-value. The paired sample T-test was done using statistical software SPSS 12.0.

7.3.1 Comparison of Observed AADT with the AADT Estimates Obtained Through CART Classification System

The SPSS output for the paired sample T-test for the observed AADT and the estimated AADT obtained through CART classification system is shown in the Figure 7-3.

The obtained T-value as seen from Figure 7-3 was 1.045 and the critical T-value 2.009. Thus, the null hypothesis cannot be rejected indicating that there is no statistical difference between the two samples. Also, as seen from the Standard error of the mean i.e., the RMSE for the estimates made through CART classification system
7.3.2 Comparison of Observed AADT with the AADT Estimates Obtained Through HPMS Conventional Method

The SPSS output for the paired sample T-test for the observed AADT and the estimated AADT obtained through HPMS conventional method is shown in the Figure 7-4. The obtained T-value as seen from Figure 7-4 was 1.069 and the critical T-value 2.009. Thus, the null hypothesis cannot be rejected indicating that there is no statistical difference between the two samples.

![Paired Sample T-Test for Observed AADT and Estimated AADT by HPMS Conventional Method](image)

Figure 7-4 T-Test Result Output for the Observed AADT and HPMS Method Estimated AADT

7.3.3 Comparison of AADT Estimates Obtained Through CART Classification System and HPMS Conventional Method

The SPSS output for the paired sample T-test for the AADT estimates obtained through CART classification system and HPMS conventional method is shown in the Figure 7-5 below:
### Paired Sample T-test for AADT estimates from CART Classification and HPMS Conventional Methods

#### Paired Samples Statistics

<table>
<thead>
<tr>
<th>Pair</th>
<th>AADT_CART</th>
<th>HPMSCO_MTD</th>
<th>Mean</th>
<th>N</th>
<th>Std. Deviation</th>
<th>Std. Error Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>42796.54</td>
<td>44213.75</td>
<td>52</td>
<td></td>
<td>33417.69660</td>
<td>4634.201</td>
</tr>
<tr>
<td></td>
<td>33417.69660</td>
<td>37352.73141</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Paired Samples Correlations

<table>
<thead>
<tr>
<th>Pair</th>
<th>AADT_CART &amp; HPMSCO_MTD</th>
<th>N</th>
<th>Correlation</th>
<th>Sig.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>52</td>
<td>.922</td>
<td>.000</td>
</tr>
</tbody>
</table>

#### Paired Samples Test

<table>
<thead>
<tr>
<th>Pair</th>
<th>AADT_CART &amp; HPMSCO_MTD</th>
<th>Paired Differences</th>
<th>95% Confidence Interval of the Difference</th>
<th>t</th>
<th>df</th>
<th>Sig. (2-tailed)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Mean</td>
<td>Std. Error Mean</td>
<td>Lower</td>
<td>Upper</td>
<td>-0.705</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1417.21</td>
<td>4487.80447</td>
<td>-5450.64</td>
<td>6166.219</td>
<td>.484</td>
</tr>
</tbody>
</table>

---

**Figure 7-5 T-Test Result Output for the AADT Estimates Obtained Through HPMS Method and CART Classification Method**

The obtained T-value as seen from Figure 7-5 was -0.705 and the critical T-value was 2.009. Thus, the null hypothesis cannot be rejected indicating that there is no statistical difference between the two samples.

**7.4 Comparison of RMSE for the HPMS and CART Estimated AADT Values**

The RMSE of the HPMS conventional method and CART classification system method were calculated using the observed and estimated AADT values. The RMSE for the HPMS conventional method obtained was 547.761 and the RMSE of the CART classification system was 1918.223. From the RMSE values it can be inferred that the HPMS conventional method provides better estimates than the CART classification system.
7.5 Graphical Comparison of the Observed and the Estimated AADT Estimates

Figure 7-6 shows the average percentage difference of the observed estimated AADT values through HPMS conventional method and CART classification system. The figure shows that the AADT estimates obtained by HPMS conventional method are better than the values estimated by CART classification system.

Figure 7-6 Average Percentage Difference between Observed and Estimated AADT
8. Demonstration of the CART Classification System Add-in for the East Baton Rouge Non-attainment Parish

8.1 Background

As required by Title I of the Clean Air Act Amendments of 1990, in 1991 EPA designated as "non-attainment" ninety-eight areas of the United States that did not meet the National Ambient Air Quality Standard (NAAQS) for ground-level ozone (a primary constituent of smog). This ozone health standard was based on a 1-hour average concentration of 124 parts per billion. Non-attainment areas are classified as marginal, moderate, serious, severe, or extreme, based on air quality monitoring data.

In Louisiana, the following parishes are classified under severe ozone non-attainment areas, as of the year 2003:

1. Ascension Parish
2. East Baton Rouge Parish
3. Iberville Parish
4. Livingston Parish
5. West Baton Rouge Parish

The CART classification system add-in can be used for any region, county, state, or for the entire country which has CART specified variables. For the demonstration purposes East Baton Rouge parish was selected.

8.2 Data

The data used in this demonstration was obtained from a variety of sources. The Louisiana HPMS data was obtained from the FHWA and the Louisiana highway network from the LADOTD. The EBR highway network containing the CART classification system specified variables for the universe and sample sections, created as explained in section 7.2.1 was used for the demonstration purposes. A table containing the VMT default MOBILE6 values was created and joined to the EBR highway network dataview. The federally adjusted census urbanized boundary of EBR parish was obtained from the BTS website www.bts.gov. The parish boundary data was obtained from the ArcView software package. This data was also available with other software such as ArcGIS and TransCAD.
8.3 Results

The CART classification system add-in, which is a GISDK program, was run on the EBR highway network map. This is a user interactive code, and at every step asks the user what the user wants to see. When the program is run, the code first asks the user whether to start the CART classification scheme.

When the user clicks on “Yes” the code prompts the user to select the map on which the user wants to run the add-in. In this case the file name was “EBRHPMS Hwy.map”. As soon as the user selects the map, the CART classification system add-in is run and based on the user specified DVMT values to be displayed as a message, the code displays the value. In this case, it was specified to display the total DVMT for the Baton Rouge rural area (a rural area as defined by FHWA is an area which has population less than 5,000). The code calculated the value and displayed it as a message as shown in Figure 8-1.

Figure 8-1 CART Classification System Add-in Generated DVMT Estimate
After displaying the VMT through CART classification equation, in the next step an “Info” tool box was be displayed as shown in the Figure 8-3. This toolbox can be used by the user to display the system and individual link VMT.

The “Info” button on the toolbox was clicked and then using the arrow, a link was selected by click on the map. When the link is selected, the code prompts whether the user wants to see the VMT distribution as shown in the Figure 8-4. If the user clicks “Yes”, then a dialog box appears to Select Option among “MOBILE6 Default Values” or System VMT as shown in Figure 8-5.
Figure 8-3 CART Classification System Add-in Created VMT Distribution

Figure 8-4 Selecting VMT Distribution Options
If the user selects “MOBILE6 Default Values”, a dialog box appears to “Choose VMT Input” i.e., “VMT by Hour”, “VMT by FC”, “VMT by Mix”, or “VMT by Speed” as shown in Figure 8-6.

![Figure 8-5 Choosing VMT Input Options](image)

Bases on the VMT input selected, the code displays the MOBILE6 default VMT values of the specified input. For example, if “VMT by Hour” option is selected, MOBILE6 default VMT values by 24 hours of the day will be displayed. If “VMT by FC” is selected, a dialog box appears prompting the user to select the vehicle type for which the user wants to retrieve information shown in Figure 8-7. When the vehicle type is selected, MOBILE6 default values for the specified vehicle type for 24 hours of the day will be displayed. If the user selects “VMT by Mix”, MOBILE6 default VMT values
for the 16 composite vehicle types (Table 2(b)) will be displayed. If “VMT by Speed” input is selected, then MOBILE6 default VMT values for that particular speed bin and facility type for 24 hours of the day will be displayed.

However, if the user selects the option “System VMT”, then a dialog box appears prompting the user to select the system, i.e., “Urban”, “Donut”, or for the “Total” system as shown in the Figure 8-8. If the user selects the system “Urban” the code estimates VMT for all the links in the urbanized area, sums up all the value and displays the DVMT value. If the user selects the system “Donut”, the code identifies the area within the non-attainment boundaries but outside the adjusted-census urbanized area. Then, the code identifies the links in this area and returns the estimated donut DVMT. If the user selects

Figure 8-6 Selecting Vehicle Type

However, if the user selects the option “System VMT”, then a dialog box appears prompting the user to select the system, i.e., “Urban”, “Donut”, or for the “Total” system as shown in the Figure 8-8. If the user selects the system “Urban” the code estimates VMT for all the links in the urbanized area, sums up all the value and displays the DVMT value. If the user selects the system “Donut”, the code identifies the area within the non-attainment boundaries but outside the adjusted-census urbanized area. Then, the code identifies the links in this area and returns the estimated donut DVMT. If the user selects
the system “Total”, the code estimates VMT for all the links sums up the value and displays it.

Figure 8-7 Selecting System VMT
9. Summary and Conclusions

9.1 Study Summary

The study presented a methodology to identify the homogenous traffic volume sections based on link attributes. The approach focused on using the CART classification system for identifying those links attributes on the Louisiana standard sample HPMS data. An equation was derived based on the links that effectively distinguished the traffic volume sections. A line layer was created for the Louisiana HPMS data using the process of dynamic segmentation with the NHPN data. The Louisiana highway network data was obtained and a line layer was created. Then, the Louisiana highway network layer was overlaid on the Louisiana HPMS network layer, thus adding the speed limit and lane width attributes to the HPMS universe data. Links with missing information were tabulated using Inference. Then the CART classification system estimated VMT were assigned on individual links. MOBILE6 default VMT values were then assigned based on speed bin and facility type. The federally adjusted census boundaries were obtained and systems were identified. Then, the EBR highway network was created. The whole procedure was automated in by creating customized add-in in TransCAD using GISDK. This add-in uses the CART system to (a) estimate system VMT, and (b) estimate individual link default MOBILE6 VMT. For instance, if the user can retrieve information on any particular link, e.g: 1) link volume at a particular hour, speed VMT on that link, 2) link volume by hour, speed, facility type for links outside urbanized area and within the “donut” area.

To check how well the CART classification system analysis estimates VMT than the HPMS conventional method, paired sample statistical T-test was done on the AADT values observed and obtained on the sample sections of EBR parish. This was done to compare the observed AADT values with the estimates obtained from HPMS conventional method and the CART classification system. Based on the paired sample test, we were unable to reject the null hypotheses indicating that observed AADT values and the AADT values estimated through CART classification system and HPMS conventional method were same at 5% level of significance for the three samples.
9.2 Conclusions

The main objective of the study was to develop a comprehensive classification system to estimate VMT as an alternative to the HPMS conventional method. The specific objectives of the study were to test both the methods stated above with the observed values on the sample sections, automate the CART classification system in TransCAD by creating a customized add-in using GISDK and then demonstrate the procedure for the EBR parish.

Using the CART algorithm in SPSS Answer Tree 1.0, the classification variables that effectively distinguish the homogeneous traffic volume sections were identified. Then, using the CART classification variables an equation was derived to estimate VMT. This procedure was automated in TransCAD which facilitates certain data operations and allows a convenient means of graphically displaying the results. The process provides an automated means of identifying the CART variables and estimates the individual and system VMT.

The methodology developed can be applied to any area which has CART specified variables. The developed add-in can be used on any other traffic count data other than the HPMS traffic count data that has the CART classified variables.

Graphical and statistical comparisons were made for the observed and the estimated AADT values by CART classification system and HPMS conventional method. Based on the analyses and results, the following conclusions drawn from the present study:

- Based on the three statistical comparisons of the observed and the estimated AADT values through CART classification system and HPMS conventional method, the values obtained by both the methods were similar at 5% level of significance.
- Statistical comparisons indicated that the CART classification method performs better in estimating AADT than the HPMS conventional method.
- However, the graphical and RMSE results indicated that the HPMS conventional method performs better than the CART classification system.
- The results can possibly be biased. This is because the analysis is based on tests conducted on sample sections where the observed HPMS traffic count
values were present. Since HPMS conventional method classification is based on volume group classification, the test proved that the HPMS conventional method to be better. Hence, we do not know which method performs better on estimated sections i.e., on the universe sections. However, if the actual traffic count data for the East Baton Rouge parish was available, then it would have been appropriate to compare the actual traffic count data with the estimated values by CART classification system method and HPMS conventional method on the universe sections. Since this data was not available the comparison could not be made. Therefore, more investigation is required.

9.3 Further Research

- The methodology developed in the present study was not statistically tested for the universe sections. If traffic count data could be obtained on the universe sections, then it would be appropriate to test the HPMS conventional method and CART classification system as to which method performs better in estimation.

- The HPMS does not contain much information on local roads. Also, it does not contain vehicle classification, hour of the day and speed information on individual links. LADOTD is currently planning a three year extensive traffic count program in about taking 55,000 traffic counts. This is in order to provide VMT estimates by urban area, parish, functional system, or by any other break down that can be identified via public road inventory database.

