Gainesville Urbanized Area
Year 2040 Long Range Transportation Plan Update

Technical Report No. 4

Model Update and Validation Report

Prepared for:
Metropolitan Transportation Planning Organization for the Gainesville Urbanized Area
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update

Technical Report No. 4:

2010 Model Update and Validation

Prepared for

Metropolitan Transportation Planning Organization for the Gainesville Urbanized Area

Prepared by

HDR

The preparation of this report has been financed in part through grants from the Federal Highway Administration, Federal Transit Administration and U.S. Department of Transportation, under the State Planning and Research Program, Section 505 (or Metropolitan Planning Program, Section 104 (f) of Title 23, U.S. Code. The contents of this report do not necessarily reflect the official views or policy of the U.S. Department of Transportation.
Table of Contents

1 Introduction............................................................................................................................. 5

2 External Trips.......................................................................................................................... 6
   2.1 Internal-External and External-External Trips................................................................. 6
   2.2 External Validation Adjustments.................................................................................... 9
   2.3 External Validation Results......................................................................................... 9

3 Trip Generation ..................................................................................................................... 10
   3.1 Trip Generation Process ................................................................................................. 11
   3.2 Socioeconomic Data and Traffic Analysis Zone Structure ............................................. 14
   3.3 Special Generators ......................................................................................................... 20
   3.4 Trip Generation Validation Results ............................................................................. 20

4 Trip Distribution .................................................................................................................. 24
   4.1 Trip Distribution Model Structure ................................................................................ 25
   4.2 Trip Distribution Model Development and Validation .................................................. 27
   4.3 Trip Distribution Validation Results ............................................................................ 28

5 Transit Accessibility and Path-Building .............................................................................. 34
   5.1 Transit Access and Path-Building Module Structure ..................................................... 35
   5.2 Transit Access and Path-Building Model Development and Validation ....................... 35
   5.3 Transit Access and Path-Building Model Validation Results ......................................... 37

6 Mode Choice......................................................................................................................... 38
   6.1 Mode Choice Model Structure ....................................................................................... 39
   6.2 Development and Validation of Mode Choice Model .................................................... 40
   6.3 Mode Choice Model Results ........................................................................................ 40

7 Highway Assignment ............................................................................................................ 43
   7.1 Highway Assignment Procedure .................................................................................. 44
   7.2 Highway Assignment Validation Results ...................................................................... 44

8 Transit Procedure and Assignment ..................................................................................... 49
   8.1 Transit Assignment Validation Results ........................................................................ 52

9 Summary and Conclusions .................................................................................................. 56
Appendix A: Socioeconomic Data Format ................................................................. 57
Appendix B: Turn Penalties (TURN.PEN) .................................................................. 59
Appendix C: Friction Factors .................................................................................... 61
Appendix D: Model Flowchart, Scripts and File Locations ........................................ 65
Appendix E: Input and Output Network Format ....................................................... 226
Appendix F: Glossary and Abbreviations ................................................................. 230
Appendix G: Scenario Manager / Running the Model ................................................ 235
List of Tables

Table 2.1 Internal-External and External-External Percentage Splits.................................................. 8
Table 3.1 Trip Production Rates............................................................................................................. 12
Table 3.2 Trip Attraction Rates ................................................................................................................ 13
Table 3.3 Dwelling Unit Weights .............................................................................................................. 13
Table 3.4 Year 2010 Socioeconomic Data Summary .................................................................................. 16
Table 3.5 Special Generators ..................................................................................................................... 20
Table 3.6 Percent Trips by Purpose ......................................................................................................... 21
Table 3.7 Trip Generation Aggregate Rates ............................................................................................. 21
Table 4.1 Terminal Times ............................................................................................................................ 26
Table 4.2 Average Trip Lengths (in Minutes) ........................................................................................... 29
Table 4.3 Intrazonal Trip Summary ........................................................................................................... 29
Table 5.1 Pedestrian Environment Variables ............................................................................................. 35
Table 6.1 Mode Choice Coefficients for 2010 Model ............................................................................... 40
Table 6.2 Mode Choice Validation Summary ........................................................................................... 41
Table 7.1 Volume to Count Performance by Category .............................................................................. 45
Table 7.2 Volume to Count Performance by Screenline ........................................................................... 46
Table 7.3 Root Mean Squared Error ........................................................................................................... 47
Table 8.1 Mode Choice Coefficient for Transit Variables ........................................................................... 50
Table 8.2 Year 2010 Transit Loading Estimates ......................................................................................... 54
List of Figures

Figure 2.1 External Station Locations ................................................................. 7
Figure 3.1 Trip Generation Model Chain ............................................................ 10
Figure 3.2 Study Area and Traffic Analysis Zone Structure ............................. 17
Figure 3.3 Year 2010 Population by Traffic Analysis Zone ............................... 18
Figure 3.4 Year 2010 Employment by Traffic Analysis Zone ........................... 19
Figure 3.5 Year 2010 Trip Productions by Traffic Analysis Zone .................... 22
Figure 3.6 Year 2010 Trip Attractions by Traffic Analysis Zone ....................... 23
Figure 4.1 Trip Distribution Model Chain ....................................................... 24
Figure 4.2 Desire Lines by Traffic Analysis Zone .............................................. 31
Figure 4.3 Trip Length Frequency Distribution for Home-Based Work ............... 32
Figure 4.4 Trip Length Frequency Distribution for Home-Based Shopping .......... 32
Figure 4.5 Trip Length Frequency Distribution for Home-Based Social/Recreational 33
Figure 4.6 Trip Length Frequency Distribution for Home-Based Other ............. 33
Figure 5.1 Transit Network Module ................................................................. 34
Figure 6.1 Mode Choice Model Chain ............................................................. 38
Figure 7.1 Highway Assignment Module ......................................................... 43
Figure 7.2 Volume to Count Scatter Plot ......................................................... 45
Figure 7.3 Gainesville Urbanized Area Transportation Study 2010 Model Screenlines 48
Figure 8.1 Transit Assignment Module ............................................................ 51
1 Introduction

This report documents the update and validation of the Gainesville Urbanized Area Transportation Study 2010 base year travel demand model conducted as integral part of the Gainesville Urbanized Area Year 2040 Long Range Transportation Plan update. The 2010 model study area covers all of Alachua County, including the nine municipalities within the county (cities of Alachua, Archer, Gainesville, Hawthorne, High Springs, Newberry, Waldo and towns of La Crosse and Micanopy). The effort involved updating the input files of the previous 2007 model to 2010 and reviewing and enhancing the model parameters and scripts for trip generation, distribution, highway and transit network development, mode choice, assignment and reporting steps. The Florida Department of Transportation’s FSUTMS-Cube Framework Phase II: Model Calibration and Validation Standards was followed as a technical guide for the validation effort.

The structure of the report generally follows the sequence of the processes in the model chain. It starts with documenting the update of the external trips in Section 2. Sections 3 and 4 cover trip generation and trip distribution. Section 5 documents transit accessibility and path building followed by mode choice in Section 6. Sections 7 and 8 document highway and transit assignments. Finally, Section 9 concludes the report with closing remarks.
2 External Trips

The external zones are placed along roadways entering and leaving the Gainesville Urbanized Area Transportation Study model region. The zones used in the 2010 base model are based on the 2007 model. There are 26 external zones outside of study area, and they are assigned unique Traffic Analysis Zone numbers from 600 to 625.

2.1 Internal-External and External-External Trips

External trips are vehicle trips having at least one trip end outside of the model study area. External trips include two categories: External-to-External trips and Internal-to-External trips or External-to-Internal trips. External-External trips are trips with both ends outside the study area, and trips with one end inside and one end outside the study area are called either Internal-External trips or External-Internal trips depending on the origin and destination.

External trips are generated or attracted at the external stations located outside of the model boundary. As described above, these stations are points of entry or exit of the study area. The external stations for the Gainesville Urbanized Area Transportation Study regional travel demand model are shown in Figure 2.1.

In the validation process of external trips, the External-External and External-Internal should ideally be estimated based on an external station origin-destination travel survey. In the absence of a recent origin-destination travel survey, the base year 2010 External-External and External-Internal trips were derived from the 2007 External-External and External-Internal trips factored by the 2010 traffic counts obtained from Florida Department of Transportation and Alachua County. There are two input files that influence external trips. These are the INTEXT and EETRIPS files with scenario year at the end of each file name. The percent of External-External and External-Internal trips were calculated from the 2007 model. However, because the external trips are derived from traffic counts without an origin-destination travel survey, it is necessary to make adjustments based on logic and local knowledge. These adjustments were conducted in an iterative manner in order to ensure that generated traffic volumes correspond to the 2010 traffic counts.
Figure 2.1 External Station Locations
The final 2010 External-External and External-Internal trips are shown in Table 2.1. Since the traffic patterns in Florida are heavily impacted by seasonality, the annual average daily traffic (AADT) is converted to peak-season weekday average daily traffic (PSWADT) through a model output conversion factor (MOCF) before coding into the model. The percentage of External-External and External-Internal are applied to the peak-season weekday average daily traffic for each external zone to calculate total External-External and External-Internal trips.