- The customized CART classification add-in that was created during this study provides a framework that can be followed and shared with other researchers in the future.

- The present study uses comprehensive classification developed by CART which does not use other link attributes such as land use, shoulder width, HOV lanes etc that may be important factors in another region. Estimating VMT based on better classification may enhance the application of the methodology.
References


Date of reference: 09-03-2003.


Date of reference: 09-03-2003.


Erlbaum, N.S. *Estimated County Level Vehicle Miles of Travel*. Data Services Bureau Report Planning Division, New York State Department of Transportation, Albany, April 1989.


Date of reference: 09-15-2003

UTCA: Williamson, G. D., and Yao, M. “Improved Data Collection and Evaluation for MOBILE6 Based Conformity Analysis of Transportation Projects for the Alabama Department of Transportation”, University Transportation Center for Alabama (UTCA), February 2003.

# Appendix A: HPMS Full List of Data Items

<table>
<thead>
<tr>
<th></th>
<th>Data Year</th>
<th>17 Functional System</th>
<th>33 AADT (PA,NHS,Samp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>State Code</td>
<td>18 Func Sys Code (generated)</td>
<td>34 No of Through Lanes</td>
</tr>
<tr>
<td>3</td>
<td>English/Metric Units</td>
<td>19 NHS</td>
<td>35 IRI (Roughness Index)</td>
</tr>
<tr>
<td>4</td>
<td>Co FIPS Code</td>
<td>20 Planned / Unbuilt</td>
<td>36 PSR (Condition)</td>
</tr>
<tr>
<td>5</td>
<td>Section ID</td>
<td>21 Interstate Route No</td>
<td>37 HOV OPS</td>
</tr>
<tr>
<td>6</td>
<td>Standard Sample</td>
<td>22 Route Signing</td>
<td><strong>HIGHWAY SURVEILLANCE</strong></td>
</tr>
<tr>
<td>7</td>
<td>Donut Sample</td>
<td>23 Rte Sign Qualifier</td>
<td>38 Hwy Surv - Electronic</td>
</tr>
<tr>
<td>8</td>
<td>State Control</td>
<td>24 Signed Route No</td>
<td>39 Hwy Surv - Metered ramp</td>
</tr>
<tr>
<td>9</td>
<td>Grouped Section</td>
<td>25 Government Ownership</td>
<td>40 Hwy Surv - Var message sign</td>
</tr>
<tr>
<td>10</td>
<td>LRS Section ID</td>
<td>26 Spec Sys (STRAHNET)</td>
<td>41 Hwy Surv - Hwy advisory radio</td>
</tr>
<tr>
<td>11</td>
<td>LRS Begin Point</td>
<td>27 Type of Facility</td>
<td>42 Hwy Surv - Surveillance cameras</td>
</tr>
<tr>
<td>12</td>
<td>LRS Ending Point</td>
<td>28 Designated Truck Rte</td>
<td>43 Hwy Surv - Incident detection</td>
</tr>
<tr>
<td>13</td>
<td>Rural/Urban Desig</td>
<td>29 Toll facility</td>
<td>44 Hwy Surv - Free cell phone</td>
</tr>
<tr>
<td>14</td>
<td>Sampling Technique</td>
<td>30 Section Length</td>
<td>45 Hwy Surv - Service patrol</td>
</tr>
<tr>
<td>15</td>
<td>Urbanized Area Code</td>
<td>31 Donut Volume Group</td>
<td>46 Hwy Surv - In-vehicle sign</td>
</tr>
<tr>
<td>16</td>
<td>NAAQS Code</td>
<td>32 Sample Volume Group</td>
<td></td>
</tr>
</tbody>
</table>

| 47 | Sample ID | 65 Length of Curves-Class C | 81 %Single Trucks - Peak Travel |
| 48 | Donut Expansion Factor | 66 Length of Curves-Class D | 82 % Single Trucks - Average Travel |
| 49 | Sample Expansion Factor | 67 Length of Curves-Class E | 83 %Combo Trucks - Peak Travel |
| 50 | Surface Pavement Type | 68 Length of Curves-Class F | 84 %Combo Trucks - Average Travel |
| 51 | SN or D | 69 Horiz Alignment Adequacy | 85 K Factor (Design Hour Volume) |
| 52 | Climate Zone | 70 Terrain Type (rural) | 86 D Factor (Directional Factor) |
| 53 | Yr. of Surface Improvement | 71 Vert Alignment Adequacy | 87 No of Lanes in Peak Hour Dir |
| 54 | Lane Width | 72 Length of Grades-Class A | 88 Turning Lanes / Bays (left) |
| 55 | Access Control | 73 Length of Grades-Class B | 89 Turning Lanes / Bays (right) |
| 56 | Median Type | 74 Length of Grades-Class C | 90 Type of Signals |
| 57 | Median Width | 75 Length of Grades-Class D | 91 % Green Time (signals) |
| 58 | Shoulder Type | 76 Length of Grades-Class E | 92 No of At-Grade Signals |
| 59 | Shoulder Width (RT) | 77 Length of Grades-Class F | 93 No of At-Grade Stops |
| 60 | Shoulder Width (LT) | 78 Passing St Distance (rural) | 94 No of At-Grade Others |
| 61 | Peak Parking (urban) | 79 Weighted Design Speed | 95 Peak Capacity |
| 62 | Widening Feasibility | 80 Posted Speed Limit | 96 Volume / Service Flow Ratio |
| 63 | Length of Curves-Class A | 97 Future AADT |
| 64 | Length of Curves-Class B | 98 Future AADT Year |

72
Appendix B: Creating a Coverage in Arc/INFO

Firstly, the Louisiana NHPN shapefiles were stored in the Arc/INFO Workspace folder. The Arc/INFO Workstation was then opened. Then using the following procedure coverage to the NHPN route system was created.

Procedure
1. Opening the workspace

On opening the Arc/INFO Workstation, at the command prompt Arc, type

Arc: workspace
Current location: C:\

Arc: workspace NHPN
Current location: C:\Workspace\NHPN

This creates a workspace for the Louisiana NHPN network.

2. Set the coverage coordinate precision to DOUBLE DOUBLE

Usage: PRECISION <SINGLE | DOUBLE>

{HIGHEST | LOWEST | SINGLE | DOUBLE}

Arc: precision double double

The precision is to be set to double double otherwise the coverage coordinates will default to single precision. Shape files store feature coordinates as double-precision floating point numbers. Hence, double-precision coverage should be created from shapefile so as to maintain maximum detail and coordinate accuracy.

3. Use the SHAPEARC command to create a coverage from the shapefile

Usage: SHAPEARC <in_shape_file> <out_cover> {out_subclass}

{DEFAULT | DEFINE}

Arc: shapearc input_nhpn.shp nhpn_cov

Here, “input_nhpn.shp” is the name of the input shapefile and “nhpn_cov” is the coverage that will be created.

4. Adding a route-system using ARCSECTION

Arc: arcsection nhpn_cov nhpn lrskey # begmp endmp
Appendix C: GISDK Program

Macro "INTERFACE"
  Answer = RunDbox("Confirm starting CART")
  If Answer = "Yes" then do
    fnm = ChooseFile({{"Map File", "*.map"}}, "Open a Map",)

    on error do
      ShowMessage("Unable to open map file" + fnm + ").")
      Return()
    end
    map = OpenMap(fnm,
    on error default
    end
    RunMacro("Script", fnm)
    tool = RunDbox("ToolBox")
  endMacro

Macro "Script" (fnm)
vwl = GetviewLayer("EBRHPMS Hwy ntw")
SetView(vwl)
TSL = 61027.35
Total = 0
A5 = 0
A4 = 0
A3 = 0
A2 = 0
A1 = 0
DVMT = 0
TSL = 61027.35
for i = 1 to 4 do
  for j = 1 to 17 do
    for k = 1 to 4 do
      for l = 1 to 9 do
        for m = 8 to 18 do
          for n = 25 to 70 step 5 do
            rec = GetFirstRecord("EBRHPMS Hwy ntw|
            while rec <> null do
              samp = vw1.IS_SAMPLE
              ru = vw1.RURAL_URBA
              fs = vw1.F_SYSTEM
              tf = vw1.TYPE_FACIL
              th = vw1.THRUH_CHANNEL
              lw = vw1.LANE_WIDTH
              sl = vw1.SPEED_LIMI
              aadt = vw1.AADT
              TSL = 61027.35
if samp = 1 & ru = i & fs = j & tf = k & th = l & lw = m & sl = n then do
    DVMT = DVMT + aadt * TSL
end
rec = GetNextRecord("EBRHPMS Hwy ntw|",null,null)
end
ShowMessage("DVMT is " + String(DVMT))
A1 = A1 + DVMT
end
ShowMessage("DVMT is " + String(A1))
A2 = A2 + A1
end
ShowMessage("DVMT is " + String(A2))
A3 = A3 + A2
end
ShowMessage("DVMT is " + String(A3))
A4 = A4 + A3
end
ShowMessage("DVMT is " + String(A4))
A5 = A5 + A4
end
if i = 1 then
    ShowMessage("DVMT for Rural area (pop < 5,000) is " +
    Format(A5,".*0.00"))
if i = 4 then
    ShowMessage("DVMT for Large Urban area (pop > 200,000) is " +
    Format(A5,".*0.00"))
A6 = A6 + A5
end
ShowMessage("Total DVMT estimated by CART CS is " + Format(A6,".*0.00"))
endMacro
Dbox "Confirm starting CART"
    init do
        str = "Do you want to start the CART Classification Scheme?"
        enditem
text 1, 1, 45, 1
        variable: str
        align: center
        button "Yes" 2, 4.5, 9
            do Return("Yes") enditem
        button "No" 45.5, 4.5, 9
            do Return("No") enditem
endDbox
Dbox "ToolBox" right,center title: "Info" ToolBox
    NoKeyboard
    Init do
        highway = "EBRHPMS Hwy ntw"
    end
endItem
Close do
    if aid <> null then DropAnnotation(, aid)
        return()
endItem
Tool"Info" 4,2 Cursor:"Arrow"
do
    point = ClickCoord()
    //ShowMessage(RealToString(point.lon) + RealToString(point.lat))
    choice = RunDbox("VMT Distribution")
    if choice = "Yes" then
        options = RunDbox("Options")
        on error do
            Return()
        end
    if options = "MOBILE6 Default Values" then do
        vmt_inputs = RunDbox("Choose VMT Inputs")
        if vmt_inputs = "VMT by Hour" then do
            rh = LocateNearestRecord(point, 15.0)
            //ShowMessage("The nearest id is"+ String(highway.ID))
            rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID",{highway.ID},)
            record_values = GetRecordValues("EBRHPMS Hwy ntw", rh1,{"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/Hr1]", "[VMT/Hr2]", "[VMT/Hr3]", "[VMT/Hr4]", "[VMT/Hr5]", "[VMT/Hr6]", "[VMT/Hr7]", "[VMT/Hr8]", "[VMT/Hr9]", "[VMT/Hr10]", "[VMT/Hr11]", "[VMT/Hr12]", "[VMT/Hr13]", "[VMT/Hr14]", "[VMT/Hr15]", "[VMT/Hr16]", "[VMT/Hr17]", "[VMT/Hr18]", "[VMT/Hr19]", "[VMT/Hr20]", "[VMT/Hr21]", "[VMT/Hr22]", "[VMT/Hr23]", "[VMT/Hr24]"})
            Run = RunDbox("VMT_Hour", record_values)
        end
        vt1 = RunDbox("VMT Hour", record_values)
    end
    if vt1 = "LDGV" then do
        rh = LocateNearestRecord(point, 15.0)
        //ShowMessage("The nearest id is"+ String(highway.ID))
        rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID",{highway.ID},)
        record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1,{"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/LDGV/Hr1]", "[VMT/FC/LDGV/Hr2]", "[VMT/FC/LDGV/Hr3]", "[VMT/FC/LDGV/Hr4]", "[VMT/FC/LDGV/Hr5]", "[VMT/FC/LDGV/Hr6]", "[VMT/FC/LDGV/Hr7]", "[VMT/FC/LDGV/Hr8]", "[VMT/FC/LDGV/Hr9]", "[VMT/FC/LDGV/Hr10]", "[VMT/FC/LDGV/Hr11]", "[VMT/FC/LDGV/Hr12]"},

Run1 = RunDbox("VMT_FC_LDGV", record_values3)
end
if vt1 = "LDGT1" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/LDGT1/Hr1]", "[VMT/FC/LDGT1/Hr2]", "[VMT/FC/LDGT1/Hr3]", "[VMT/FC/LDGT1/Hr4]", "[VMT/FC/LDGT1/Hr5]", "[VMT/FC/LDGT1/Hr6]", "[VMT/FC/LDGT1/Hr7]", "[VMT/FC/LDGT1/Hr8]", "[VMT/FC/LDGT1/Hr9]", "[VMT/FC/LDGT1/Hr10]", "[VMT/FC/LDGT1/Hr11]", "[VMT/FC/LDGT1/Hr12]", "[VMT/FC/LDGT1/Hr13]", "[VMT/FC/LDGT1/Hr14]", "[VMT/FC/LDGT1/Hr15]", "[VMT/FC/LDGT1/Hr16]", "[VMT/FC/LDGT1/Hr17]", "[VMT/FC/LDGT1/Hr18]", "[VMT/FC/LDGT1/Hr19]", "[VMT/FC/LDGT1/Hr20]", "[VMT/FC/LDGT1/Hr21]", "[VMT/FC/LDGT1/Hr22]", "[VMT/FC/LDGT1/Hr23]", "[VMT/FC/LDGT1/Hr24]"})
Run1 = RunDbox("VMT_FC_LDGTV1", record_values3)
end
if vt1 = "LDGT2" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
Run1 = RunDbox("VMT_FC_LDGTV2", record_values3)
end
Run1 = RunDbox("VMT_FC_LDGT2", record_values3)
end
if vt1 = "LDGT3" then do
rh = LocateNearestRecord(point, 15.0)
//ShowMessage("The nearest id is"+ String(highway.ID))
rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID",
"RURAL_URBA","F_SYSTEM","SECTION_LE","SPEED_LIMI",
"F_Type","[VMT/FC/LDGT3/Hr1]", 
[VMT/FC/LDGT3/Hr2]",
[VMT/FC/LDGT3/Hr3]", [VMT/FC/LDGT3/Hr4]",
[VMT/FC/LDGT3/Hr5]", [VMT/FC/LDGT3/Hr6]",
[VMT/FC/LDGT3/Hr7]", [VMT/FC/LDGT3/Hr8]",
[VMT/FC/LDGT3/Hr9]", [VMT/FC/LDGT3/Hr10]",
[VMT/FC/LDGT3/Hr11]", [VMT/FC/LDGT3/Hr12]",
[VMT/FC/LDGT3/Hr13]", [VMT/FC/LDGT3/Hr14]",
[VMT/FC/LDGT3/Hr15]", [VMT/FC/LDGT3/Hr16]",
[VMT/FC/LDGT3/Hr17]",[VMT/FC/LDGT3/Hr18]",
[VMT/FC/LDGT3/Hr19]", [VMT/FC/LDGT3/Hr20]",
[VMT/FC/LDGT3/Hr21]", [VMT/FC/LDGT3/Hr22]",
[VMT/FC/LDGT3/Hr23]", [VMT/FC/LDGT3/Hr24]"})
Run1 = RunDbox("VMT_FC_LDGT3", record_values3)
end
if vt1 = "LDGT4" then do
rh = LocateNearestRecord(point, 15.0)
//ShowMessage("The nearest id is"+ String(highway.ID))
rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID",
"RURAL_URBA","F_SYSTEM","SECTION_LE",
"SPEED_LIMI","F_Type","[VMT/FC/LDGT4/Hr1]",
[VMT/FC/LDGT4/Hr2]",
[VMT/FC/LDGT4/Hr3]", [VMT/FC/LDGT4/Hr4]",
[VMT/FC/LDGT4/Hr5]", [VMT/FC/LDGT4/Hr6]",
[VMT/FC/LDGT4/Hr7]", [VMT/FC/LDGT4/Hr8]",
[VMT/FC/LDGT4/Hr9]", [VMT/FC/LDGT4/Hr10]",
[VMT/FC/LDGT4/Hr11]", [VMT/FC/LDGT4/Hr12]",
[VMT/FC/LDGT4/Hr13]", [VMT/FC/LDGT4/Hr14]",
[VMT/FC/LDGT4/Hr15]", [VMT/FC/LDGT4/Hr16]",
[VMT/FC/LDGT4/Hr17]",[VMT/FC/LDGT4/Hr18]",
[VMT/FC/LDGT4/Hr19]", [VMT/FC/LDGT4/Hr20]",
[VMT/FC/LDGT4/Hr21]", [VMT/FC/LDGT4/Hr22]",
[VMT/FC/LDGT4/Hr23]", [VMT/FC/LDGT4/Hr24]"})
Run1 = RunDbox("VMT_FC_LDGT4", record_values3)
end
if vt1 = "HDGV2b" then do
rh = LocateNearestRecord(point, 15.0)
//ShowMessage("The nearest id is"+ String(highway.ID))
rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", 
[VMT/FC/HDGV2b/Hr1]", [VMT/FC/HDGV2b/Hr2]", [VMT/FC/HDGV2b/Hr3]", [VMT/FC/HDGV2b/Hr4]", [VMT/FC/HDGV2b/Hr5]", [VMT/FC/HDGV2b/Hr6]", [VMT/FC/HDGV2b/Hr7]", [VMT/FC/HDGV2b/Hr8]", [VMT/FC/HDGV2b/Hr9]", [VMT/FC/HDGV2b/Hr10]", [VMT/FC/HDGV2b/Hr11]", [VMT/FC/HDGV2b/Hr12]", [VMT/FC/HDGV2b/Hr13]", [VMT/FC/HDGV2b/Hr14]", [VMT/FC/HDGV2b/Hr15]", [VMT/FC/HDGV2b/Hr16]", [VMT/FC/HDGV2b/Hr17]", [VMT/FC/HDGV2b/Hr18]", [VMT/FC/HDGV2b/Hr19]", [VMT/FC/HDGV2b/Hr20]", [VMT/FC/HDGV2b/Hr21]", [VMT/FC/HDGV2b/Hr22]", [VMT/FC/HDGV2b/Hr23]", [VMT/FC/HDGV2b/Hr24]})
Run1 = RunDbox("VMT_FC_HDGV2b", record_values3)
end
if vt1 = "HDGV3" then do
rh = LocateNearestRecord(point, 15.0)
//ShowMessage("The nearest id is" + String(highway.ID))
rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
Run1 = RunDbox("VMT_FC_HDGV3", record_values3)
end
if vt1 = "HDGV4" then do
rh = LocateNearestRecord(point, 15.0)
//ShowMessage("The nearest id is" + String(highway.ID))
rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", 
Run1 = RunDbox("VMT_FC_HDGV4", record_values3)
end
Run1 = RunDbox("VMT_FC_HDGV4", record_values3)

if vt1 = "HDGV5" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, "ID",
                           "RURAL_URBA", "F_SYSTEM", "SECTION_LE",
                           "SPEED_LIMI", "F_Type", "[VMT/FC/HDGV5/Hr1]",
                           "[VMT/FC/HDGV5/Hr2]",
                           "[VMT/FC/HDGV5/Hr3]", "[VMT/FC/HDGV5/Hr4]",
                           "[VMT/FC/HDGV5/Hr5]", "[VMT/FC/HDGV5/Hr6]",
                           "[VMT/FC/HDGV5/Hr7]", "[VMT/FC/HDGV5/Hr8]",
                           "[VMT/FC/HDGV5/Hr9]", "[VMT/FC/HDGV5/Hr10]",
                           "[VMT/FC/HDGV5/Hr11]", "[VMT/FC/HDGV5/Hr12]",
                           "[VMT/FC/HDGV5/Hr13]", "[VMT/FC/HDGV5/Hr14]",
                           "[VMT/FC/HDGV5/Hr15]", "[VMT/FC/HDGV5/Hr16]",
                           "[VMT/FC/HDGV5/Hr17]", "[VMT/FC/HDGV5/Hr18]",
                           "[VMT/FC/HDGV5/Hr19]", "[VMT/FC/HDGV5/Hr20]",
                           "[VMT/FC/HDGV5/Hr21]", "[VMT/FC/HDGV5/Hr22]",
                           "[VMT/FC/HDGV5/Hr23]", "[VMT/FC/HDGV5/Hr24]"})
    Run1 = RunDbox("VMT_FC_HDGV5", record_values3)
end