**Table 2.1 Internal-External and External-External Percentage Splits**

<table>
<thead>
<tr>
<th>External Traffic Analysis Zone</th>
<th>Total External Trips</th>
<th>Road Way</th>
<th>Total IE Trips</th>
<th>Total EE Trips</th>
<th>IE %</th>
<th>EE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>600</td>
<td>47,368</td>
<td>I-75 (North) at Columbia County Line</td>
<td>11,290</td>
<td>36,078</td>
<td>24%</td>
<td>76%</td>
</tr>
<tr>
<td>601</td>
<td>1,029</td>
<td>CR 241 (North) at Union County Line</td>
<td>699</td>
<td>330</td>
<td>68%</td>
<td>32%</td>
</tr>
<tr>
<td>602</td>
<td>5,446</td>
<td>SR 121 (North) at Union County Line</td>
<td>3,740</td>
<td>1,706</td>
<td>69%</td>
<td>31%</td>
</tr>
<tr>
<td>603</td>
<td>103</td>
<td>CR 237 (North) at Bradford County Line</td>
<td>89</td>
<td>14</td>
<td>86%</td>
<td>14%</td>
</tr>
<tr>
<td>604</td>
<td>2,783</td>
<td>SR 235 (North) at Bradford County Line</td>
<td>2,519</td>
<td>264</td>
<td>91%</td>
<td>9%</td>
</tr>
<tr>
<td>605</td>
<td>669</td>
<td>CR 1475 (North) at Bradford County Line</td>
<td>469</td>
<td>200</td>
<td>70%</td>
<td>30%</td>
</tr>
<tr>
<td>606</td>
<td>23,077</td>
<td>US 301 (North) at Bradford County Line</td>
<td>9,081</td>
<td>13,996</td>
<td>39%</td>
<td>61%</td>
</tr>
<tr>
<td>607</td>
<td>1,132</td>
<td>CR 325 (North) at Bradford County Line</td>
<td>776</td>
<td>356</td>
<td>69%</td>
<td>31%</td>
</tr>
<tr>
<td>608</td>
<td>6,392</td>
<td>SR 26 (East) at Putnam County Line</td>
<td>3,178</td>
<td>3,214</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>609</td>
<td>412</td>
<td>CR 1474 (East) at Putnam County Line</td>
<td>264</td>
<td>148</td>
<td>64%</td>
<td>36%</td>
</tr>
<tr>
<td>610</td>
<td>8,830</td>
<td>SR 20 (East) at Putnam County Line</td>
<td>4,386</td>
<td>4,444</td>
<td>50%</td>
<td>50%</td>
</tr>
<tr>
<td>611</td>
<td>11,251</td>
<td>US 301 (North) at Marion County Line</td>
<td>1,261</td>
<td>9,990</td>
<td>11%</td>
<td>89%</td>
</tr>
<tr>
<td>612</td>
<td>344</td>
<td>CR 225 (South) at Marion County Line</td>
<td>288</td>
<td>56</td>
<td>84%</td>
<td>16%</td>
</tr>
<tr>
<td>613</td>
<td>7,938</td>
<td>US 441 (South) at Marion County Line</td>
<td>7,090</td>
<td>848</td>
<td>89%</td>
<td>11%</td>
</tr>
<tr>
<td>614</td>
<td>48,947</td>
<td>I-75 (South) at Marion County Line</td>
<td>17,123</td>
<td>31,824</td>
<td>35%</td>
<td>65%</td>
</tr>
<tr>
<td>615</td>
<td>1,336</td>
<td>CR 234 (South) at Marion County Line</td>
<td>870</td>
<td>466</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>616</td>
<td>6,906</td>
<td>SR 121 (South) at Levy County Line</td>
<td>5,230</td>
<td>1,676</td>
<td>76%</td>
<td>24%</td>
</tr>
<tr>
<td>617</td>
<td>4,520</td>
<td>SR 45 (South) at Levy County Line</td>
<td>3,190</td>
<td>1,330</td>
<td>71%</td>
<td>29%</td>
</tr>
<tr>
<td>618</td>
<td>973</td>
<td>CR 241 (South) at Levy County Line</td>
<td>745</td>
<td>228</td>
<td>77%</td>
<td>23%</td>
</tr>
<tr>
<td>619</td>
<td>7,296</td>
<td>SR 24 (Southwest) at Levy County Line</td>
<td>5,190</td>
<td>2,106</td>
<td>71%</td>
<td>29%</td>
</tr>
<tr>
<td>620</td>
<td>1,233</td>
<td>CR 337 (South) at Levy County Line</td>
<td>885</td>
<td>348</td>
<td>72%</td>
<td>28%</td>
</tr>
<tr>
<td>621</td>
<td>9,999</td>
<td>SR 26 (West) at Gilchrist County Lin</td>
<td>7,639</td>
<td>2,360</td>
<td>76%</td>
<td>24%</td>
</tr>
<tr>
<td>622</td>
<td>2,052</td>
<td>CR 232 (West) at Gilchrist County Lin</td>
<td>1,474</td>
<td>578</td>
<td>72%</td>
<td>28%</td>
</tr>
<tr>
<td>623</td>
<td>3,742</td>
<td>NW 182 (West) at Gilchrist County Line</td>
<td>2,672</td>
<td>1,070</td>
<td>71%</td>
<td>29%</td>
</tr>
<tr>
<td>624</td>
<td>8,350</td>
<td>US 27 (Northwest) at Gilchrist County</td>
<td>5,988</td>
<td>2,362</td>
<td>72%</td>
<td>28%</td>
</tr>
<tr>
<td>625</td>
<td>6,062</td>
<td>US 441 (Northwest) at Columbia County Line</td>
<td>4,262</td>
<td>1,800</td>
<td>70%</td>
<td>30%</td>
</tr>
</tbody>
</table>
2.2 External Validation Adjustments

In the absence of a travel survey, several iterations of adjustments were conducted to achieve a goal of 1.0 volume-over-count ratio at each external station. For external zones 601, 603, 605, 607, 609, 615, 618, 622 and 623, 2010 annual average daily traffic was not available. Traffic count data from 2007 and 2011 were used to interpolate counts for 2010. For external station 612, which is located on a minor collector road, the count used in the 2007 model was utilized due to the absence of any nearby count station. Traffic counts at adjacent links were used with appropriate adjustments for the external zone 610.

An Iterative Proportional Fitting (IPF) process was used by creating a seed distribution and iteratively adjusting the cell values until a good match between the target row and column totals was achieved. For External-External trip tables, the row and column total targets represent the portions of the external station productions and attractions. For the External-Internal and Internal-External trips, productions and attractions are generated for both ends of the trip, the internal end of the trip and the external end of the trip. The traffic counts acted as control total at each external zone such that the total number of vehicle trips (Internal-External, External-Internal, and External-External) generated at an external station is equal to the traffic count at the external station.

2.3 External Validation Results

The final model validation results will be discussed in the highway assignment section. The assignment results indicate reasonable match between the external travel movements estimated by the model and the traffic counts. The external cordon line achieves a volume-over-count ratio of 1.00. Corridors leading to or nearby external zones also were validated to satisfactory levels, based on this iterative adjustment process.
3 Trip Generation

Trip Generation is the first step of the four-step travel demand forecasting process. In this step, the amount of travel by trip purposes are calculated. Trip generation focuses on the locations generating the travel, and not the directionality of travel. Trip generation models provide the estimates of the number of trips by purpose produced by (trip production) or attracted to (trip attraction) a traffic analysis zone as a function of the demographic, socioeconomic, and land use characteristics of the zone. The most common forms of trip production and attraction models are cross-classification and linear regression. Cross-classification is generally used for trip production models and regression is used for trip attraction models. Figure 3.1 shows the trip generation model chain.

Figure 3.1 Trip Generation Model Chain
3.1 Trip Generation Process

The Alachua County 2010 model uses cross classification trip production rates stratified by auto availability (0, 1, 2, and 3+ auto Households), dwelling unit type (single-family, multifamily, and transient units), and household size (1, 2, 3, 4, and 5+ persons per household). Trip production rates for home-based work, home-based shop, home-based social/recreation, and home-based other purposes are shown in Table 3.1. The original source for these trip production rates was the North Florida Household Travel Survey, consistent with the previous 2007 Alachua County model. These trip production rates were not updated since there is no updated household travel survey available at this time.

Trip attraction rates were originally derived from the 2005 Northeast Florida Regional Planning Model (NERPM). As the Gainesville Urbanized Area Transportation Study model and the Northeast Regional Planning Model are from Northeast Florida, both model regions share similar socioeconomic characteristics, such as less of a reliance on tourism and seasonal residents than other parts of Florida. The trip attraction rates and dwelling unit (DU) weights are kept the same as the 2007 Gainesville Urbanized Area Transportation Study model, as shown in Table 3.2 and Table 3.3, respectively. Again, these were kept unchanged since an updated source of trip attraction rates is not available. The 2010 small area demographic were not available at the time that this study was completed, so the dwelling unit weights were retained from the 2007 model.

The Alachua County 2010 model uses 11 trip purposes:

1. Home-based work;
2. Home-based shop;
3. Home-based social/recreation;
4. Home-based other (Home-based non-work, excluding university trips);
5. Non-home-based;
6. Home-based university;
7. UF campus/dorm;
8. Four-tire truck;
9. Single-unit truck;
10. Tractor-trailer; and
11. Internal-external.
### Table 3.1 Trip Production Rates

#### Home-Based Work

<table>
<thead>
<tr>
<th>Dwelling Unit Type</th>
<th>Number of Autos Available</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Family</td>
<td>0</td>
<td>0.35</td>
<td>0.64</td>
<td>1.01</td>
<td>1.5</td>
<td>2.08</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.69</td>
<td>0.98</td>
<td>1.35</td>
<td>1.84</td>
<td>2.42</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.35</td>
<td>1.64</td>
<td>2.01</td>
<td>2.5</td>
<td>3.08</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>1.76</td>
<td>2.05</td>
<td>2.42</td>
<td>2.9</td>
<td>3.49</td>
</tr>
<tr>
<td>Multifamily</td>
<td>0</td>
<td>0.41</td>
<td>0.7</td>
<td>1.01</td>
<td>1.31</td>
<td>1.62</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.95</td>
<td>1.49</td>
<td>2.02</td>
<td>2.56</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.65</td>
<td>2.3</td>
<td>2.95</td>
<td>3.6</td>
<td>4.25</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>2.21</td>
<td>2.89</td>
<td>3.59</td>
<td>4.27</td>
<td>4.96</td>
</tr>
<tr>
<td>Hotel/Motel</td>
<td></td>
<td>1.04</td>
<td>0.72</td>
<td>0.5</td>
<td>0.39</td>
<td>0.39</td>
</tr>
</tbody>
</table>

#### Home-Based Shopping

<table>
<thead>
<tr>
<th>Dwelling Unit Type</th>
<th>Number of Autos Available</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Family</td>
<td>0</td>
<td>0.4</td>
<td>0.65</td>
<td>1.36</td>
<td>1.62</td>
<td>2.07</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.98</td>
<td>1.35</td>
<td>1.84</td>
<td>2.42</td>
<td>2.89</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.5</td>
<td>2.01</td>
<td>2.5</td>
<td>3.08</td>
<td>3.59</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>2.05</td>
<td>2.42</td>
<td>2.9</td>
<td>3.49</td>
<td>4.01</td>
</tr>
<tr>
<td>Multifamily</td>
<td>0</td>
<td>0.41</td>
<td>0.70</td>
<td>1.01</td>
<td>1.31</td>
<td>1.62</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.95</td>
<td>1.49</td>
<td>2.02</td>
<td>2.56</td>
<td>3.1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.65</td>
<td>2.30</td>
<td>2.95</td>
<td>3.6</td>
<td>4.25</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>2.21</td>
<td>2.89</td>
<td>3.59</td>
<td>4.27</td>
<td>4.96</td>
</tr>
<tr>
<td>Hotel/Motel</td>
<td></td>
<td>1.04</td>
<td>0.72</td>
<td>0.5</td>
<td>0.39</td>
<td>0.39</td>
</tr>
</tbody>
</table>

#### Home-Based Social/Recreational

<table>
<thead>
<tr>
<th>Dwelling Unit Type</th>
<th>Number of Autos Available</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Family</td>
<td>0</td>
<td>0.21</td>
<td>0.28</td>
<td>1.28</td>
<td>1.47</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.48</td>
<td>0.85</td>
<td>1.43</td>
<td>1.31</td>
<td>2.37</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.53</td>
<td>0.89</td>
<td>1.85</td>
<td>2.07</td>
<td>2.77</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>0.7</td>
<td>1.07</td>
<td>2.04</td>
<td>2.24</td>
<td>2.97</td>
</tr>
<tr>
<td>Multifamily</td>
<td>0</td>
<td>0.18</td>
<td>0.63</td>
<td>1.08</td>
<td>1.53</td>
<td>1.98</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.22</td>
<td>0.67</td>
<td>1.12</td>
<td>1.57</td>
<td>2.02</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.64</td>
<td>1.09</td>
<td>1.54</td>
<td>1.99</td>
<td>2.44</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>0.84</td>
<td>1.29</td>
<td>1.74</td>
<td>2.19</td>
<td>2.64</td>
</tr>
<tr>
<td>Hotel/Motel</td>
<td></td>
<td>0.66</td>
<td>1.81</td>
<td>2.97</td>
<td>4.29</td>
<td>6.49</td>
</tr>
</tbody>
</table>

#### Home-Based Other

<table>
<thead>
<tr>
<th>Dwelling Unit Type</th>
<th>Number of Autos Available</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Family</td>
<td>0</td>
<td>0.29</td>
<td>0.64</td>
<td>1.67</td>
<td>3.38</td>
<td>5.78</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.48</td>
<td>1.29</td>
<td>2.59</td>
<td>4.38</td>
<td>6.67</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.62</td>
<td>1.79</td>
<td>3.34</td>
<td>5.2</td>
<td>7.33</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>0.68</td>
<td>1.94</td>
<td>3.58</td>
<td>5.59</td>
<td>7.99</td>
</tr>
<tr>
<td>Multifamily</td>
<td>0</td>
<td>0.35</td>
<td>0.78</td>
<td>2.28</td>
<td>4</td>
<td>6.23</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>0.74</td>
<td>1.36</td>
<td>3.16</td>
<td>4.92</td>
<td>6.91</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.12</td>
<td>1.87</td>
<td>3.71</td>
<td>5.59</td>
<td>7.34</td>
</tr>
<tr>
<td></td>
<td>3+</td>
<td>1.17</td>
<td>2.09</td>
<td>4.05</td>
<td>5.75</td>
<td>7.56</td>
</tr>
<tr>
<td>Hotel/Motel</td>
<td></td>
<td>0.55</td>
<td>1.32</td>
<td>2.31</td>
<td>3.63</td>
<td>4.84</td>
</tr>
</tbody>
</table>
Table 3.2 Trip Attraction Rates

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Manufacturing</th>
<th>Other Industrial</th>
<th>Commercial</th>
<th>Service</th>
<th>Total</th>
<th>Dwelling Units</th>
<th>School Enrollment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home-Based Work</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.8</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>Home-Based Shopping</td>
<td>0</td>
<td>0</td>
<td>6.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Home-Based Social/Recreational</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1.61</td>
<td>0</td>
</tr>
<tr>
<td>Home-Based Other</td>
<td>0</td>
<td>0</td>
<td>1.5</td>
<td>1.5</td>
<td>0</td>
<td>0.3</td>
<td>1.5</td>
</tr>
<tr>
<td>Non-home-Based</td>
<td>0</td>
<td>0</td>
<td>3.54</td>
<td>1.71</td>
<td>0</td>
<td>0.3</td>
<td>0</td>
</tr>
<tr>
<td>Four-Tire Truck</td>
<td>0.47</td>
<td>0.55</td>
<td>0.45</td>
<td>0.22</td>
<td>0</td>
<td>0.13</td>
<td>0</td>
</tr>
<tr>
<td>Single-Unit Truck</td>
<td>0.12</td>
<td>0.15</td>
<td>0.13</td>
<td>0.04</td>
<td>0</td>
<td>0.05</td>
<td>0</td>
</tr>
<tr>
<td>Tractor-Trailer</td>
<td>0.05</td>
<td>0.09</td>
<td>0.04</td>
<td>0.01</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.3 Dwelling Unit Weights

<table>
<thead>
<tr>
<th>Average Persons Per Dwelling Unit</th>
<th>One-Person Households</th>
<th>Two-Person Households</th>
<th>Three-Person Households</th>
<th>Four-Person Households</th>
<th>Five-Person Households</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00-1.12</td>
<td>0.89</td>
<td>0.11</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.13-1.37</td>
<td>0.76</td>
<td>0.22</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.38-1.62</td>
<td>0.59</td>
<td>0.34</td>
<td>0.05</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td>1.63-1.87</td>
<td>0.46</td>
<td>0.34</td>
<td>0.11</td>
<td>0.06</td>
<td>0.03</td>
</tr>
<tr>
<td>1.88-2.12</td>
<td>0.32</td>
<td>0.36</td>
<td>0.16</td>
<td>0.11</td>
<td>0.05</td>
</tr>
<tr>
<td>2.13-2.37</td>
<td>0.24</td>
<td>0.36</td>
<td>0.18</td>
<td>0.14</td>
<td>0.08</td>
</tr>
<tr>
<td>2.38-2.62</td>
<td>0.21</td>
<td>0.33</td>
<td>0.19</td>
<td>0.16</td>
<td>0.12</td>
</tr>
<tr>
<td>2.62-2.87</td>
<td>0.12</td>
<td>0.35</td>
<td>0.19</td>
<td>0.23</td>
<td>0.11</td>
</tr>
<tr>
<td>2.88-3.12</td>
<td>0.13</td>
<td>0.34</td>
<td>0.18</td>
<td>0.16</td>
<td>0.19</td>
</tr>
<tr>
<td>3.13-3.37</td>
<td>0.12</td>
<td>0.29</td>
<td>0.18</td>
<td>0.17</td>
<td>0.24</td>
</tr>
<tr>
<td>3.38-3.62</td>
<td>0.08</td>
<td>0.24</td>
<td>0.2</td>
<td>0.2</td>
<td>0.28</td>
</tr>
<tr>
<td>3.63-3.87</td>
<td>0.05</td>
<td>0.2</td>
<td>0.19</td>
<td>0.23</td>
<td>0.33</td>
</tr>
<tr>
<td>3.88-4.12</td>
<td>0.04</td>
<td>0.16</td>
<td>0.17</td>
<td>0.24</td>
<td>0.39</td>
</tr>
<tr>
<td>4.13-4.37</td>
<td>0.02</td>
<td>0.15</td>
<td>0.14</td>
<td>0.21</td>
<td>0.48</td>
</tr>
<tr>
<td>4.38-4.62</td>
<td>0.01</td>
<td>0.15</td>
<td>0.13</td>
<td>0.17</td>
<td>0.54</td>
</tr>
<tr>
<td>4.63-5.99</td>
<td>0</td>
<td>0.05</td>
<td>0.07</td>
<td>0.14</td>
<td>0.74</td>
</tr>
<tr>
<td>6.00+</td>
<td>0</td>
<td>0</td>
<td>0.02</td>
<td>0.05</td>
<td>0.93</td>
</tr>
</tbody>
</table>

Home-based university and University of Florida Campus/Dorm trip purposes are unique to the Gainesville Urbanized Area Transportation Study model. These additional purposes also were used in the Gainesville Urbanized Area Transportation Study 2007 and 2000 models. It is important to capture these trips in a university town, such as the City of Gainesville. The home-based university purpose is for trips traveling from off-campus housing to parking spaces within
the University of Florida Campus. On the other hand, the University of Florida Campus/Dorm trip purpose is trips from University of Florida on-campus dormitories to classrooms that are specified in the ZONEDATA file. It should be noted that the model has limited capabilities in simulating parking capacity beyond the number of parking spaces being stored in the ZONEDATA file and used in the attraction equations.

Home-based university and Dorm trip production and attraction equations for the Home-based university and Dorm purposes are listed below, as extracted from the model scripts. During validation, these trip rates were relocated to the Cube catalog keys (names depicted in {brackets}) to enhance model transparency.

**Home-Based University Productions:**
RO.HBUP = (RATE_HBUP)*ZI.1.UF_OC_ST
; UF_OC_ST is off-campus (students)
; Default value of {RATE_HBUP} is 2.996

**Home-Based University Attractions:**
RO.HBUA = (RATE_HBUA)*ZI.1.UF_PARKING
; PARKING is UF Parking Spaces
; Default value of {RATE_HBUA} is 1.375

**UF Campus/Dorm Productions:**
RO.HDORMUP = (RATE_HDORMUP) *ZI.1.UF_DORM_ST
; UF_DORM_ST is Campus housing/Dormitory students
; Default value of {RATE_HDORMUP} is 2.262

**UF Campus/Dorm Attractions:**
RO.HDORMUA = (RATE_HDORMUA) *ZI.1.SEATS
; SEATS is UF Classroom Seats
; Default value of {RATE_HDORMUA} is 0.7513

### 3.2 Socioeconomic Data and Traffic Analysis Zone Structure

Socioeconomic data is a very important input to the trip generation process. During the last model update in 2007, the ZDATA files were converted to the new Florida Standard Urban Transportation Model Structure file formats to be consistent with the Cube-Voyager scripting. The Gainesville Urbanized Area Transportation Study 2000 model used the NERGEN FORTRAN program for trip generation and relied on a number of ASCII text file formats for input data. The base year 2007 model used Cube-Voyager scripting in place of NERGEN FORTRAN routines and used input files in a database, rather than text format. As part of the 2010 effort, staff from the Metropolitan Transportation Planning Organization for the Gainesville Urbanized Area updated the 2007 socioeconomic data files to the new base year 2010, which were reviewed and refined by the consultant team in coordination with staff. Special generator and external trip files were updated by the consultant team.
Like the Gainesville Urbanized Area Transportation Study 2007 model, the 2010 model study area covers the entirety of Alachua County, including all nine municipalities within the county. The zonal structure of the 2007 model was reviewed by the consultant team and no changes were deemed necessary. The 2010 model has a total of 586 traffic analysis zones; 560 internal zones and 26 external zones within and outside study area. Figure 3.2 shows the Gainesville Urbanized Area Transportation Study travel demand model geographic coverage and the traffic analysis zone structure of the study area.