if vt1 = "HDGV6" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, "ID",
                           "RURAL_URBA", "F_SYSTEM", "SECTION_LE",
                           "SPEED_LIMI", "F_Type", "[VMT/FC/HDGV6/Hr1]",
                           "[VMT/FC/HDGV6/Hr2]",
                           "[VMT/FC/HDGV6/Hr3]", "[VMT/FC/HDGV6/Hr4]",
                           "[VMT/FC/HDGV6/Hr5]", "[VMT/FC/HDGV6/Hr6]",
                           "[VMT/FC/HDGV6/Hr7]", "[VMT/FC/HDGV6/Hr8]",
                           "[VMT/FC/HDGV6/Hr9]", "[VMT/FC/HDGV6/Hr10]",
                           "[VMT/FC/HDGV6/Hr11]", "[VMT/FC/HDGV6/Hr12]",
                           "[VMT/FC/HDGV6/Hr13]", "[VMT/FC/HDGV6/Hr14]",
                           "[VMT/FC/HDGV6/Hr15]", "[VMT/FC/HDGV6/Hr16]",
                           "[VMT/FC/HDGV6/Hr17]", "[VMT/FC/HDGV6/Hr18]",
                           "[VMT/FC/HDGV6/Hr19]", "[VMT/FC/HDGV6/Hr20]",
                           "[VMT/FC/HDGV6/Hr21]", "[VMT/FC/HDGV6/Hr22]",
                           "[VMT/FC/HDGV6/Hr23]", "[VMT/FC/HDGV6/Hr24]"})
    Run1 = RunDbox("VMT_FC_HDGV6", record_values3)
end
Run1 = RunDbox("VMT_FC_HDGV6", record_values3)
end
if vt1 = "HDGV7" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", highway.ID,)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, ID, RURAL_URBA, F_SYSTEM, SECTION_LE, SPEED_LIMIT, F_Type, [VMT/FC/HDGV7/Hr1], [VMT/FC/HDGV7/Hr2], [VMT/FC/HDGV7/Hr3], [VMT/FC/HDGV7/Hr4], [VMT/FC/HDGV7/Hr5], [VMT/FC/HDGV7/Hr6], [VMT/FC/HDGV7/Hr7], [VMT/FC/HDGV7/Hr8], [VMT/FC/HDGV7/Hr9], [VMT/FC/HDGV7/Hr10], [VMT/FC/HDGV7/Hr11], [VMT/FC/HDGV7/Hr12], [VMT/FC/HDGV7/Hr13], [VMT/FC/HDGV7/Hr14], [VMT/FC/HDGV7/Hr15], [VMT/FC/HDGV7/Hr16], [VMT/FC/HDGV7/Hr17], [VMT/FC/HDGV7/Hr18], [VMT/FC/HDGV7/Hr19], [VMT/FC/HDGV7/Hr20], [VMT/FC/HDGV7/Hr21], [VMT/FC/HDGV7/Hr22], [VMT/FC/HDGV7/Hr23], [VMT/FC/HDGV7/Hr24])
  Run1 = RunDbox("VMT_FC_HDGV7", record_values3)
end
if vt1 = "HDGV8a" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", highway.ID,)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, ID, RURAL_URBA, F_SYSTEM, SECTION_LE, SPEED_LIMIT, F_Type, [VMT/FC/HDGV8a/Hr1], [VMT/FC/HDGV8a/Hr2], [VMT/FC/HDGV8a/Hr3], [VMT/FC/HDGV8a/Hr4], [VMT/FC/HDGV8a/Hr5], [VMT/FC/HDGV8a/Hr6], [VMT/FC/HDGV8a/Hr7], [VMT/FC/HDGV8a/Hr8], [VMT/FC/HDGV8a/Hr9], [VMT/FC/HDGV8a/Hr10], [VMT/FC/HDGV8a/Hr11], [VMT/FC/HDGV8a/Hr12], [VMT/FC/HDGV8a/Hr13], [VMT/FC/HDGV8a/Hr14], [VMT/FC/HDGV8a/Hr15], [VMT/FC/HDGV8a/Hr16], [VMT/FC/HDGV8a/Hr17], [VMT/FC/HDGV8a/Hr18], [VMT/FC/HDGV8a/Hr19], [VMT/FC/HDGV8a/Hr20], [VMT/FC/HDGV8a/Hr21], [VMT/FC/HDGV8a/Hr22], [VMT/FC/HDGV8a/Hr23], [VMT/FC/HDGV8a/Hr24])
  Run1 = RunDbox("VMT_FC_HDGV8a", record_values3)
end
Run1 = RunDbox("VMT_FC_HDGV8a", record_values3)
end
if vt1 = "HDGV8b" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw|", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID",
  "RURAL_URBA", "F_SYSTEM", "SECTION_LE",
  "SPEED_LIMI", "F_Type", "[VMT/FC/HDGV8b/Hr1]",
  "[VMT/FC/HDGV8b/Hr2]", "[VMT/FC/HDGV8b/Hr3]",
  "[VMT/FC/HDGV8b/Hr4]", "[VMT/FC/HDGV8b/Hr5]",
  "[VMT/FC/HDGV8b/Hr6]", "[VMT/FC/HDGV8b/Hr7]",
  "[VMT/FC/HDGV8b/Hr8]", "[VMT/FC/HDGV8b/Hr9]",
  "[VMT/FC/HDGV8b/Hr10]", "[VMT/FC/HDGV8b/Hr11]",
  "[VMT/FC/HDGV8b/Hr12]", "[VMT/FC/HDGV8b/Hr13]",
  "[VMT/FC/HDGV8b/Hr14]", "[VMT/FC/HDGV8b/Hr15]",
  "[VMT/FC/HDGV8b/Hr16]", "[VMT/FC/HDGV8b/Hr17]",
  "[VMT/FC/HDGV8b/Hr18]", "[VMT/FC/HDGV8b/Hr19]",
  "[VMT/FC/HDGV8b/Hr20]", "[VMT/FC/HDGV8b/Hr21]",
  "[VMT/FC/HDGV8b/Hr22]", "[VMT/FC/HDGV8b/Hr23]",
  "[VMT/FC/HDGV8b/Hr24]"})
  Run1 = RunDbox("VMT_FC_HDGV8b", record_values3)
end
if vt1 = "LDDV" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw|", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID",
  "RURAL_URBA", "F_SYSTEM", "SECTION_LE",
  "SPEED_LIMI", "F_Type", "[VMT/FC/LDDV/Hr1]",
  "[VMT/FC/LDDV/Hr2]", "[VMT/FC/LDDV/Hr3]",
  "[VMT/FC/LDDV/Hr4]", "[VMT/FC/LDDV/Hr5]",
  "[VMT/FC/LDDV/Hr6]", "[VMT/FC/LDDV/Hr7]",
  "[VMT/FC/LDDV/Hr8]", "[VMT/FC/LDDV/Hr9]",
  "[VMT/FC/LDDV/Hr10]", "[VMT/FC/LDDV/Hr11]",
  "[VMT/FC/LDDV/Hr12]", "[VMT/FC/LDDV/Hr13]",
  "[VMT/FC/LDDV/Hr14]", "[VMT/FC/LDDV/Hr15]",
  "[VMT/FC/LDDV/Hr16]", "[VMT/FC/LDDV/Hr17]",
  "[VMT/FC/LDDV/Hr18]", "[VMT/FC/LDDV/Hr19]",
  "[VMT/FC/LDDV/Hr20]", "[VMT/FC/LDDV/Hr21]",
  "[VMT/FC/LDDV/Hr22]", "[VMT/FC/LDDV/Hr23]",
  "[VMT/FC/LDDV/Hr24]"})
  Run1 = RunDbox("VMT_FC_HDGV8b", record_values3)
end
Run1 = RunDbox("VMT_FC_LDDV", record_values3)
end
if vt1 = "LDDT12" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, 
  {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", 
  "SPEED_LIMI", "F_Type", 
  [VMT/FC/LDDT12/Hr1], [VMT/FC/LDDT12/Hr2], 
  [VMT/FC/LDDT12/Hr3], [VMT/FC/LDDT12/Hr4], 
  [VMT/FC/LDDT12/Hr5], [VMT/FC/LDDT12/Hr6], 
  [VMT/FC/LDDT12/Hr7], [VMT/FC/LDDT12/Hr8],
  [VMT/FC/LDDT12/Hr9], [VMT/FC/LDDT12/Hr10], 
  [VMT/FC/LDDT12/Hr11], [VMT/FC/LDDT12/Hr12], 
  [VMT/FC/LDDT12/Hr13], [VMT/FC/LDDT12/Hr14], 
  [VMT/FC/LDDT12/Hr15], [VMT/FC/LDDT12/Hr16],
  [VMT/FC/LDDT12/Hr17], [VMT/FC/LDDT12/Hr18], 
  [VMT/FC/LDDT12/Hr19], [VMT/FC/LDDT12/Hr20], 
  [VMT/FC/LDDT12/Hr21], [VMT/FC/LDDT12/Hr22],
  [VMT/FC/LDDT12/Hr23], [VMT/FC/LDDT12/Hr24])
Run1 = RunDbox("VMT_FC_LDDT12", record_values3)
end
if vt1 = "HDDV2b" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, 
  {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", 
  "SPEED_LIMI", "F_Type", 
  [VMT/FC/HDDV2b/Hr1], [VMT/FC/HDDV2b/Hr2], 
  [VMT/FC/HDDV2b/Hr3], [VMT/FC/HDDV2b/Hr4], 
  [VMT/FC/HDDV2b/Hr5], [VMT/FC/HDDV2b/Hr6], 
  [VMT/FC/HDDV2b/Hr7], [VMT/FC/HDDV2b/Hr8],
  [VMT/FC/HDDV2b/Hr9], [VMT/FC/HDDV2b/Hr10], 
  [VMT/FC/HDDV2b/Hr11], [VMT/FC/HDDV2b/Hr12], 
  [VMT/FC/HDDV2b/Hr13], [VMT/FC/HDDV2b/Hr14], 
  [VMT/FC/HDDV2b/Hr15], [VMT/FC/HDDV2b/Hr16],
  [VMT/FC/HDDV2b/Hr17], [VMT/FC/HDDV2b/Hr18], 
  [VMT/FC/HDDV2b/Hr19], [VMT/FC/HDDV2b/Hr20], 
  [VMT/FC/HDDV2b/Hr21], [VMT/FC/HDDV2b/Hr22],
  [VMT/FC/HDDV2b/Hr23], [VMT/FC/HDDV2b/Hr24])
Run1 = RunDbox("VMT_FC_HDDV2b", record_values3)
end
if vt1 = "HDDV3" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw","ID",{highway.ID},)
    Run1 = RunDbox("VMT_FC_HDDV3", record_values3)
end
if vt1 = "HDDV4" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw","ID",{highway.ID},)
    Run1 = RunDbox("VMT_FC_HDDV4", record_values3)
end
if vt1 = "HDDV5" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw","ID",{highway.ID},)
record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", 
[VMT/FC/HDDV5/Hr1]", 
[VMT/FC/HDDV5/Hr2]", 
[VMT/FC/HDDV5/Hr3]", 
[VMT/FC/HDDV5/Hr4]", 
[VMT/FC/HDDV5/Hr5]", 
[VMT/FC/HDDV5/Hr6]", 
[VMT/FC/HDDV5/Hr7]", 
[VMT/FC/HDDV5/Hr8]", 
[VMT/FC/HDDV5/Hr9]", 
[VMT/FC/HDDV5/Hr10]", 
[VMT/FC/HDDV5/Hr11]", 
[VMT/FC/HDDV5/Hr12]", 
[VMT/FC/HDDV5/Hr13]", 
[VMT/FC/HDDV5/Hr14]", 
[VMT/FC/HDDV5/Hr15]", 
[VMT/FC/HDDV5/Hr16]", 
[VMT/FC/HDDV5/Hr17]", 
[VMT/FC/HDDV5/Hr18]", 
[VMT/FC/HDDV5/Hr19]", 
[VMT/FC/HDDV5/Hr20]", 
[VMT/FC/HDDV5/Hr21]", 
[VMT/FC/HDDV5/Hr22]", 
[VMT/FC/HDDV5/Hr23]", 
[VMT/FC/HDDV5/Hr24]})
Run1 = RunDbox("VMT_FC_HDDV5", record_values3)
end
if vt1 = "HDDV6" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", 
[VMT/FC/HDDV6/Hr1]", 
[VMT/FC/HDDV6/Hr2]", 
[VMT/FC/HDDV6/Hr3]", 
[VMT/FC/HDDV6/Hr4]", 
[VMT/FC/HDDV6/Hr5]", 
[VMT/FC/HDDV6/Hr6]", 
[VMT/FC/HDDV6/Hr7]", 
[VMT/FC/HDDV6/Hr8]", 
[VMT/FC/HDDV6/Hr9]", 
[VMT/FC/HDDV6/Hr10]", 
[VMT/FC/HDDV6/Hr11]", 
[VMT/FC/HDDV6/Hr12]", 
[VMT/FC/HDDV6/Hr13]", 
[VMT/FC/HDDV6/Hr14]", 
[VMT/FC/HDDV6/Hr15]", 
[VMT/FC/HDDV6/Hr16]", 
[VMT/FC/HDDV6/Hr17]", 
[VMT/FC/HDDV6/Hr18]", 
[VMT/FC/HDDV6/Hr19]", 
[VMT/FC/HDDV6/Hr20]", 
[VMT/FC/HDDV6/Hr21]", 
[VMT/FC/HDDV6/Hr22]", 
[VMT/FC/HDDV6/Hr23]", 
[VMT/FC/HDDV6/Hr24]})
Run1 = RunDbox("VMT_FC_HDDV6", record_values3)
end
if vt1 = "HDDV7" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is"+ String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", 
[VMT/FC/HDDV7/Hr1]", 
[VMT/FC/HDDV7/Hr2]", 
[VMT/FC/HDDV7/Hr3]", 
[VMT/FC/HDDV7/Hr4]", 
[VMT/FC/HDDV7/Hr5]", 
[VMT/FC/HDDV7/Hr6]", 
[VMT/FC/HDDV7/Hr7]", 
[VMT/FC/HDDV7/Hr8]", 
[VMT/FC/HDDV7/Hr9]", 
[VMT/FC/HDDV7/Hr10]", 
[VMT/FC/HDDV7/Hr11]", 
[VMT/FC/HDDV7/Hr12]", 
[VMT/FC/HDDV7/Hr13]", 
[VMT/FC/HDDV7/Hr14]", 
[VMT/FC/HDDV7/Hr15]", 
[VMT/FC/HDDV7/Hr16]", 
[VMT/FC/HDDV7/Hr17]", 
[VMT/FC/HDDV7/Hr18]", 
[VMT/FC/HDDV7/Hr19]", 
[VMT/FC/HDDV7/Hr20]", 
[VMT/FC/HDDV7/Hr21]", 
[VMT/FC/HDDV7/Hr22]", 
[VMT/FC/HDDV7/Hr23]", 
[VMT/FC/HDDV7/Hr24]})
Run1 = RunDbox("VMT_FC_HDDV7", record_values3)
end
Run1 = RunDbox("VMT_FC_HDDV7", record_values3)
if vt1 = "HDDV8a" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is" + String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/HDDV8a/Hr1]", "[VMT/FC/HDDV8a/Hr2]", "[VMT/FC/HDDV8a/Hr3]", "[VMT/FC/HDDV8a/Hr4]", "[VMT/FC/HDDV8a/Hr5]", "[VMT/FC/HDDV8a/Hr6]", "[VMT/FC/HDDV8a/Hr7]", "[VMT/FC/HDDV8a/Hr8]", "[VMT/FC/HDDV8a/Hr9]", "[VMT/FC/HDDV8a/Hr10]", "[VMT/FC/HDDV8a/Hr11]", "[VMT/FC/HDDV8a/Hr12]", "[VMT/FC/HDDV8a/Hr13]", "[VMT/FC/HDDV8a/Hr14]", "[VMT/FC/HDDV8a/Hr15]", "[VMT/FC/HDDV8a/Hr16]", "[VMT/FC/HDDV8a/Hr17]", "[VMT/FC/HDDV8a/Hr18]", "[VMT/FC/HDDV8a/Hr19]", "[VMT/FC/HDDV8a/Hr20]", "[VMT/FC/HDDV8a/Hr21]", "[VMT/FC/HDDV8a/Hr22]", "[VMT/FC/HDDV8a/Hr23]", "[VMT/FC/HDDV8a/Hr24]"})
  Run1 = RunDbox("VMT_FC_HDDV8a", record_values3)
end
if vt1 = "HDDV8b" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is" + String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/HDDV8b/Hr1]", "[VMT/FC/HDDV8b/Hr2]", "[VMT/FC/HDDV8b/Hr3]", "[VMT/FC/HDDV8b/Hr4]", "[VMT/FC/HDDV8b/Hr5]", "[VMT/FC/HDDV8b/Hr6]", "[VMT/FC/HDDV8b/Hr7]", "[VMT/FC/HDDV8b/Hr8]", "[VMT/FC/HDDV8b/Hr9]", "[VMT/FC/HDDV8b/Hr10]", "[VMT/FC/HDDV8b/Hr11]", "[VMT/FC/HDDV8b/Hr12]", "[VMT/FC/HDDV8b/Hr13]", "[VMT/FC/HDDV8b/Hr14]", "[VMT/FC/HDDV8b/Hr15]", "[VMT/FC/HDDV8b/Hr16]", "[VMT/FC/HDDV8b/Hr17]", "[VMT/FC/HDDV8b/Hr18]", "[VMT/FC/HDDV8b/Hr19]", "[VMT/FC/HDDV8b/Hr20]", "[VMT/FC/HDDV8b/Hr21]", "[VMT/FC/HDDV8b/Hr22]", "[VMT/FC/HDDV8b/Hr23]", "[VMT/FC/HDDV8b/Hr24]"})
  Run1 = RunDbox("VMT_FC_HDDV8b", record_values3)
end
Run1 = RunDbox("VMT_FC_HDDV8b", record_values3)
end
if vt1 = "MC" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is" + String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/MC/Hr1]", 
  