As mentioned earlier, staff from the Gainesville Urbanized Area Transportation Study developed the 2010 socioeconomic dataset which includes information on population disaggregated by single family, multifamily, and hotel/motel units. It also provides information on auto availability, property vacancy rates, and seasonal use. On the employment side, the dataset contains information disaggregated by service, commercial, manufacturing, and other industrial sectors. It also contains information on school enrollment, university employment, dormitory students, and parking. The parking costs in the 2010 dataset were borrowed from the 2007 dataset based on discussions with staff from the Metropolitan Transportation Planning Organization for the Gainesville Urbanized Area.

Error! Reference source not found. and Figure 3.4 show the total population and employment of each traffic analysis zone within the regional travel demand model. The details of the SE data format are included in Appendix A of this report and the data is summarized in Table 3.4.
### Table 3.4 Year 2010 Socioeconomic Data Summary

<table>
<thead>
<tr>
<th>Socio-Economic data summary</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permanent Population</td>
<td>247,336</td>
</tr>
<tr>
<td>Total Population</td>
<td>251,951</td>
</tr>
<tr>
<td>Permanently Occupied Dwelling Units</td>
<td>99,089</td>
</tr>
<tr>
<td>Transient and Permanently Occupied Dwelling Units</td>
<td>101,996</td>
</tr>
<tr>
<td>Total Service Employment</td>
<td>91,399</td>
</tr>
<tr>
<td>Total Commercial Employment</td>
<td>32,669</td>
</tr>
<tr>
<td>Total Manufacturing Employment</td>
<td>4,048</td>
</tr>
<tr>
<td>Total Other Industrial Employment</td>
<td>9,478</td>
</tr>
<tr>
<td>Total Employment</td>
<td>137,594</td>
</tr>
<tr>
<td>Permanent Population per Permanently Occupied Dwelling Unit</td>
<td>2.50</td>
</tr>
<tr>
<td>Total Population per Total Occupied Dwelling Unit</td>
<td>2.47</td>
</tr>
<tr>
<td>Total Employment per Permanent Population</td>
<td>0.556</td>
</tr>
<tr>
<td>Service to Total Employment</td>
<td>0.664</td>
</tr>
<tr>
<td>Commercial to Total Employment</td>
<td>0.237</td>
</tr>
<tr>
<td>Manufacturing to Total Employment</td>
<td>0.029</td>
</tr>
<tr>
<td>Other Industrial to Total Employment</td>
<td>0.069</td>
</tr>
<tr>
<td>Internal Person Trips per Permanently Occupied Dwelling Unit</td>
<td>11.63</td>
</tr>
<tr>
<td>Internal Person Trips per Total Occupied Dwelling Units</td>
<td>11.29</td>
</tr>
<tr>
<td>Internal Person Trips per Employee</td>
<td>8.372</td>
</tr>
</tbody>
</table>
Figure 3.2 Study Area and Traffic Analysis Zone Structure
Figure 3.3 Year 2010 Population by Traffic Analysis Zone
Figure 3.4 Year 2010 Employment by Traffic Analysis Zone
3.3 Special Generators

There are certain traffic analysis zones where the standard trip generation algorithms did not produce sufficient trips due to the special land use characteristics of these zones. These traffic analysis zones are called special generators. It is best practice in travel demand forecasting to minimize the use of special generators. Special generators should only be used where validation discrepancies exist that cannot be corrected with edits to other model files and parameters. The special generators are shown in Table 3.5.

<table>
<thead>
<tr>
<th>TAZ</th>
<th>PA dummy</th>
<th>Addition or Subtraction</th>
<th>Person Trips</th>
<th>HBW %</th>
<th>HBSH %</th>
<th>HBSR %</th>
<th>HBO %</th>
<th>NHB %</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>536</td>
<td>A</td>
<td>+</td>
<td>27,000</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>92</td>
<td>2</td>
<td>Santa Fe College</td>
</tr>
<tr>
<td>440</td>
<td>A</td>
<td>+</td>
<td>655</td>
<td>20</td>
<td>38</td>
<td>38</td>
<td>0</td>
<td>4</td>
<td>University of Florida</td>
</tr>
<tr>
<td>441</td>
<td>A</td>
<td>+</td>
<td>576</td>
<td>20</td>
<td>38</td>
<td>38</td>
<td>0</td>
<td>4</td>
<td>University of Florida</td>
</tr>
<tr>
<td>443</td>
<td>A</td>
<td>+</td>
<td>408</td>
<td>20</td>
<td>38</td>
<td>38</td>
<td>0</td>
<td>4</td>
<td>University of Florida</td>
</tr>
<tr>
<td>449</td>
<td>A</td>
<td>+</td>
<td>662</td>
<td>20</td>
<td>38</td>
<td>38</td>
<td>0</td>
<td>4</td>
<td>University of Florida</td>
</tr>
<tr>
<td>453</td>
<td>A</td>
<td>+</td>
<td>1,816</td>
<td>20</td>
<td>38</td>
<td>38</td>
<td>0</td>
<td>4</td>
<td>University of Florida</td>
</tr>
<tr>
<td>460</td>
<td>A</td>
<td>+</td>
<td>362</td>
<td>20</td>
<td>38</td>
<td>38</td>
<td>0</td>
<td>4</td>
<td>University of Florida</td>
</tr>
</tbody>
</table>

TAZ = traffic analysis zone  
PA = production/attraction  
HBW = Home-based work  
HBSH = Home-based shopping  
HBSR = Home-based social-recreation  
HBO = Home-based other  
NHB = Non home-based

3.4 Trip Generation Validation Results
Throughout the validation process, trip generation statistics were summarized to assess model validity. Statistical comparisons were made between the Gainesville Urbanized Area Transportation Study 2007 and 2010 models as well as model validation standards from the FSUTMS-Cube Framework Phase II: Model Calibration and Validation Standards Final Report. The trip generation validation statistics is summarized in Table 3.6 and Table 3.7. As shown by the results, the trip generation estimated by the model is within acceptable ranges. Figure 3.5 and Figure 3.6 show the trip productions and attractions by traffic analysis zone.
Table 3.6 Percent Trips by Purpose

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Gainesville Urbanized Area Transportation Study 2010 Model</th>
<th>Gainesville Urbanized Area Transportation Study 2010 Model</th>
<th>FSUTMS*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Production</td>
<td>Percent by Productions</td>
<td>Unbalanced Attractions</td>
</tr>
<tr>
<td>Home-Based Work</td>
<td>185,994</td>
<td>13.87%</td>
<td>356,347</td>
</tr>
<tr>
<td>Home-Based Shopping</td>
<td>135,822</td>
<td>10.13%</td>
<td>201,523</td>
</tr>
<tr>
<td>Home-Based Social-Rec.</td>
<td>120,838</td>
<td>9.01%</td>
<td>259,957</td>
</tr>
<tr>
<td>Home-Based Other</td>
<td>270,822</td>
<td>20.20%</td>
<td>336,606</td>
</tr>
<tr>
<td>Non home-Based</td>
<td>354,086</td>
<td>26.40%</td>
<td>354,806</td>
</tr>
<tr>
<td>Home-Based University</td>
<td>64,423</td>
<td>4.80%</td>
<td>20,070</td>
</tr>
<tr>
<td>Dormitory-Based University</td>
<td>22,767</td>
<td>1.70%</td>
<td>21,288</td>
</tr>
<tr>
<td>Truck-Taxi</td>
<td>84,379</td>
<td>6.29%</td>
<td></td>
</tr>
<tr>
<td>Internal-External</td>
<td>101,877</td>
<td>7.60%</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>1,341,008</td>
<td>100.00%</td>
<td></td>
</tr>
</tbody>
</table>

*Table F.1 - FSUTMS-Cube Framework Phase II Model Calibration and Validation Standards

Table 3.7 Trip Generation Aggregate Rates

<table>
<thead>
<tr>
<th></th>
<th>Gainesville Urbanized Area Transportation Study 2010 Model</th>
<th>Gainesville Urbanized Area Transportation Study 2007 Model</th>
<th>Gainesville Urbanized Area Transportation Study 2000 Model</th>
<th>FSUTMS*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Persons/Dwelling Units</td>
<td>Employment/Person</td>
<td>Internal Person Trips/Dwelling Units</td>
<td>Internal Person Trips/Person</td>
</tr>
<tr>
<td></td>
<td>2.47</td>
<td>0.56</td>
<td>11.63</td>
<td>4.66</td>
</tr>
<tr>
<td></td>
<td>2.21</td>
<td>11.05</td>
<td>11.35</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>2.84</td>
<td>11.35</td>
<td>7.73</td>
<td>4.65</td>
</tr>
</tbody>
</table>

*Table F.1 - FSUTMS-Cube Framework Phase II Model Calibration and Validation Standards
Figure 3.5 Year 2010 Trip Productions by Traffic Analysis Zone
Figure 3.6 Year 2010 Trip Attractions by Traffic Analysis Zone
4 Trip Distribution

Trip distribution models link trip productions and attractions between pairs of traffic analysis zones. The most common method for trip distribution in four-step models is the gravity model. The trip distribution step distributes all trips produced in a zone to all possible attraction zones using a gravity model. The gravity model uses the trip productions and attractions estimated in the trip generation model and the network friction to distribute the trips. Friction factors are used to represent the effects of travel impedance. Friction factors can be computed by a formula, such as gamma function, or can be calibrated such that a modeled trip length frequency distribution matches an observed trip length frequency distribution. Trip distribution results in person trip matrices detailing the trips between each production and attraction traffic analysis zone. These person trips matrices are later processed in the model chain during mode choice to allocate trips by auto occupancy and transit categories. Figure 4.1 shows the trip distribution model chain.

Figure 4.1 Trip Distribution Model Chain

DISTRIBUTION
For evaluation of trip distribution, the *FHWA Travel Model Validation and Reasonability Checking Manual* recommends two general types of aggregate checks of trip distribution model results - trip length checks and origin-destination pattern checks. For this model, evaluation was accomplished by comparing statistics for average trip length and the percentage of intrazonal trips between the Gainesville Urbanized Area Transportation Study 2010 model, the Gainesville Urbanized Area Transportation Study 2007 model, and the Florida Standard Urban Transportation Model Structure standards. Additionally, desire line maps were prepared to show travel movements between zones and trip length frequency distributions were reviewed for reasonableness.

### 4.1 Trip Distribution Model Structure

The general distribution process includes the building of highway networks and travel time skims as well as application of the gravity model. The elements of these processes are described below.

#### Building Highway Networks

The Florida Standard Urban Transportation Model Structure includes a module known as “Highway Network” to process highway networks of model areas. As part of the model validation process, the consultant conducted an in depth review of the highway network. Network characteristics (number of lanes, area type, and facility type) were updated to reflect 2010 conditions of the roadway system throughout Alachua County.

#### Travel Time Skims

Free-flow travel time skims between zone pairs are developed as the last substep in the “Highway Network” module in the Florida Standard Urban Transportation Model Structure, including the updating of travel time skims with intrazonal and terminal times. Highway network characteristics are an input to this process. In addition to the highway network characteristics, other input files are generally used during network skimming. The first of these is the turn penalties file which contains a record of all prohibited turning movements in the network. Turning movements were reviewed to include any updated prohibited movements for year 2010 conditions during validation. The turn penalty file can also include time penalties; however, time penalties were not recommended in the model area as the highway assignment validated reasonably well without supplemental travel time factors. Also, a toll file is used in most Florida models to identify toll plaza characteristics. However, because no toll roads exist
in Alachua County, this file is not used in the Gainesville Urbanized Area Transportation Study 2010 model.

Intrazonal times represent the travel time it takes to travel within or across a zone. These times are calculated as one-half the average travel time from each zone to the two nearest adjacent zones. Terminal times represent the time required at either end of a trip to travel from an origin to a vehicle or from the vehicle to a final destination. More specifically, this accounts for the time necessary to walk to or from the vehicle used for any given trip. Terminal times are typically greatest in central business districts and lowest in residential areas. Table 4.1 lists the terminal times by area type used in the Gainesville Urbanized Area Transportation Study 2010 model.

### Table 4.1 Terminal Times

<table>
<thead>
<tr>
<th>Terminal Times*</th>
<th>Area Type</th>
<th>Area Type Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>12</td>
<td>Urbanized Area (under 500,000) Primary City Central Business District</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>Other Urbanized Area Central Business District and Small City Downtown</td>
</tr>
<tr>
<td>5</td>
<td>14</td>
<td>Non Urbanized Area Small City Downtown</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>Central Business District Fringe Areas</td>
</tr>
<tr>
<td>3</td>
<td>22</td>
<td>Industrial</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>Residential Area of Urbanized Areas</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>Undeveloped Portions of Urbanized Areas</td>
</tr>
<tr>
<td>1</td>
<td>33</td>
<td>Transitioning Areas/Urban Areas over 5,000 Population</td>
</tr>
<tr>
<td>2</td>
<td>42</td>
<td>Other Outlying Business District</td>
</tr>
<tr>
<td>1</td>
<td>51</td>
<td>Developed Rural Areas/Small Cities under 5,000 Population</td>
</tr>
<tr>
<td>1</td>
<td>52</td>
<td>Undeveloped Rural Areas</td>
</tr>
</tbody>
</table>

*Terminal Times listed in whole minutes

### Trip Distribution Module

The “DISTRIBUTION” module distributes trips between zones using a gravity model. The primary input data used for DISTRIBUTION are the friction factor (FF) file and a set of congested highway skims. These are used by the gravity model to measure the effects of spatial separation between zones for the purposes of trip distribution. It is generally assumed that productions are less likely to be linked to destinations with greater travel times if alternative destinations with lesser travel times and similar attractiveness are available. Friction factors from the Gainesville Urbanized Area Transportation Study 2007 model were used for the Gainesville Urbanized Area Transportation Study 2010 model without modification. Since no new household travel surveys were conducted since the 2007 model calibration and validation,
and the resulting trip distribution was reasonable, it was decided to maintain the existing friction factor set in the 2010 model.

Friction factors are used by the gravity model to link the trip productions and attractions generated by GENERATION. These trip interchanges denote person trips traveling from one zone in the model to another. Trips are distributed by the trip purposes found in the Gainesville Urbanized Area Transportation Study 2010 model. These person trips are later converted into vehicle trips during mode choice and then loaded onto networks during highway and transit assignment. The next subsection describes the checks, modifications, and adjustments made to trip distribution assumptions in order to verify and improve model validity.

4.2 Trip Distribution Model Development and Validation

Errors in the trip distribution phase can lead to significant problems in the execution of following steps in the model chain (i.e., mode choice and trip assignment). Consequently, efforts were taken to maximize the accuracy of the Gainesville Urbanized Area Transportation Study 2010 trip distribution module. These efforts included adjustments to network speeds and capacities and corrections of network link attributes.

**Speeds and Capacities**

The speed and capacity look up table is the same as the 2007 model; the main effort for validating the capacity was to update the number of lanes for the highway network. The Florida Department of Transportation Roadway Characteristics Inventory has been used as the primary source for updating the number of lanes to represent the 2010 conditions.

**Penalties and Prohibitions**

The TURN.pen file, formerly known as TCARDS, allows for the adjustment of travel times on specific links by either including a time penalty to pass from one link to another or by prohibiting the movement all together. Prohibitions are confined to ramps located along Interstate 75, mainly to guide trips to the correct ramps for each travel movement. No time penalties were added during the 2010 model validation effort. The TURN.pen file is listed in Appendix B.
Friction Factors

Friction factors are used in the gravity model to represent the effects of travel impedance. These factors define the measure of separation based on travel impedances between traffic analysis zones. The friction factors from the 2007 model were reviewed and no updates were necessary for the 2010 model validation. Analysis of the trip distribution summaries did not indicate a need for modification in order to improve trip distribution. Average trip lengths seemed reasonable, intrazonal percentages made sense, and aggregate trip distribution patterns looked logical. Furthermore, there are no updated household travel diary survey data for Alachua County to allow for calibration of new friction factors. A copy of the validated friction factor file (FF.dbf) is listed in Appendix C.

4.3 Trip Distribution Validation Results

Based on the Florida Standard Urban Transportation Model Structure standards, three gravity model checks were developed. They are average trip length by purpose, percentage of intrazonal trips and trip flow patterns/trip length frequency distribution.

Average Trip Length by Purpose

Table 4.2 shows a comparison of average trip length statistics generated by the 2010, 2007, and 2000 Gainesville Urbanized Area Transportation Study models, and Florida Standard Urban Transportation Model Structure standards. Comparisons between the 2010 and 2007 Gainesville Urbanized Area Transportation Study models show a small trip length increase in home-based university, truck-taxi and Internal-External. All other categories of trips have shorter trip lengths than 2007. Overall, the trip lengths are very similar to the 2007 model. Although the average trip lengths are on the low side compared to the Florida Standard Urban Transportation Model Structure standards, this is to be expected given the relatively smaller region covered by the Gainesville Urbanized Area Transportation Study model.
Table 4.2 Average Trip Lengths (in Minutes)

<table>
<thead>
<tr>
<th>Purpose</th>
<th>2010 Alachua County</th>
<th>2007 Alachua County</th>
<th>2000 Alachua County</th>
<th>FSUTMS*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home-Based Work</td>
<td>14.67</td>
<td>14.73</td>
<td>13.92</td>
<td>15-28</td>
</tr>
<tr>
<td>Home-Based Shop</td>
<td>13.09</td>
<td>13.1</td>
<td>13.6</td>
<td>10-18</td>
</tr>
<tr>
<td>Home-Based Social/Recreation</td>
<td>12.49</td>
<td>12.55</td>
<td>11.97</td>
<td>11-19</td>
</tr>
<tr>
<td>Home-Based Other</td>
<td>13.24</td>
<td>13.37</td>
<td>12.79</td>
<td>10-20</td>
</tr>
<tr>
<td>Non-home-Based</td>
<td>10.51</td>
<td>10.79</td>
<td>9.05</td>
<td>10-18</td>
</tr>
<tr>
<td>Home-Based University</td>
<td>9.31</td>
<td>9.14</td>
<td>8.08</td>
<td>NA</td>
</tr>
<tr>
<td>UF Campus/Dorm</td>
<td>6.20</td>
<td>6.22</td>
<td>4.19</td>
<td>NA</td>
</tr>
<tr>
<td>Truck-Taxi</td>
<td>15.40</td>
<td>15.18</td>
<td>13.74</td>
<td>12-20</td>
</tr>
<tr>
<td>Internal-External</td>
<td>25.77</td>
<td>25.69</td>
<td>25.78</td>
<td>27-45</td>
</tr>
</tbody>
</table>

* Table F.1 - FSUTMS-Cube Framework Phase II Model Calibration and Validation Standards

Intrazonal Trip Distribution

Comparisons between the 2010 and 2007 Gainesville Urbanized Area Transportation Study models indicate that the percentage of intrazonal trips is similar for most trip purposes. These results are illustrated in Table 4.3.