  
  "[VMT/FC/MC/Hr2]", "[VMT/FC/MC/Hr3]", 
  "[VMT/FC/MC/Hr4]", "[VMT/FC/MC/Hr5]", 
  "[VMT/FC/MC/Hr6]", "[VMT/FC/MC/Hr7]", 
  "[VMT/FC/MC/Hr8]", "[VMT/FC/MC/Hr9]", 
  "[VMT/FC/MC/Hr10]", "[VMT/FC/MC/Hr11]", 
  "[VMT/FC/MC/Hr12]", "[VMT/FC/MC/Hr13]", 
  "[VMT/FC/MC/Hr14]", "[VMT/FC/MC/Hr15]", 
  "[VMT/FC/MC/Hr16]","[VMT/FC/MC/Hr17]", 
  "[VMT/FC/MC/Hr18]", "[VMT/FC/MC/Hr19]", 
  "[VMT/FC/MC/Hr20]", "[VMT/FC/MC/Hr21]", 
  "[VMT/FC/MC/Hr22]", "[VMT/FC/MC/Hr23]", 
  "[VMT/FC/MC/Hr24]"})
  Run1 = RunDbox("VMT_FC_MC", record_values3)
end
if vt1 = "HDGB" then do
  rh = LocateNearestRecord(point, 15.0)
  //ShowMessage("The nearest id is" + String(highway.ID))
  rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
  record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", 
  "[VMT/FC/HDGB/Hr1]", "[VMT/FC/HDGB/Hr2]", 
  "[VMT/FC/HDGB/Hr3]", "[VMT/FC/HDGB/Hr4]", 
  "[VMT/FC/HDGB/Hr5]", "[VMT/FC/HDGB/Hr6]", 
  "[VMT/FC/HDGB/Hr7]", "[VMT/FC/HDGB/Hr8]", "[VMT/FC/HDGB/Hr9]", 
  "[VMT/FC/HDGB/Hr10]", "[VMT/FC/HDGB/Hr11]", 
  "[VMT/FC/HDGB/Hr12]", "[VMT/FC/HDGB/Hr13]", 
  "[VMT/FC/HDGB/Hr14]", "[VMT/FC/HDGB/Hr15]", 
  "[VMT/FC/HDGB/Hr16]", "[VMT/FC/HDGB/Hr17]", 
  "[VMT/FC/HDGB/Hr18]", "[VMT/FC/HDGB/Hr19]", 
  "[VMT/FC/HDGB/Hr20]", "[VMT/FC/HDGB/Hr21]", 
  "[VMT/FC/HDGB/Hr22]", "[VMT/FC/HDGB/Hr23]", 
  "[VMT/FC/HDGB/Hr24]"})
  Run1 = RunDbox("VMT_FC_HDBV8b", record_values3)
end
Run1 = RunDbox("VMT_FC_HDGB", record_values3)

if vt1 = "HDDBT" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", [VMT/FC/HDDBT/Hr1], [VMT/FC/HDDBT/Hr2], [VMT/FC/HDDBT/Hr3], [VMT/FC/HDDBT/Hr4], [VMT/FC/HDDBT/Hr5], [VMT/FC/HDDBT/Hr6], [VMT/FC/HDDBT/Hr7], [VMT/FC/HDDBT/Hr8], [VMT/FC/HDDBT/Hr9], [VMT/FC/HDDBT/Hr10], [VMT/FC/HDDBT/Hr11], [VMT/FC/HDDBT/Hr12], [VMT/FC/HDDBT/Hr13], [VMT/FC/HDDBT/Hr14], [VMT/FC/HDDBT/Hr15], [VMT/FC/HDDBT/Hr16], [VMT/FC/HDDBT/Hr17], [VMT/FC/HDDBT/Hr18], [VMT/FC/HDDBT/Hr19], [VMT/FC/HDDBT/Hr20], [VMT/FC/HDDBT/Hr21], [VMT/FC/HDDBT/Hr22], [VMT/FC/HDDBT/Hr23], [VMT/FC/HDDBT/Hr24]})
    Run1 = RunDbox("VMT_FC_HDDBT", record_values3)
end

if vt1 = "HDDBS" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", [VMT/FC/HDDBS/Hr1], [VMT/FC/HDDBS/Hr2], [VMT/FC/HDDBS/Hr3], [VMT/FC/HDDBS/Hr4], [VMT/FC/HDDBS/Hr5], [VMT/FC/HDDBS/Hr6], [VMT/FC/HDDBS/Hr7], [VMT/FC/HDDBS/Hr8], [VMT/FC/HDDBS/Hr9], [VMT/FC/HDDBS/Hr10], [VMT/FC/HDDBS/Hr11], [VMT/FC/HDDBS/Hr12], [VMT/FC/HDDBS/Hr13], [VMT/FC/HDDBS/Hr14], [VMT/FC/HDDBS/Hr15], [VMT/FC/HDDBS/Hr16], [VMT/FC/HDDBS/Hr17], [VMT/FC/HDDBS/Hr18], [VMT/FC/HDDBS/Hr19], [VMT/FC/HDDBS/Hr20], [VMT/FC/HDDBS/Hr21], [VMT/FC/HDDBS/Hr22], [VMT/FC/HDDBS/Hr23], [VMT/FC/HDDBS/Hr24]})
    Run1 = RunDbox("VMT_FC_HDDBS", record_values3)
end
Run1 = RunDbox("VMT_FC_HDDBS", record_values3)
end
if vt1 = "LDDT34" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values3 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/LDDT34/Hr1]", "[VMT/FC/LDDT34/Hr2]", "[VMT/FC/LDDT34/Hr3]", "[VMT/FC/LDDT34/Hr4]", "[VMT/FC/LDDT34/Hr5]", "[VMT/FC/LDDT34/Hr6]", "[VMT/FC/LDDT34/Hr7]", "[VMT/FC/LDDT34/Hr8]", "[VMT/FC/LDDT34/Hr9]", "[VMT/FC/LDDT34/Hr10]", "[VMT/FC/LDDT34/Hr11]", "[VMT/FC/LDDT34/Hr12]", "[VMT/FC/LDDT34/Hr13]", "[VMT/FC/LDDT34/Hr14]", "[VMT/FC/LDDT34/Hr15]", "[VMT/FC/LDDT34/Hr16]", "[VMT/FC/LDDT34/Hr17]", "[VMT/FC/LDDT34/Hr18]", "[VMT/FC/LDDT34/Hr19]", "[VMT/FC/LDDT34/Hr20]", "[VMT/FC/LDDT34/Hr21]", "[VMT/FC/LDDT34/Hr22]", "[VMT/FC/LDDT34/Hr23]", "[VMT/FC/LDDT34/Hr24]"})
    Run1 = RunDbox("VMT_FC_LDDT34", record_values3)
end
if vmt_inputs = "VMT Mix" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values1 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMTmix/LDV]", "[VMTmix/LDT1]", "[VMTmix/LDT2]", "[VMTmix/LDT3]", "[VMTmix/LDT4]", "[VMTmix/HDV2B]", "[VMTmix/HDV3]", "[VMTmix/HDV4]", "[VMTmix/HDV5]", "[VMTmix/HDV6]", "[VMTmix/HDV711]", "[VMTmix/HDV8A]", "[VMTmix/HDV8B]", "[VMTmix/HDVBS]", "[VMTmix/HDVBT]", "[VMTmix/MC]"})
    Run2 = RunDbox("VMT_Mix", record_values1)
end
if vmt_inputs = "VMT by Speed" then do
    rh = LocateNearestRecord(point, 15.0)
    //ShowMessage("The nearest id is"+ String(highway.ID))
    rh1 = LocateRecord("EBRHPMS Hwy ntw", "ID", {highway.ID},)
    record_values2 = GetRecordValues("EBRHPMS Hwy ntw", rh1, {"ID", "RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", "[VMT/FC/LDDT34/Hr1]", "[VMT/FC/LDDT34/Hr2]", "[VMT/FC/LDDT34/Hr3]", "[VMT/FC/LDDT34/Hr4]", "[VMT/FC/LDDT34/Hr5]", "[VMT/FC/LDDT34/Hr6]", "[VMT/FC/LDDT34/Hr7]", "[VMT/FC/LDDT34/Hr8]", "[VMT/FC/LDDT34/Hr9]", "[VMT/FC/LDDT34/Hr10]", "[VMT/FC/LDDT34/Hr11]", "[VMT/FC/LDDT34/Hr12]", "[VMT/FC/LDDT34/Hr13]", "[VMT/FC/LDDT34/Hr14]", "[VMT/FC/LDDT34/Hr15]", "[VMT/FC/LDDT34/Hr16]", "[VMT/FC/LDDT34/Hr17]", "[VMT/FC/LDDT34/Hr18]", "[VMT/FC/LDDT34/Hr19]", "[VMT/FC/LDDT34/Hr20]", "[VMT/FC/LDDT34/Hr21]", "[VMT/FC/LDDT34/Hr22]", "[VMT/FC/LDDT34/Hr23]", "[VMT/FC/LDDT34/Hr24]"})
    Run1 = RunDbox("VMT_FC_LDDT34", record_values3)
end
"RURAL_URBA", "F_SYSTEM", "SECTION_LE", "SPEED_LIMI", "F_Type", 
"[SpeedVMT/Hr1]", 
"[SpeedVMT/Hr2]", "[SpeedVMT/Hr3]", "[SpeedVMT/Hr4]", "[SpeedVMT/Hr5]", "[SpeedVMT/Hr6]", "[SpeedVMT/Hr7]", "[SpeedVMT/Hr8]", "[SpeedVMT/Hr9]", 
"[SpeedVMT/Hr10]", "[SpeedVMT/Hr11]", "[SpeedVMT/Hr12]", "[SpeedVMT/Hr13]", "[SpeedVMT/Hr14]", "[SpeedVMT/Hr15]", 
"[SpeedVMT/Hr16]", "[SpeedVMT/Hr17]", "[SpeedVMT/Hr18]", "[SpeedVMT/Hr19]", "[SpeedVMT/Hr20]", "[SpeedVMT/Hr21]", 
"[SpeedVMT/Hr22]", "[SpeedVMT/Hr23]", "[SpeedVMT/Hr24]"
)
Run3 = RunDbox("VMT_Speed", record_values2)
end

if options = "System VMT" then
  vmt = RunDbox("Opts")
  if vmt = "Urban" then do
    ur = RunMacro("urban", fnm)
  end
  if vmt = "Donut" then do
    don = RunMacro("donut", fnm)
  end
  if vmt = "Total" then do
    tot = RunMacro("total", fnm)
  end
end
endItem
endDbox

eDbox "VMT_Hour" (record_values)
Init do
  dim values[record_values.length]
  for i=1 to record_values.length do
    values[i] = record_values[i][2]
  end

  new_array = {
    {{1, "L", "ID"}, {14, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {14, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {14, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {14, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {14, "L", values[5]}},
    {{1, "L", "F_Type"}, {14, "L", values[6]}},
    {{1, "L", "[VMT/Hr1]"}, {14, "L", values[7]}},
    {{1, "L", "[VMT/Hr2]"}, {14, "L", values[8]}},
    {{1, "L", "[VMT/Hr3]"}, {14, "L", values[9]}},
  }
Macro "total"
vw1 = GetviewLayer("EBRHPMS Hwy ntw")
SetView(vw1)
Answer = 0
DVMT = 0
TSL = 61027.35
for a = 1 to 4 do
    rec = GetFirstRecord("EBRHPMS Hwy ntw",)
    while rec <> null do
        ru = vw1.RURAL_URBA
        aadt = vw1.AADT
        TSL = 61027.35
        if ru = a then do
            DVMT = DVMT + aadt * TSL
        end
Macro "donut"
vw1 = GetviewLayer("EBRHPMS Hwy ntw")
SetView(vw1)
Answer = 0
DVMT = 0
TSL = 61027.35
for a = 1 to 2 do
    rec = GetFirstRecord("EBRHPMS Hwy ntw"),
    while rec <> null do
        ru = vw1.RURAL_URBA
        nc = vw1.NONATT_COD
        uc = vw1.URBAN_CODE
        aadt = vw1.AADT_CART
        TSL = 61027.35
        if ru = a & nc = 88 & uc = 0 then do
            DVMT = DVMT + aadt * TSL
        end
    rec = GetNextRecord("EBRHPMS Hwy ntw"),null,null)
end
//ShowMessage("Donut VMT " + " is " + String(DVMT))
Answer = Answer + DVMT
end
ShowMessage("Total VMT in the EBR non-attainment area is " +
    Format(Answer,"*0.00"))
endMacro

Macro "urban"
vw1 = GetviewLayer("EBRHPMS Hwy ntw")
SetView(vw1)
Answer = 0
DVMT = 0
TSL = 61027.35
for a = 3 to 4 do
    rec = GetFirstRecord("EBRHPMS Hwy ntw"),
    while rec <> null do
        ru = vw1.RURAL_URBA
        uc = vw1.URBAN_CODE
        aadt = vw1.AADT_CART
        TSL = 61027.35
        if ru = a & nc = 88 & uc = 0 then do
            DVMT = DVMT + aadt * TSL
        end
    rec = GetNextRecord("EBRHPMS Hwy ntw"),null,null)
end
//ShowMessage("Donut VMT for EBR Parish is" + Format(Answer,"*0.00"))
endMacro
TSL = 61027.35
if uc = 88 & ru = a then do
   DVMT = DVMT + aadt * TSL
end
   rec = GetNextRecord("EBRHPMS Hwy ntwj",null,null)
end
//ShowMessage("Urbanized VMT " + " is " + String(DVMT))
Answer = Answer + DVMT
end
ShowMessage("Urbanized VMT for EBR Parish is " + Format(Answer,"*,0.00"))
ednMacro