Table 4.3 Intrazonal Trip Summary

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Alachua County 2010</th>
<th>Alachua County 2007</th>
<th>FSUTMS*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Trips</td>
<td>Intrazonal Trips</td>
<td>Percent</td>
</tr>
<tr>
<td>Home-Based Work</td>
<td>185,977</td>
<td>2,640</td>
<td>1.42%</td>
</tr>
<tr>
<td>Home-Based Shop</td>
<td>135,794</td>
<td>4,894</td>
<td>3.60%</td>
</tr>
<tr>
<td>Home-Based Social/Recreation</td>
<td>120,814</td>
<td>8,953</td>
<td>7.41%</td>
</tr>
<tr>
<td>Home-Based Other</td>
<td>270,787</td>
<td>10,457</td>
<td>3.86%</td>
</tr>
<tr>
<td>Non-home-Based</td>
<td>354,049</td>
<td>24,913</td>
<td>7.04%</td>
</tr>
<tr>
<td>Home-Based University</td>
<td>64,423</td>
<td>35</td>
<td>0.05%</td>
</tr>
<tr>
<td>UF Campus/Dorm</td>
<td>22,767</td>
<td>1,336</td>
<td>5.87%</td>
</tr>
<tr>
<td>Truck-Taxi</td>
<td>84,055</td>
<td>1,290</td>
<td>1.53%</td>
</tr>
<tr>
<td>Internal-External</td>
<td>101,867</td>
<td>0</td>
<td>0.00%</td>
</tr>
<tr>
<td>Total</td>
<td>1,340,533</td>
<td>54,518</td>
<td>4.07%</td>
</tr>
</tbody>
</table>

* Table F.1 - FSUTMS-Cube Framework Phase II Model Calibration and Validation Standards
Trip Flow Patterns and Trip Length Frequency Distributions

The model area trip flow pattern analysis is another aggregate visualization check in the validation process. Traffic analysis zone-to-traffic analysis zone trip flows were conducted and checked with local knowledge to verify the reasonableness of trip distribution. In Figure 4.2, traffic analysis zone-to-traffic analysis zone daily total trip flows obtained from the model are represented by desire lines. Only daily trip flows that are greater than 20 trips are shown. The concentration of trip desires in the downtown area and general radial patterns make intuitive sense given the land use patterns of Alachua County. Figures 4.3 through 4.6 show the trip length distribution by purpose - the 2007 and 2010 trip length frequency distributions are found to be very similar.

In summary, the 2010 trip distribution validation results are consistent with the 2007 model and within acceptable Florida Standard Urban Transportation Model Structure standards.
Figure 4.2 Desire Lines by Traffic Analysis Zone
Figure 4.3 Trip Length Frequency Distribution for Home-Based Work

Home-Based Work Trip Length Distribution

Figure 4.4 Trip Length Frequency Distribution for Home-Based Shopping

Home-Based Shopping Trip Length Distribution
Figure 4.5 Trip Length Frequency Distribution for Home-Based Social/Recreational

Home-Based Social/Recreational Trip Length Distribution

Figure 4.6 Trip Length Frequency Distribution for Home-Based Other

Home-Based Other Trip Length Distribution
5 Transit Accessibility and Path-Building

Integral to the building of a transit network is the availability of access to transit. A critical component of transit access is identifying the zones that are within an acceptable walking distance to a transit stop. Walk access is generally provided from centroids to stops within a specified distance. Transit path-building involves the transit fares, transit routes and stops/stations. These inputs are used for each of the transit modes during peak and off-peak periods used in the model. The Alachua County 2010 model currently only has one transit mode, local bus, but has the capability of expansion to other transit modes in the future. Figure 5.1 shows the structure of transit network module.

Figure 5.1 Transit Network Module
5.1 Transit Access and Path-Building Module Structure

The Florida Standard Urban Transportation Model Structure transit network module is used to establish transit access and path-building. The module constructs separate peak-period (AM) and off-peak (MD) transit networks using congested highway networks as an input to produce transit vehicle travel times. Transit path-building involves the generation of transit path skim matrices, fares, and station-to-station interchanges.

Transit accessibility is represented by each zone’s pedestrian environment variables (PEV) that are stored in the ZONEDATA file. The pedestrian environment variables define several factors that are essential to have sufficient accessibility to bus stops, such as sidewalk availability, ease of street crossing, non-motorized connections, and building setbacks. Each variable is given a score between 0 and 3 and the accumulated scores of the four pedestrian environment variables are saved as “SUM” in the ZONEDATA file, which ranges from 0 to 12. Future changes to the zonal transit accessibility will require modification of pedestrian environment variable scores as well as updating “SUM” values to get total pedestrian environment variable scores for each traffic analysis zone. Table 5.1 indicates what each pedestrian enhancement variable value represents. These variables and categories remain unchanged from the 2007 model.

<table>
<thead>
<tr>
<th>Variables</th>
<th>PEV = 0</th>
<th>PEV = 1</th>
<th>PEV = 2</th>
<th>PEV = 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sidewalk Availability</td>
<td>No sidewalks</td>
<td>&lt;10 percent have sidewalks</td>
<td>10 to 90 percent have sidewalks</td>
<td>&gt;90 percent have sidewalks</td>
</tr>
<tr>
<td>Ease of Street Crossing</td>
<td>Crossing difficult</td>
<td>&lt;10 percent have easy crossing</td>
<td>10 to 90 percent with easy crossing</td>
<td>&gt;90 percent with easy crossing</td>
</tr>
<tr>
<td>Non-motorized Connections</td>
<td>No connections</td>
<td>&lt;10 percent have connections</td>
<td>10 to 90 percent have connections</td>
<td>&gt;90 percent have connections</td>
</tr>
<tr>
<td>Building Setbacks</td>
<td>All large setbacks</td>
<td>&lt;10 percent have minimum setbacks</td>
<td>10 to 90 percent have minimum setbacks</td>
<td>&gt;90 percent have minimum setbacks</td>
</tr>
</tbody>
</table>

5.2 Transit Access and Path-Building Model Development and Validation

Most of the effort in validating the transit accessibility and path-building focused on ensuring that the transit network was up to date and accurately reflected base year conditions. In addition, walk access links were checked in order to ensure adequate connectivity.
Transit Accessibility

The key effort in validating the transit network consisted of three parts. The first was to review the existing transit network and update it to year 2010 operating conditions. This required significant additions and rerouting of the transit lines. The second part was to check that there is no “Later Gator” bus route coded around the University of Florida campus since this may result in over assignment. Finally, pedestrian environment variables in the ZONEDATA file were reviewed and updated for each zone. However, because of limited data availability, the pedestrian environment variables are generally kept the same as 2007.

Transit Fare

The Gainesville Regional Transit System (RTS) transit fare was $1.00 during the year 2007, used for the Gainesville Urbanized Area Transportation Study base year 2007 model validation. This fare was increased to $1.50 in 2010, according to the Gainesville Regional Transit System. Details on how the bus fare was adjusted during validation are further explained in Transit Procedure and Assignment section. The fare has been set to $1.50 for base year 2010 in ALACHUA.FAR file. For the future year transit analysis, the bus fare factor (BUSSFAREFAC) value may need to be updated to represent fare increases beyond inflation.

Transit Routes

As validation efforts moved towards reasonable transit assignments, transit network rerouting, bus stop locations and local zonal access to bus stops were reviewed using the Gainesville Regional Transit System General Transit Feed Specification files. Headway data was also extracted from the Gainesville Regional Transit System General Transit Feed Specification files and the transit route file (troute10.lin) was updated. The transit route file had to be overlaid on the highway network when new bus locations were added due to the necessity of splitting highway links where a bus stop exists. The highway network was therefore updated at the same time as the transit route file was updated. “Later Gator” bus routes are not included in the 2010 Alachua County model as these are evening bus services specifically for University of Florida students and operate for only limited hours while the model is designed to estimate daily peak and off-peak transit ridership.

Transit Skimming Parameters

The path parameters used during transit skims were reviewed for reasonableness and consistency with mode choice coefficients. Consistency between skimming and mode choice is
an area that is considered by the Federal Transit Administration (FTA) when a model is used for New Starts or Small Starts purposes. A few skim parameters were identified (in the 2007 model) that were not consistent with mode choice. Auto driving time to park and ride lots had a perceived weight parameter of 1.0 in the 2007 model skim parameters while the ratio of out-of-vehicle travel time to in-vehicle travel time in mode choice is 2.0 for home-based work. Also, transfer waiting time had a perceived weight parameter of 1.4 in the 2007 model. Both of these skimming parameters were set to 2.0 in the 2010 model for consistency with mode choice.

5.3 Transit Access and Path-Building Model Validation Results

The Federal Transit Administration has noted that certain common practices in transit path building can have undesired impacts on ridership forecasts. Minimum and maximum values of time and distance used to determine valid transit paths and modal availability can have unexpected effects. Path building parameters and settings should remain the same for all steps of the model (skimming, assignment). All transit parameters in the model were reviewed for consistency across the model chain. The careful review of all transit parameters ensured reasonableness of transit access and path building.

Average weekday transit trips in the year 2010 computed from the data provided by the Gainesville Regional Transit System were 41,535. The model currently is estimating an average of 40,403 transit riders per day, which is only 2.7% lower than observed data and is well within the acceptable range of 3% - 9% from the Florida Standard Urban Transportation Structure standards. Additional details on transit assignment are provided in Section 8.0 of this report.
6 Mode Choice

Since Alachua County is the home of University of Florida and Santa Fe Colleges, like other college towns the transit system plays an important role in daily transportation and needs to be considered in modeling. The mode choice model used in the 2010 Alachua County model is a set of nested logit models that estimate modal shares among several categories of auto and transit modes. This section of the report describes the structure and validation of the mode choice model. Figure 6.1 shows the structure of this part of model.

Figure 6.1 Mode Choice Model Chain

MODE CHOICE
6.1 Mode Choice Model Structure

The Florida Standard Urban Transportation Model Structure process makes use of a nested logit model for mode choice, except in the case of highway only models (i.e., those that do not include transit networks). The entire mode choice process is executed via Cube/Voyager scripting, using the MATRIX program.

One script, MCMAT00A.S, creates a trip table matrix set containing trips for all five purposes: home-based work (HBW), home-based other (HBO), non-home-based (NHB), home-based university (HBU) and University of Florida Campus/Dorm (DORM). The trip tables for each purpose are separated into 8 mode specific tables: auto drive-alone (DA), auto 2-person carpool (CP), auto 3+ person carpool (CX), walk to local bus (WB), walk to premium bus (WX), drive to transit (BA), walk (WK), and bike (BK). Transit skim data compiled in the transit network module and highway skims generated during trip distribution are inputs into the “MODE” module. The peak period utilizes the congested highway skims, whereas the off-peak period uses the free flow skims.

After running the mode choice model, the outputs are balanced into origin and destination trip tables. A separate script, MCMAT00C.S, combines trip purposes and outputs separate trip tables for the following modes and these trip tables are then used during the highway and transit assignment phases of the model:

1. Drive alone auto;
2. Carpool auto;
3. Light duty trucks;
4. Heavy duty trucks;
5. External-external trips;
6. Peak period transit;
7. Off-peak period transit;
8. Non-motorized travelers; and
9. Internal auto persons (combination of modes 1 and 2, above).

A few minor changes were made to fix the deficiencies embedded in the 2007 mode choice script. Since the unit of the fare cost is dollars while other costs (like parking) are using cents as the unit, fare cost needs to be multiplied 100 to become comparable to other costs. This was not correctly done for some of the trip purposes in the 2007 model. The scripts referenced above and others pertaining to Mode Choice can be found in Appendix D, along with the corresponding model flowchart for each step.
6.2 Development and Validation of Mode Choice Model

The mode choice model step for the Gainesville Urbanized Area Transportation Study 2010 model was developed from the Gainesville Urbanized Area Transportation Study 2007 model, with a few changes to mode choice scripting (MCMAT00A.S). The constants file (MCCONSTANTS.CSV) was modified with recalibration of the transit trips. Coefficients (MCCOEFFICIENTS.CSV) are kept the same as 2007 and are in a numeric range consistent with current Florida Standard Urban Transportation Model Structure validation standards. The mode choice coefficients were not revised given that a better source of data, such as a new household travel survey, is not available. All parameters are shown in Table 6.1.

**Table 6.1 Mode Choice Coefficients for 2010 Model**

<table>
<thead>
<tr>
<th>Mode Choice Model Parameters</th>
<th>2010</th>
<th>FSUTMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home-Based Work IVTT</td>
<td>-0.025</td>
<td>-0.01 to -0.05</td>
</tr>
<tr>
<td>Home-Based Non-Work IVTT</td>
<td>-0.02</td>
<td>-0.007 to -0.033</td>
</tr>
<tr>
<td>Non-Home Based IVTT</td>
<td>-0.024</td>
<td>-0.01 to -0.05</td>
</tr>
<tr>
<td>University IVTT</td>
<td>-0.024</td>
<td>-0.02 to -0.03</td>
</tr>
<tr>
<td>Home-Based Work OVTT</td>
<td>-0.049</td>
<td>NA</td>
</tr>
<tr>
<td>Home-Based Non-Work OVTT</td>
<td>-0.048</td>
<td>NA</td>
</tr>
<tr>
<td>Non Home-Based OVTT</td>
<td>-0.07</td>
<td>NA</td>
</tr>
<tr>
<td>University OVTT</td>
<td>-0.048</td>
<td>NA</td>
</tr>
<tr>
<td>Home-Based Work OVTT/IVTT</td>
<td>2.0</td>
<td>1.5 to 3.0</td>
</tr>
<tr>
<td>Home-Based Non-Work OVTT/IVTT</td>
<td>2.4</td>
<td>2.0 to 6.0</td>
</tr>
<tr>
<td>Non Home-Based OVTT/IVTT</td>
<td>2.9</td>
<td>2.0 to 7.0</td>
</tr>
<tr>
<td>University OVTT/IVTT</td>
<td>2.0</td>
<td>2.0 to 3.0</td>
</tr>
</tbody>
</table>

*IVTT = in-vehicle travel time, OVTT = out-vehicle travel time

6.3 Mode Choice Model Results

Mode choice results were compared with statistics from Census American Community Survey data for the home-based work (HBW) trip purpose. With respect to transit mode split, the Gainesville Urbanized Area Transportation Study 2010 mode split percentages were smaller than the 2007 model for the home-based work purpose. The home-based work purpose showed 4.18 percent of all home-based work trips being transit as opposed to 4.39 percent for the 2007 model. The 2010 results are reasonable compared to the transit mode share estimate from the American Community Survey (refer to the end of Table 6.2). In total, 2.38 percent of all trips in the 2010 model were allocated to transit modes while 2.17 percent of all trips in the 2007 model were allocated to transit. Table 6.2 contains the Gainesville Urbanized Area Transportation Study 2010 model Mode Choice validation results.
### Table 6.2 Mode Choice Validation Summary

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2007</th>
<th>2000</th>
<th>FSUTMS*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trips</td>
<td>Percent</td>
<td>Trips</td>
<td>Percent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>of Trips</td>
<td></td>
<td>of Trips</td>
</tr>
<tr>
<td>Home-Based Work</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive Alone</td>
<td>145,470</td>
<td>78.25%</td>
<td>143,880</td>
<td>78.51%</td>
</tr>
<tr>
<td>Two Passengers</td>
<td>16,609</td>
<td>8.93%</td>
<td>15,951</td>
<td>8.70%</td>
</tr>
<tr>
<td>3+ Passengers</td>
<td>8,287</td>
<td>4.46%</td>
<td>8,019</td>
<td>4.38%</td>
</tr>
<tr>
<td>Total Transit</td>
<td>7,779</td>
<td>4.18%</td>
<td>8,043</td>
<td>4.39%</td>
</tr>
<tr>
<td>Walk</td>
<td>4,339</td>
<td>2.33%</td>
<td>4,098</td>
<td>2.24%</td>
</tr>
<tr>
<td>Bike</td>
<td>3,412</td>
<td>1.84%</td>
<td>3,278</td>
<td>1.79%</td>
</tr>
<tr>
<td>Total</td>
<td>185,896</td>
<td>100.00%</td>
<td>183,269</td>
<td>100.00%</td>
</tr>
<tr>
<td>Home-Based Other</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive Alone</td>
<td>204,045</td>
<td>38.71%</td>
<td>200,318</td>
<td>39.63%</td>
</tr>
<tr>
<td>Two Passengers</td>
<td>201,574</td>
<td>38.24%</td>
<td>191,683</td>
<td>37.92%</td>
</tr>
<tr>
<td>3+ Passengers</td>
<td>98,476</td>
<td>18.68%</td>
<td>90,914</td>
<td>17.99%</td>
</tr>
<tr>
<td>Total Transit</td>
<td>1,852</td>
<td>0.35%</td>
<td>2,943</td>
<td>0.58%</td>
</tr>
<tr>
<td>Walk</td>
<td>19,425</td>
<td>3.68%</td>
<td>17,952</td>
<td>3.55%</td>
</tr>
<tr>
<td>Bike</td>
<td>1,786</td>
<td>0.34%</td>
<td>1,631</td>
<td>0.32%</td>
</tr>
<tr>
<td>Total</td>
<td>527,158</td>
<td>100.00%</td>
<td>505,441</td>
<td>100.00%</td>
</tr>
<tr>
<td>Non Home-Based</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive Alone</td>
<td>172,087</td>
<td>48.61%</td>
<td>166,902</td>
<td>50.70%</td>
</tr>
<tr>
<td>Two Passengers</td>
<td>117,608</td>
<td>33.22%</td>
<td>106,625</td>
<td>32.39%</td>
</tr>
<tr>
<td>3+ Passengers</td>
<td>52,509</td>
<td>14.83%</td>
<td>45,341</td>
<td>13.77%</td>
</tr>
<tr>
<td>Total Transit</td>
<td>2,415</td>
<td>0.68%</td>
<td>2,587</td>
<td>0.79%</td>
</tr>
<tr>
<td>Walk</td>
<td>6,274</td>
<td>1.77%</td>
<td>5,105</td>
<td>1.55%</td>
</tr>
<tr>
<td>Bike</td>
<td>3,158</td>
<td>0.89%</td>
<td>2,640</td>
<td>0.80%</td>
</tr>
<tr>
<td>Total</td>
<td>354,051</td>
<td>100.00%</td>
<td>329,200</td>
<td>100.00%</td>
</tr>
<tr>
<td>Home-Based University</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive Alone</td>
<td>32,169</td>
<td>49.93%</td>
<td>34,900</td>
<td>45.96%</td>
</tr>
<tr>
<td>Two Passengers</td>
<td>3,915</td>
<td>6.08%</td>
<td>4,399</td>
<td>5.79%</td>
</tr>
<tr>
<td>3+ Passengers</td>
<td>1,960</td>
<td>3.04%</td>
<td>2,261</td>
<td>2.98%</td>
</tr>
<tr>
<td>Total Transit</td>
<td>13,478</td>
<td>20.92%</td>
<td>9,178</td>
<td>12.09%</td>
</tr>
<tr>
<td>Walk</td>
<td>6,811</td>
<td>10.57%</td>
<td>15,025</td>
<td>19.78%</td>
</tr>
<tr>
<td>Bike</td>
<td>6,090</td>
<td>9.45%</td>
<td>10,179</td>
<td>13.40%</td>
</tr>
<tr>
<td>Total</td>
<td>64,423</td>
<td>100.00%</td>
<td>75,942</td>
<td>100.00%</td>
</tr>
<tr>
<td>University of Florida Campus/Dorm</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Transit</td>
<td>1,947</td>
<td>8.55%</td>
<td>1,981</td>
<td>8.40%</td>
</tr>
<tr>
<td>Walk</td>
<td>15,180</td>
<td>66.68%</td>
<td>15,615</td>
<td>66.25%</td>
</tr>
<tr>
<td>Bike</td>
<td>5,641</td>
<td>24.78%</td>
<td>5,974</td>
<td>25.35%</td>
</tr>
</tbody>
</table>
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

<table>
<thead>
<tr>
<th></th>
<th>2010</th>
<th>2007</th>
<th>2000</th>
<th>FSUTMS*</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Trips</td>
<td>Percent of Trips</td>
<td>Trips</td>
<td>Percent of Trips</td>
</tr>
<tr>
<td>Total</td>
<td>22,767</td>
<td>100.00%</td>
<td>23,570</td>
<td>100.00%</td>
</tr>
<tr>
<td>All Purposes</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive Alone</td>
<td>553,771</td>
<td>47.97%</td>
<td>546,000</td>
<td>47.80%</td>
</tr>
<tr>
<td>Two Passengers</td>
<td>339,706</td>
<td>29.43%</td>
<td>318,658</td>
<td>27.90%</td>
</tr>
<tr>
<td>3+ Passengers</td>
<td>161,232</td>
<td>13.97%</td>
<td>146,535</td>
<td>12.83%</td>
</tr>
<tr>
<td>Total Transit</td>
<td>27,471</td>
<td>2.38%</td>
<td>24,732</td>
<td>2.17%</td>
</tr>
<tr>
<td>Walk</td>
<td>52,029</td>
<td>4.51%</td>
<td>57,795</td>
<td>5.06%</td>
</tr>
<tr>
<td>Bike</td>
<td>20,087</td>
<td>1.74%</td>
<td>23,702</td>
<td>2.08%</td>
</tr>
<tr>
<td>Total</td>
<td>1,154,295</td>
<td>100.00%</td>
<td>1,142,154</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