Dbox "Opts"
title: "Select System"
   button "Urban" 2, 2, 6
      do Return("Urban") enditem
   button "Donut" 10, 2, 6
      do Return("Donut") enditem
   button "Total" 18, 2, 6
      do Return("Total") enditem
endDBox

Dbox "VMT Distribution"
title: "VMT Distribution"
    init do
       str1 = "Do you want to see the VMT Distribution?"
    enditem
    text 1, 1, 60, 1
        variable: str1
        align: center
        button "Yes" 2, 4.5, 9
           do Return("Yes") enditem
       button "No" 62.5, 4.5, 9
           do Return("No") enditem
endDbox

DBox "Options"
title: "Select Option"
   button "MOBILE6 Default Values" 2, 3,20
      do Return("MOBILE6 Default Values") enditem
button "System VMT" 2,6, 20
do Return("System VMT") enditem
endDBox

Dbox "Choose VMT Inputs"
title: "Choose VMT Inputs"
button "VMT by Hour" 5, 1, 16
do Return("VMT by Hour") enditem
button "VMT by FC" 5, 3, 16
do Return("VMT by FC") enditem
button "VMT Mix" 5, 5, 16
do Return("VMT Mix") enditem
button "VMT by Speed" 5, 7, 16
do Return("VMT by Speed") enditem
endDBox

Dbox "Vehicle Type"
title: "Choose Vehicle Type"
button "LDGV" 5, 2, 8
do Return("LDGV") enditem
button "LDGT1" 15, 2, 8
do Return("LDGT1") enditem
button "LDGT2" 25, 2, 8
do Return("LDGT2") enditem
button "LDGT3" 35, 2, 8
do Return("LDGT3") enditem
button "LDGT4" 45, 2, 8
do Return("LDGT4") enditem
button "HDGV2b" 55, 2, 8
do Return("HDGV2b") enditem
button "HDGV3" 65, 2, 8
do Return("HDGV3") enditem
button "HDGV4" 5, 4, 8
do Return("HDGV4") enditem
button "HDGV5" 15, 4, 8
do Return("HDGV5") enditem
button "HDGV6" 25, 4, 8
do Return("HDGV6") enditem
button "HDGV7" 35, 4, 8
do Return("HDGV7") enditem
button "HDGV8a" 45, 4, 8
do Return("HDGV8a") enditem
button "HDGV8b" 55, 4, 8
do Return("HDGV8b") enditem
button "LDDV" 65, 4, 8
do Return("LDDV") enditem
button "LDDT12" 5, 6, 8
do Return("LDDT12") enditem
button "HDDV2b" 15, 6, 8
do Return("HDDV2b") enditem
button "HDDV3" 25, 6, 8
do Return("HDDV3") enditem
button "HDDV4" 35, 6, 8
do Return("HDDV4") enditem
button "HDDV5" 45, 6, 8
do Return("HDDV5") enditem
button "HDDV6" 55, 6, 8
do Return("HDDV6") enditem
button "HDDV7" 65, 6, 8
do Return("HDDV7") enditem
button "HDDV8a" 5, 8, 8
do Return("HDDV8a") enditem
button "HDDV8b" 15, 8, 8
do Return("HDDV8b") enditem
button "MC" 25, 8, 8
do Return("MC") enditem
button "HDGB" 35, 8, 8
do Return("HDGB") enditem
button "HDDBT" 45, 8, 8
do Return("HDDBT") enditem
button "HDDBS" 55, 8, 8
do Return("HDDBS") enditem
button "LDDT34" 65, 8, 8
do Return("LDDT34") enditem
Button "Close" 70.75, 10, 10 Cancel do
Return()
endItem
endDbox

Dbox"VMT_FC_LDGV" (record_values3)
Init do
dim values[record_values3.length]
for i=1 to values.length do
values[i] = record_values3[i][2]
new_array = {
  {{1, "L", "ID"}, {25, "L", values[1]}},
  {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
  {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
  {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
  {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
  {{1, "L", "F_Type"}, {25, "L", values[6]}},
  {{1, "L", "[VMT/FC/LDGV/Hr1]"}, {25, "L", values[7]}},
}
Dbox"VMT_FC_LDGTV1" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
new_array = { 
    {{1, "L", "ID"}, {25, "L", values[1]}}, 
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}}, 
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}}, 
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}}, 
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}}, 
    {{1, "L", "F_Type"}, {25, "L", values[6]}}, 
    {{1, "L", "[VMT/FC/LDGTV1/Hr1]"}, {25, "L", values[7]}}, 
    {{1, "L", "[VMT/FC/LDGTV1/Hr2]"}, {25, "L", values[8]}}, 
    { {1, "L", "[VMT/FC/LDGTV1/Hr3]"}, {25, "L", values[9]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr4]"}, {25, "L", values[10]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr5]"}, {25, "L", values[11]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr6]"}, {25, "L", values[12]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr7]"}, {25, "L", values[13]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr8]"}, {25, "L", values[14]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr9]"}, {25, "L", values[15]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr10]"}, {25, "L", values[16]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr11]"}, {25, "L", values[17]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr12]"}, {25, "L", values[18]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr13]"}, {25, "L", values[19]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr14]"}, {25, "L", values[20]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr15]"}, {25, "L", values[21]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr16]"}, {25, "L", values[22]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr17]"}, {25, "L", values[23]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr18]"}, {25, "L", values[24]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr19]"}, {25, "L", values[25]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr20]"}, {25, "L", values[26]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr21]"}, {25, "L", values[27]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr22]"}, {25, "L", values[28]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr23]"}, {25, "L", values[29]} },
    { {1, "L", "[VMT/FC/LDGTV1/Hr24]"}, {25, "L", values[30]} }

```plaintext
for i=1 to values.length do
    values[i] = record_values3[i][2]
endfor
```
End//for
EndItem
Scroll list "VMT/FC/LDGT2" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enddbox

Dbox"VMT_FC_LDGT3" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "[VMT/FC/LDGT3/Hr1]"}, {25, "L", values[1]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr2]"}, {25, "L", values[2]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr3]"}, {25, "L", values[3]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr4]"}, {25, "L", values[4]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr5]"}, {25, "L", values[5]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr6]"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr7]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr8]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr9]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr10]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr12]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr13]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr14]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr15]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr16]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr17]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr18]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr19]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr20]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr21]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr22]"}, {25, "L", values[22]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr23]"}, {25, "L", values[23]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr24]"}, {25, "L", values[24]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr26]"}, {25, "L", values[26]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr27]"}, {25, "L", values[27]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr28]"}, {25, "L", values[28]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr29]"}, {25, "L", values[29]}},
    {{1, "L", "[VMT/FC/LDGT3/Hr30]"}, {25, "L", values[30]}}
  }
  endItem
Enddbox
{{1, "L", "[VMT/FC/LDGT3/Hr5]"}, {25, "L", values[11]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr6]"}, {25, "L", values[12]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr7]"}, {25, "L", values[13]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr8]"}, {25, "L", values[14]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr9]"}, {25, "L", values[15]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr10]"}, {25, "L", values[16]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr11]"}, {25, "L", values[17]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr12]"}, {25, "L", values[18]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr13]"}, {25, "L", values[19]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr14]"}, {25, "L", values[20]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr15]"}, {25, "L", values[21]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr16]"}, {25, "L", values[22]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr17]"}, {25, "L", values[23]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr18]"}, {25, "L", values[24]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr19]"}, {25, "L", values[25]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr20]"}, {25, "L", values[26]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr21]"}, {25, "L", values[27]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr22]"}, {25, "L", values[28]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr23]"}, {25, "L", values[29]}}},
{{1, "L", "[VMT/FC/LDGT3/Hr24]"}, {25, "L", values[30]}}}
End//for
EndItem
scroll list "VMT/FC/LDGT3" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
ednItem
Enddbox

Dbox"VMT_FC_LDGT4" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}}},
    {{1, "L", "[VMT/FC/LDGT4/Hr1]"}, {25, "L", values[7]}}},
    {{1, "L", "[VMT/FC/LDGT4/Hr2]"}, {25, "L", values[8]}}},
    {{1, "L", "[VMT/FC/LDGT4/Hr3]"}, {25, "L", values[9]}}},
    {{1, "L", "[VMT/FC/LDGT4/Hr4]"}, {25, "L", values[10]}}},
    {{1, "L", "[VMT/FC/LDGT4/Hr5]"}, {25, "L", values[11]}}},
Dbox "VMT_FC_HDGV2b" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDGV2b/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDGV2b/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDGV2b/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDGV2b/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDGV2b/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDGV2b/Hr6]"}, {25, "L", values[12]}},
  }
End//for
EndItem
scroll list "VMT/FC/LDGT4" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
  Return()
endItem
Enddbox
Dbox"VMT_FC_HDGV3" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
  {{1, "L", "ID"}, {25, "L", values[1]}},
  {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
  {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
  {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
  {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
  {{1, "L", "F_Type"}, {25, "L", values[6]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr1]"}, {25, "L", values[7]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr2]"}, {25, "L", values[8]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr3]"}, {25, "L", values[9]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr4]"}, {25, "L", values[10]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr5]"}, {25, "L", values[11]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr6]"}, {25, "L", values[12]}},
  {{1, "L", "[VMT/FC/HGVDV3/Hr7]"}, {25, "L", values[13]}}};
Dbox"VMT_FC_HDGV4" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/HDGV4/Hr8]"}, {25, "L", values[14]}}
    }
End//for
EndItem
scroll list "VMT/FC/HDGV3" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
    Return()
endItem
Enddbox
{{1, "L", "[VMT/FC/HDGV4/Hr9]"}, {25, "L", values[15]}},
{{1, "L", "[VMT/FC/HDGV4/Hr10]"}, {25, "L", values[16]}},
{{1, "L", "[VMT/FC/HDGV4/Hr11]"}, {25, "L", values[17]}},
{{1, "L", "[VMT/FC/HDGV4/Hr12]"}, {25, "L", values[18]}},
{{1, "L", "[VMT/FC/HDGV4/Hr13]"}, {25, "L", values[19]}},
{{1, "L", "[VMT/FC/HDGV4/Hr14]"}, {25, "L", values[20]}},
{{1, "L", "[VMT/FC/HDGV4/Hr15]"}, {25, "L", values[21]}},
{{1, "L", "[VMT/FC/HDGV4/Hr16]"}, {25, "L", values[22]}},
{{1, "L", "[VMT/FC/HDGV4/Hr17]"}, {25, "L", values[23]}},
{{1, "L", "[VMT/FC/HDGV4/Hr18]"}, {25, "L", values[24]}},
{{1, "L", "[VMT/FC/HDGV4/Hr19]"}, {25, "L", values[25]}},
{{1, "L", "[VMT/FC/HDGV4/Hr20]"}, {25, "L", values[26]}},
{{1, "L", "[VMT/FC/HDGV4/Hr21]"}, {25, "L", values[27]}},
{{1, "L", "[VMT/FC/HDGV4/Hr22]"}, {25, "L", values[28]}},
{{1, "L", "[VMT/FC/HDGV4/Hr23]"}, {25, "L", values[29]}},
{{1, "L", "[VMT/FC/HDGV4/Hr24]"}, {25, "L", values[30]}}

End//for
EndItem
Scroll list "VMT/FC/HDGV4" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Endbbox

Dbox"VMT_FC_HDGV5" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr8]"}, {25, "L", values[14]}},
        {{1, "L", "[VMT/FC/HDGV5/Hr9]"}, {25, "L", values[15]}},
Dbox"VMT_FC_HDGV6" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {1, "L", "ID"}, {25, "L", values[1]},
        {1, "L", "RURAL_URBA"}, {25, "L", values[2]},
        {1, "L", "F_SYSTEM"}, {25, "L", values[3]},
        {1, "L", "SECTION_LE"}, {25, "L", values[4]},
        {1, "L", "SPEED_LIMI"}, {25, "L", values[5]},
        {1, "L", "F_Type"}, {25, "L", values[6]},
        {1, "L", "[VMT/FC/HDGV6/Hr1]"}, {25, "L", values[7]},
        {1, "L", "[VMT/FC/HDGV6/Hr2]"}, {25, "L", values[8]},
        {1, "L", "[VMT/FC/HDGV6/Hr3]"}, {25, "L", values[9]},
        {1, "L", "[VMT/FC/HDGV6/Hr4]"}, {25, "L", values[10]},
        {1, "L", "[VMT/FC/HDGV6/Hr5]"}, {25, "L", values[11]},
        {1, "L", "[VMT/FC/HDGV6/Hr6]"}, {25, "L", values[12]},
        {1, "L", "[VMT/FC/HDGV6/Hr7]"}, {25, "L", values[13]},
        {1, "L", "[VMT/FC/HDGV6/Hr8]"}, {25, "L", values[14]},
        {1, "L", "[VMT/FC/HDGV6/Hr9]"}, {25, "L", values[15]},
        {1, "L", "[VMT/FC/HDGV6/Hr10]"}, {25, "L", values[16]},
    }
EndItem
Dbox"VMT_FC_HDGV7" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr8]"}, {25, "L", values[14]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr9]"}, {25, "L", values[15]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr10]"}, {25, "L", values[16]}},
        {{1, "L", "[VMT/FC/HDGV7/Hr11]"}, {25, "L", values[17]}}
    End//for
EndItem
Scroll list "VMT/FC/HDGV6" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
    Return()
endItem
Enddbox
{{1, "L", "[VMT/FC/HDGV7/Hr12]"}, {25, "L", values[18]}},
{{1, "L", "[VMT/FC/HDGV7/Hr13]"}, {25, "L", values[19]}},
{{1, "L", "[VMT/FC/HDGV7/Hr14]"}, {25, "L", values[20]}},
{{1, "L", "[VMT/FC/HDGV7/Hr15]"}, {25, "L", values[21]}},
{{1, "L", "[VMT/FC/HDGV7/Hr16]"}, {25, "L", values[22]}},
{{1, "L", "[VMT/FC/HDGV7/Hr17]"}, {25, "L", values[23]}},
{{1, "L", "[VMT/FC/HDGV7/Hr18]"}, {25, "L", values[24]}},
{{1, "L", "[VMT/FC/HDGV7/Hr19]"}, {25, "L", values[25]}},
{{1, "L", "[VMT/FC/HDGV7/Hr20]"}, {25, "L", values[26]}},
{{1, "L", "[VMT/FC/HDGV7/Hr21]"}, {25, "L", values[27]}},
{{1, "L", "[VMT/FC/HDGV7/Hr22]"}, {25, "L", values[28]}},
{{1, "L", "[VMT/FC/HDGV7/Hr23]"}, {25, "L", values[29]}},
{{1, "L", "[VMT/FC/HDGV7/Hr24]"}, {25, "L", values[30]}}

End//for
EndItem
Scroll list "VMT/FC/HDGV7" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Endddbox

Dbox"VMT_FC_HDGV8a" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr8]"}, {25, "L", values[14]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr9]"}, {25, "L", values[15]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr10]"}, {25, "L", values[16]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr11]"}, {25, "L", values[17]}},
        {{1, "L", "[VMT/FC/HDGV8a/Hr12]"}, {25, "L", values[18]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr13]"}, {25, "L", values[19]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr14]"}, {25, "L", values[20]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr15]"}, {25, "L", values[21]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr16]"}, {25, "L", values[22]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr17]"}, {25, "L", values[23]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr18]"}, {25, "L", values[24]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr19]"}, {25, "L", values[25]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr20]"}, {25, "L", values[26]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr21]"}, {25, "L", values[27]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr22]"}, {25, "L", values[28]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr23]"}, {25, "L", values[29]}},
{{1, "L", "[VMT/FC/HDGV8a/Hr24]"}, {25, "L", values[30]}}}
End//for
EndItem
Scroll list "VMT/FC/HDGV8a" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enddbox
Dbox"VMT_FC_HDGV8b" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/HDGV8b/Hr13]"}, {25, "L", values[19]}},
Dbox"VMT_FC_LDDV" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/LDDV/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/LDDV/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/LDDV/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/LDDV/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/LDDV/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/LDDV/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/LDDV/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/LDDV/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/LDDV/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/LDDV/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/LDDV/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/LDDV/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/LDDV/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/LDDV/Hr14]"}, {25, "L", values[20]}}
  End//for
  EndItem
  Scroll list "VMT/FC/HDGV8b" 3,3,30,30
  List: new_array
  Button "OK" 40.75, 10, 10 Cancel do
    Return()
  endItem
Enddbox
Dbox"VMT_FC_LDDT12" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr8]"}, {25, "L", values[14]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr9]"}, {25, "L", values[15]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr10]"}, {25, "L", values[16]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr11]"}, {25, "L", values[17]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr12]"}, {25, "L", values[18]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr13]"}, {25, "L", values[19]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr14]"}, {25, "L", values[20]}},
        {{1, "L", "[VMT/FC/LDDT12/Hr15]"}, {25, "L", values[21]}},
new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr14]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr15]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/HDDV2b/Hr16]"}, {25, "L", values[22]}},
Dbbox"VMT_FC_HDDV3" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  end for
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr14]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr15]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr16]"}, {25, "L", values[22]}},
    {{1, "L", "[VMT/FC/HDDV3/Hr17]"}, {25, "L", values[23]}},
{{1, "L", "[VMT/FC/HDDV3/Hr18]"}, {25, "L", values[24]}},
{{1, "L", "[VMT/FC/HDDV3/Hr19]"}, {25, "L", values[25]}},
{{1, "L", "[VMT/FC/HDDV3/Hr20]"}, {25, "L", values[26]}},
{{1, "L", "[VMT/FC/HDDV3/Hr21]"}, {25, "L", values[27]}},
{{1, "L", "[VMT/FC/HDDV3/Hr22]"}, {25, "L", values[28]}},
{{1, "L", "[VMT/FC/HDDV3/Hr23]"}, {25, "L", values[29]}},
{{1, "L", "[VMT/FC/HDDV3/Hr24]"}, {25, "L", values[30]}}

End//for
EndItem
Scroll list "VMT/FC/HDDV3" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enddbox

Dbox"VMT_FC_HDDV4" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endItem
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr8]"}, {25, "L", values[14]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr9]"}, {25, "L", values[15]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr10]"}, {25, "L", values[16]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr11]"}, {25, "L", values[17]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr12]"}, {25, "L", values[18]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr13]"}, {25, "L", values[19]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr14]"}, {25, "L", values[20]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr15]"}, {25, "L", values[21]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr16]"}, {25, "L", values[22]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr17]"}, {25, "L", values[23]}},
        {{1, "L", "[VMT/FC/HDDV4/Hr18]"}, {25, "L", values[24]}},
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
{{1, "L", "[VMT/FC/HDDV5/Hr20]"}, {25, "L", values[26]}},
{{1, "L", "[VMT/FC/HDDV5/Hr21]"}, {25, "L", values[27]}},
{{1, "L", "[VMT/FC/HDDV5/Hr22]"}, {25, "L", values[28]}},
{{1, "L", "[VMT/FC/HDDV5/Hr23]"}, {25, "L", values[29]}},
{{1, "L", "[VMT/FC/HDDV5/Hr24]"}, {25, "L", values[30]}}
End//for
EndItem
Scroll list "VMT/FC/HDDV5" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return() endItem
Enddbox

Dbox"VMT_FC_HDDV6" (record_values3)
Init do
dim values[record_values3.length]
for i=1 to values.length do
values[i] = record_values3[i][2]
new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr14]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr15]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr16]"}, {25, "L", values[22]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr17]"}, {25, "L", values[23]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr18]"}, {25, "L", values[24]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr19]"}, {25, "L", values[25]}},
    {{1, "L", "[VMT/FC/HDDV6/Hr20]"}, {25, "L", values[26]}},
Dbox "VMT_FC_HDDV7" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  endfor
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr14]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr15]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr16]"}, {25, "L", values[22]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr17]"}, {25, "L", values[23]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr18]"}, {25, "L", values[24]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr19]"}, {25, "L", values[25]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr20]"}, {25, "L", values[26]}},
    {{1, "L", "[VMT/FC/HDDV7/Hr21]"}, {25, "L", values[27]}}
  }
End//for
EndItem
Scroll list "VMT/FC/HDDV6" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
  Return()
endItem
Enddbox
Dbox"VMT_FC_HDDV8a" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {% {1, "L", "ID"}, {25, "L", values[1]}%},
        {% {1, "L", "RURAL_URBA"}, {25, "L", values[2]}%},
        {% {1, "L", "F_SYSTEM"}, {25, "L", values[3]}%},
        {% {1, "L", "SECTION_LE"}, {25, "L", values[4]}%},
        {% {1, "L", "SPEED_LIMI"}, {25, "L", values[5]}%},
        {% {1, "L", "F_Type"}, {25, "L", values[6]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr1]"}, {25, "L", values[7]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr2]"}, {25, "L", values[8]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr3]"}, {25, "L", values[9]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr4]"}, {25, "L", values[10]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr5]"}, {25, "L", values[11]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr6]"}, {25, "L", values[12]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr7]"}, {25, "L", values[13]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr8]"}, {25, "L", values[14]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr9]"}, {25, "L", values[15]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr10]"}, {25, "L", values[16]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr11]"}, {25, "L", values[17]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr12]"}, {25, "L", values[18]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr13]"}, {25, "L", values[19]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr14]"}, {25, "L", values[20]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr15]"}, {25, "L", values[21]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr16]"}, {25, "L", values[22]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr17]"}, {25, "L", values[23]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr18]"}, {25, "L", values[24]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr19]"}, {25, "L", values[25]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr20]"}, {25, "L", values[26]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr21]"}, {25, "L", values[27]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr22]"}, {25, "L", values[28]}%},
    }
End//for
EndItem
Scroll list "VMT/FC/HDDV7" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
    Return()
endItem
Enddbox

Dbox"VMT_FC_HDDV8a" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {% {1, "L", "ID"}, {25, "L", values[1]}%},
        {% {1, "L", "RURAL_URBA"}, {25, "L", values[2]}%},
        {% {1, "L", "F_SYSTEM"}, {25, "L", values[3]}%},
        {% {1, "L", "SECTION_LE"}, {25, "L", values[4]}%},
        {% {1, "L", "SPEED_LIMI"}, {25, "L", values[5]}%},
        {% {1, "L", "F_Type"}, {25, "L", values[6]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr1]"}, {25, "L", values[7]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr2]"}, {25, "L", values[8]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr3]"}, {25, "L", values[9]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr4]"}, {25, "L", values[10]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr5]"}, {25, "L", values[11]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr6]"}, {25, "L", values[12]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr7]"}, {25, "L", values[13]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr8]"}, {25, "L", values[14]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr9]"}, {25, "L", values[15]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr10]"}, {25, "L", values[16]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr11]"}, {25, "L", values[17]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr12]"}, {25, "L", values[18]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr13]"}, {25, "L", values[19]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr14]"}, {25, "L", values[20]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr15]"}, {25, "L", values[21]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr16]"}, {25, "L", values[22]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr17]"}, {25, "L", values[23]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr18]"}, {25, "L", values[24]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr19]"}, {25, "L", values[25]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr20]"}, {25, "L", values[26]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr21]"}, {25, "L", values[27]}%},
        {% {1, "L", "[VMT/FC/HDDV8a/Hr22]"}, {25, "L", values[28]}%},
    }
End//for
EndItem
Scroll list "VMT/FC/HDDV7" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
    Return()
endItem
Enddbox
{{1, "L", "[VMT/FC/HDDV8a/Hr23]"}, {25, "L", values[29]}}
{{1, "L", "[VMT/FC/HDDV8a/Hr24]"}, {25, "L", values[30]}}

End//for
EndItem
Scroll list "VMT/FC/HDDV8a" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enndbox

Dbox"VMT_FC_HDDV8b" (record_values3)
Init do
    dim values[record_values3.length]
    for i=1 to values.length do
        values[i] = record_values3[i][2]
    endfor
    new_array = {
        {{1, "L", "ID"}, {25, "L", values[1]}},
        {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
        {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
        {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
        {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
        {{1, "L", "F_Type"}, {25, "L", values[6]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr1]"}, {25, "L", values[7]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr2]"}, {25, "L", values[8]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr3]"}, {25, "L", values[9]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr4]"}, {25, "L", values[10]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr5]"}, {25, "L", values[11]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr6]"}, {25, "L", values[12]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr7]"}, {25, "L", values[13]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr8]"}, {25, "L", values[14]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr9]"}, {25, "L", values[15]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr10]"}, {25, "L", values[16]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr11]"}, {25, "L", values[17]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr12]"}, {25, "L", values[18]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr13]"}, {25, "L", values[19]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr14]"}, {25, "L", values[20]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr15]"}, {25, "L", values[21]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr16]"}, {25, "L", values[22]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr17]"}, {25, "L", values[23]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr18]"}, {25, "L", values[24]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr19]"}, {25, "L", values[25]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr20]"}, {25, "L", values[26]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr21]"}, {25, "L", values[27]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr22]"}, {25, "L", values[28]}},
        {{1, "L", "[VMT/FC/HDDV8b/Hr23]"}, {25, "L", values[29]}},
{
  {{1, "L", "[VMT/FC/HDDV8b/Hr24]"}, {25, "L", values[30]}}
}
End//for
EndItem
Scroll list "VMT/FC/HDDV8b" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enddbox

Dbox"VMT_FC_MC" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/MC/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/MC/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/MC/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/MC/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/MC/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/MC/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/MC/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/MC/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/MC/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/MC/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/MC/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/MC/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/MC/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/MC/Hr14]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/MC/Hr15]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/MC/Hr16]"}, {25, "L", values[22]}},
    {{1, "L", "[VMT/FC/MC/Hr17]"}, {25, "L", values[23]}},
    {{1, "L", "[VMT/FC/MC/Hr18]"}, {25, "L", values[24]}},
    {{1, "L", "[VMT/FC/MC/Hr19]"}, {25, "L", values[25]}},
    {{1, "L", "[VMT/FC/MC/Hr20]"}, {25, "L", values[26]}},
    {{1, "L", "[VMT/FC/MC/Hr21]"}, {25, "L", values[27]}},
    {{1, "L", "[VMT/FC/MC/Hr22]"}, {25, "L", values[28]}},
    {{1, "L", "[VMT/FC/MC/Hr23]"}, {25, "L", values[29]}},
    {{1, "L", "[VMT/FC/MC/Hr24]"}, {25, "L", values[30]}}
}
End//for
EndItem
Scroll list "VMT/FC/MC" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enddbox

Dbox"VMT_FC_HDGB" (record_values3)
Init do
dim values[record_values3.length]
for i=1 to values.length do
values[i] = record_values3[i][2]
new_array = {
  {1, "L", "ID"}, {25, "L", values[1]},
  {1, "L", "RURAL_URBA"}, {25, "L", values[2]},
  {1, "L", "F_SYSTEM"}, {25, "L", values[3]},
  {1, "L", "SECTION_LE"}, {25, "L", values[4]},
  {1, "L", "SPEED_LIMI"}, {25, "L", values[5]},
  {1, "L", "F_Type"}, {25, "L", values[6]},
  {1, "L", "[VMT/FC/HDBG/Hr1]"}, {25, "L", values[7]},
  {1, "L", "[VMT/FC/HDBG/Hr2]"}, {25, "L", values[8]},
  {1, "L", "[VMT/FC/HDBG/Hr3]"}, {25, "L", values[9]},
  {1, "L", "[VMT/FC/HDBG/Hr4]"}, {25, "L", values[10]},
  {1, "L", "[VMT/FC/HDBG/Hr5]"}, {25, "L", values[11]},
  {1, "L", "[VMT/FC/HDBG/Hr6]"}, {25, "L", values[12]},
  {1, "L", "[VMT/FC/HDBG/Hr7]"}, {25, "L", values[13]},
  {1, "L", "[VMT/FC/HDBG/Hr8]"}, {25, "L", values[14]},
  {1, "L", "[VMT/FC/HDBG/Hr9]"}, {25, "L", values[15]},
  {1, "L", "[VMT/FC/HDBG/Hr10]"}, {25, "L", values[16]},
  {1, "L", "[VMT/FC/HDBG/Hr11]"}, {25, "L", values[17]},
  {1, "L", "[VMT/FC/HDBG/Hr12]"}, {25, "L", values[18]},
  {1, "L", "[VMT/FC/HDBG/Hr13]"}, {25, "L", values[19]},
  {1, "L", "[VMT/FC/HDBG/Hr14]"}, {25, "L", values[20]},
  {1, "L", "[VMT/FC/HDBG/Hr15]"}, {25, "L", values[21]},
  {1, "L", "[VMT/FC/HDBG/Hr16]"}, {25, "L", values[22]},
  {1, "L", "[VMT/FC/HDBG/Hr17]"}, {25, "L", values[23]},
  {1, "L", "[VMT/FC/HDBG/Hr18]"}, {25, "L", values[24]},
  {1, "L", "[VMT/FC/HDBG/Hr19]"}, {25, "L", values[25]},
  {1, "L", "[VMT/FC/HDBG/Hr20]"}, {25, "L", values[26]},
  {1, "L", "[VMT/FC/HDBG/Hr21]"}, {25, "L", values[27]},
  {1, "L", "[VMT/FC/HDBG/Hr22]"}, {25, "L", values[28]},
  {1, "L", "[VMT/FC/HDBG/Hr23]"}, {25, "L", values[29]},
  {1, "L", "[VMT/FC/HDBG/Hr24]"}, {25, "L", values[30]}
}
End//for
Scroll list "VMT/FC/HDGB" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
  Return()
endItem
Enddbox

Dbox"VMT_FC_HDDBT" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
  new_array = {
    {{1, "L", "ID"}, {25, "L", values[1]}},
    {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
    {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
    {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
    {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
    {{1, "L", "F_Type"}, {25, "L", values[6]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr1]"}, {25, "L", values[7]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr2]"}, {25, "L", values[8]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr3]"}, {25, "L", values[9]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr4]"}, {25, "L", values[10]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr5]"}, {25, "L", values[11]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr6]"}, {25, "L", values[12]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr7]"}, {25, "L", values[13]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr8]"}, {25, "L", values[14]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr9]"}, {25, "L", values[15]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr10]"}, {25, "L", values[16]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr11]"}, {25, "L", values[17]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr12]"}, {25, "L", values[18]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr13]"}, {25, "L", values[19]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr14]"}, {25, "L", values[20]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr15]"}, {25, "L", values[21]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr16]"}, {25, "L", values[22]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr17]"}, {25, "L", values[23]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr18]"}, {25, "L", values[24]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr19]"}, {25, "L", values[25]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr20]"}, {25, "L", values[26]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr21]"}, {25, "L", values[27]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr22]"}, {25, "L", values[28]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr23]"}, {25, "L", values[29]}},
    {{1, "L", "[VMT/FC/HDDBT/Hr24]"}, {25, "L", values[30]}}
  }
End//for
EndItem
Scroll list "VMT/FC/HDDBT" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
Enddbox

Dbox "VMT_FC_HDDBS" (record_values3)
Init do
  dim values[record_values3.length]
  for i=1 to values.length do
    values[i] = record_values3[i][2]
    new_array = {
      {{1, "L", "ID"}, {25, "L", values[1]}},
      {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
      {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
      {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
      {{1, "L", "SPEED_LIMI"}, {25, "L", values[5]}},
      {{1, "L", "F_Type"}, {25, "L", values[6]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr1]"}, {25, "L", values[7]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr2]"}, {25, "L", values[8]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr3]"}, {25, "L", values[9]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr4]"}, {25, "L", values[10]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr5]"}, {25, "L", values[11]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr6]"}, {25, "L", values[12]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr7]"}, {25, "L", values[13]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr8]"}, {25, "L", values[14]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr9]"}, {25, "L", values[15]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr10]"}, {25, "L", values[16]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr11]"}, {25, "L", values[17]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr12]"}, {25, "L", values[18]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr13]"}, {25, "L", values[19]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr14]"}, {25, "L", values[20]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr15]"}, {25, "L", values[21]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr16]"}, {25, "L", values[22]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr17]"}, {25, "L", values[23]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr18]"}, {25, "L", values[24]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr19]"}, {25, "L", values[25]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr20]"}, {25, "L", values[26]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr21]"}, {25, "L", values[27]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr22]"}, {25, "L", values[28]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr23]"}, {25, "L", values[29]}},
      {{1, "L", "[VMT/FC/HDDBS/Hr24]"}, {25, "L", values[30]}}
    }
  End//for
endItem
Dbox "VMT_FC_LDDT34" (record_values3)
Init do
  dim values[record_values3.length]
  for i = 1 to values.length do
    values[i] = record_values3[i][2]
    new_array = {
      {{1, "L", "ID"}, {25, "L", values[1]}},
      {{1, "L", "RURAL_URBA"}, {25, "L", values[2]}},
      {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
      {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
      {{1, "L", "SECTION_LE"}, {25, "L", values[4]}},
      {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
      {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
      {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
      {{1, "L", "F_SYSTEM"}, {25, "L", values[3]}},
      {{1, "L", "F_TYPE"}, {25, "L", values[6]}}
    }
  }
  return
End Item
Scroll list "VMT_FC/LDDT34" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
EndDbox

Dbox"VMT_Mix" (record_values1)
Init do
    dim values[record_values1.length]
    for i=1 to values.length do
        values[i] = record_values1[i][2]
    new_array = {
        {1, "L", "ID"}, {20, "L", values[1]},
        {1, "L", "RURAL_URBA"}, {20, "L", values[2]},
        {1, "L", "F_SYSTEM"}, {20, "L", values[3]},
        {1, "L", "SECTION_LENGTH"}, {20, "L", values[4]},
        {1, "L", "SPEED_LIMIT"}, {20, "L", values[5]},
        {1, "L", "F_TYPE"}, {20, "L", values[6]},
        {1, "L", "[VMTmix/LDV]"}, {20, "L", values[7]},
        {1, "L", "[VMTmix/LDT1]"}, {20, "L", values[8]},
        {1, "L", "[VMTmix/LDT2]"}, {20, "L", values[9]},
        {1, "L", "[VMTmix/LDT3]"}, {20, "L", values[10]},
        {1, "L", "[VMTmix/LDT4]"}, {20, "L", values[11]},
        {1, "L", "[VMTmix/HDV2B]"}, {20, "L", values[12]},
        {1, "L", "[VMTmix/HDV3]"}, {20, "L", values[13]},
        {1, "L", "[VMTmix/HDV4]"}, {20, "L", values[14]},
        {1, "L", "[VMTmix/HDV5]"}, {20, "L", values[15]},
        {1, "L", "[VMTmix/HDV6]"}, {20, "L", values[16]},
        {1, "L", "[VMTmix/HDV7]"}, {20, "L", values[17]},
        {1, "L", "[VMTmix/HDV8A]"}, {20, "L", values[18]},
        {1, "L", "[VMTmix/HDV8B]"}, {20, "L", values[19]},
        {1, "L", "[VMTmix/HDV8BS]"}, {20, "L", values[20]},
        {1, "L", "[VMTmix/HDVBT]"}, {20, "L", values[21]},
        {1, "L", "[VMTmix/MC]"}, {20, "L", values[22]}
    End//for
endItem
Scroll list "VMTMix" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
endItem
EndDbox

Dbox"VMT_Speed" (record_values2)
Init do
    dim values[record_values2.length]
for i=1 to values.length do
values[i] = record_values2[i][2]
new_array = {
  {{1, "L", "ID"}, {20, "L", values[1]}},
  {{1, "L", "RURAL_URBA"}, {20, "L", values[2]}},
  {{1, "L", "F_SYSTEM"}, {20, "L", values[3]}},
  {{1, "L", "SECTION_LE"}, {20, "L", values[4]}},
  {{1, "L", "SPEED_LIMI"}, {20, "L", values[5]}},
  {{1, "L", "F_Type"}, {20, "L", values[6]}},
  {{1, "L", "[SpeedVMT/Hr1]"}, {20, "L", values[7]}},
  {{1, "L", "[SpeedVMT/Hr2]"}, {20, "L", values[8]}},
  {{1, "L", "[SpeedVMT/Hr3]"}, {20, "L", values[9]}},
  {{1, "L", "[SpeedVMT/Hr4]"}, {20, "L", values[10]}},
  {{1, "L", "[SpeedVMT/Hr5]"}, {20, "L", values[11]}},
  {{1, "L", "[SpeedVMT/Hr6]"}, {20, "L", values[12]}},
  {{1, "L", "[SpeedVMT/Hr7]"}, {20, "L", values[13]}},
  {{1, "L", "[SpeedVMT/Hr8]"}, {20, "L", values[14]}},
  {{1, "L", "[SpeedVMT/Hr9]"}, {20, "L", values[15]}},
  {{1, "L", "[SpeedVMT/Hr10]"}, {20, "L", values[16]}},
  {{1, "L", "[SpeedVMT/Hr11]"}, {20, "L", values[17]}},
  {{1, "L", "[SpeedVMT/Hr12]"}, {20, "L", values[18]}},
  {{1, "L", "[SpeedVMT/Hr13]"}, {20, "L", values[19]}},
  {{1, "L", "[SpeedVMT/Hr14]"}, {20, "L", values[20]}},
  {{1, "L", "[SpeedVMT/Hr15]"}, {20, "L", values[21]}},
  {{1, "L", "[SpeedVMT/Hr16]"}, {20, "L", values[22]}},
  {{1, "L", "[SpeedVMT/Hr17]"}, {20, "L", values[23]}},
  {{1, "L", "[SpeedVMT/Hr18]"}, {20, "L", values[24]}},
  {{1, "L", "[SpeedVMT/Hr19]"}, {20, "L", values[25]}},
  {{1, "L", "[SpeedVMT/Hr20]"}, {20, "L", values[26]}},
  {{1, "L", "[SpeedVMT/Hr21]"}, {20, "L", values[27]}},
  {{1, "L", "[SpeedVMT/Hr22]"}, {20, "L", values[28]}},
  {{1, "L", "[SpeedVMT/Hr23]"}, {20, "L", values[29]}},
  {{1, "L", "[SpeedVMT/Hr24]"}, {20, "L", values[30]}}
}
End//for
EndItem
Scroll list "VMTSpeed" 3,3,30,30
List: new_array
Button "OK" 40.75, 10, 10 Cancel do
Return()
EndItem
EndDbox
Vita

Srividya Vadlamani was born in the city of Visakhapatnam, India. She completed her schooling at Belmont School. She did her bachelor’s at Gandhi Institute of Technology and Management (affiliated to Andhra University), majoring in civil engineering. She graduated with first class in 2002. She joined the Department of Civil and Environmental Engineering to do her master’s in August 2002. She will be graduating with the degree of Master of Science in Civil Engineering in May 2005.