*FSUTMS = Florida Standard Urban Transportation Model Structure

American Community Survey 5 Years
2006-2010

<table>
<thead>
<tr>
<th>Classes</th>
<th>Home-Based Work Percent of Trips</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive Alone</td>
<td>74.18%</td>
</tr>
<tr>
<td>Carpool</td>
<td>11.13%</td>
</tr>
<tr>
<td>Transit</td>
<td>3.43%</td>
</tr>
<tr>
<td>Walk</td>
<td>3.31%</td>
</tr>
<tr>
<td>Bicycle</td>
<td>3.24%</td>
</tr>
</tbody>
</table>
7 Highway Assignment

A proper validation of the highway assignment is critical to the meaningful use of travel demand models. The objective of the assignment module is to load trips to available paths to achieve system equilibrium. Highway trip assignment determines the shortest path between any given origin destination pair and loads the auto trips produced by the mode choice module onto those path links. The trips are loaded incrementally – once the first iteration is completed, link travel times are then recomputed, the new shortest paths are subsequently determined, and the next iteration is executed. This process proceeds through a series of iterative calculations until no one can shorten their travel time by changing routes. In another words, system equilibrium is achieved at the end of highway assignment. The red dotted line in Figure 7.1 shows the highway assignment model chain. Highway assignment is combined with transit assignment at this model. The details of transit assignment will be discussed in next chapter.

Figure 7.1 Highway Assignment Module

HIGHWAY AND TRANSIT ASSIGNMENT
7.1 Highway Assignment Procedure

Auto trips are loaded onto the network by means of an iterative equilibrium highway load program based on an all or nothing assignment algorithm. In total, six model runs were executed in order to validate the Gainesville Urbanized Area Transportation Study 2010 model, excluding numerous test runs that were necessary for iterative adjustment and minor calibration. Adjustments were made to key elements of the modeling process to achieve satisfactory validation results. After each run, a summary of the results was compiled and analyzed in order to identify areas for improvement in the model and successful strategies toward validation enhancement. A number of key evaluation statistics, which are recommended by the FSUTMS- Cube Framework Phase II: Model Calibration and Validation Standards Final Report, were generated for model calibration and validation purposes. Model validation was accomplished by minimizing the difference between the model estimated volumes and observed traffic counts for the year 2010 on network segments for the entire study area.

7.2 Highway Assignment Validation Results

The evaluation of the highway assignment module is based on comparisons between observed traffic counts and model estimated volumes. Base year model estimated traffic volumes are compared to the 2010 traffic counts in different ways to determine whether the model projections are reasonable. Figure 7.2 shows a volume to count scatter plot for the overall model. The indicators that are available for determining the overall performance of the highway assignment model are volume to count ratios by area type, facility type, lane categories, screen line volume over count ratios, and root mean square error (RMSE). Each of the measures is discussed in the remainder of this section.
Assignment Performance by Area Type/Facility Type/Lanes Categories

Table 7.1 contains details of the volume-over-count ratios for each link group category by area type, facility type, and number of lanes. At the area-wide level, the 2010 base year model is overestimating volumes by 6 percent.

Screenline Performance

Analyzing volume-over-count ratios along screenlines allows for examining traffic flows across geographic subareas and corridors. Florida accuracy standards along screenlines have historically varied from +/-5 percent to +/-20 percent. There are 10 screenlines in the Alachua County 2010 model including an external cordon line measuring trips coming into and going out of the study area.

Table 7.2 shows volume to count ratios by screenline (note that double counting of trips was avoided in creating the summary). Figure 7.3 shows the locations of the screenlines. Screenline 9 does fall slightly outside of the FSUTMS standard. However, this validation effort was focused on the overall region wide calibration.

### Table 7.1 Volume to Count Performance by Category

<table>
<thead>
<tr>
<th>Category</th>
<th>2010</th>
<th>Florida Standard Urban Transportation Model Structure</th>
<th>Number of Links</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acceptable</td>
<td>Preferable</td>
</tr>
<tr>
<td>Freeway</td>
<td>0.96</td>
<td>+7%</td>
<td>+6%</td>
</tr>
<tr>
<td>Divided Arterial</td>
<td>1.10</td>
<td>+15%</td>
<td>+10%</td>
</tr>
<tr>
<td>Screenline</td>
<td>Area Type</td>
<td>Volume Ratio</td>
<td>Number of Lanes</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
<td>Undivided Arterial</td>
<td>1.08</td>
<td>1 Lane</td>
</tr>
<tr>
<td>2</td>
<td>Collectors</td>
<td>0.92</td>
<td>2 Lane</td>
</tr>
<tr>
<td>3</td>
<td>One-Way/Frontage</td>
<td>1.02</td>
<td>3 Lane</td>
</tr>
<tr>
<td>4</td>
<td>Central Business District</td>
<td>0.94</td>
<td>1 Lane</td>
</tr>
<tr>
<td>5</td>
<td>Central Business District Fringe</td>
<td>1.07</td>
<td>2 Lane</td>
</tr>
<tr>
<td>6</td>
<td>Residential</td>
<td>0.98</td>
<td>3 Lane</td>
</tr>
<tr>
<td>7</td>
<td>Other Business District</td>
<td>1.11</td>
<td>1 Lane</td>
</tr>
<tr>
<td>8</td>
<td>Rural</td>
<td>1.10</td>
<td>2 Lane</td>
</tr>
</tbody>
</table>

Table 7.2 Volume to Count Performance by Screenline

<table>
<thead>
<tr>
<th>Screenline</th>
<th>2010 Volumes Over Counts</th>
<th>Florida Department of Transportation Criteria*</th>
<th>Screenline Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28,625</td>
<td>0.81 +/-20%</td>
<td>Crossing I-75</td>
</tr>
<tr>
<td>2</td>
<td>60,132</td>
<td>1.09 +/-15%</td>
<td>Crossing East University of Florida Campus</td>
</tr>
<tr>
<td>3</td>
<td>62,717</td>
<td>1.03 +/-15%</td>
<td>Crossing State Road 121</td>
</tr>
<tr>
<td>4</td>
<td>26,236</td>
<td>0.89 +/-20%</td>
<td>East-West Cutline west of I-75</td>
</tr>
<tr>
<td>5</td>
<td>137,240</td>
<td>1.05 +/-10%</td>
<td>North-South Crossing State Road 222 (39th Avenue)</td>
</tr>
<tr>
<td>6</td>
<td>78,751</td>
<td>1.11 +/-10%</td>
<td>North-South Cutline in Northwest County/High Springs</td>
</tr>
<tr>
<td>7</td>
<td>3,236</td>
<td>1.10 +/-20%</td>
<td>La Crosse Area</td>
</tr>
<tr>
<td>8</td>
<td>25,877</td>
<td>1.13 +/-20%</td>
<td>East-West Crossing U.S. 301</td>
</tr>
<tr>
<td>9</td>
<td>60,631</td>
<td>0.83 +/-15%</td>
<td>Micanopy Area</td>
</tr>
<tr>
<td>10</td>
<td>217,669</td>
<td>1.00 +/-1%</td>
<td>External Cordon</td>
</tr>
</tbody>
</table>

*FSUTMS-Cube Framework Phase II: Model Calibration and Validation Standards Final Report
Percent Root Mean Square Error

Root Mean Square Error is among the most commonly reported statistics in model validation. Root Mean Square Error, a measure of dispersion, tends to normalize model error better than volume-over-count ratios that allow for high ratios to offset low ratios. Table 7.3 shows the Root Mean Square Error statistics by volume group. All categories met established accuracy standards.

Table 7.3 Root Mean Squared Error

<table>
<thead>
<tr>
<th>Volume Range</th>
<th>2010 Model</th>
<th>Florida Standard Urban Transportation Model Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Acceptable</td>
</tr>
<tr>
<td>Average Total</td>
<td>31.3%</td>
<td>+/-45%</td>
</tr>
<tr>
<td></td>
<td>Average Total</td>
<td>+/-35%</td>
</tr>
<tr>
<td>1-5,000</td>
<td>47.1%</td>
<td>+/-45%</td>
</tr>
<tr>
<td>5,000-10,000</td>
<td>28.5%</td>
<td>+/-35%</td>
</tr>
<tr>
<td>10,000-20,000</td>
<td>23.4%</td>
<td>+/-25%</td>
</tr>
<tr>
<td>20,000-30,000</td>
<td>23.0%</td>
<td>+/-25%</td>
</tr>
<tr>
<td>30,000-40,000</td>
<td>11.7%</td>
<td>+/-15%</td>
</tr>
</tbody>
</table>
Figure 7.3 Gainesville Urbanized Area Transportation Study 2010 Model Screenlines
8 Transit Procedure and Assignment

The transit procedure in the Gainesville Urbanized Area Transportation Study 2010 model includes development of transit input data, building transit path, calculating the number of transit trips from the mode choice model, and loading the transit trips to the transit network. Each of these four aspects of the transit procedure is discussed in this document.

The development of the transit input data was discussed in Section 5. Information on routes, stop locations, service characteristics, and ridership was provided by Gainesville Regional Transit System staff. Detail transit network coding and review was conducted in order to ensure that the system representation in the model is accurate. The ridership data was utilized to assess the reasonableness of the model outputs.

A detailed review of all transit parameters including path building was conducted and documented. Transit levels of service are computed separately for peak hours and off-peak hours. During the process of distribution, the preloaded highway network with pre-assignment loads based on initial time skim is created. Then in the process of transit path creation, the preloaded network is used to calculate the time skim for transit during peak hours. Transit vehicle speed (link travel time) is determined as a function of the automobile speed on each link. The type of relationship used for each transit service type is identified based on area type and facility type of the link. These auto/transit speed relationships are used in a lookup function to assign speeds for transit paths. Then the transit travel time is calculated proportionally according to relative relationship between auto travel times. During this step, path files with shortest time and distance for peak and off-peak are also created which are later used at the transit assignment stage. Other variables, like headway, transit accessibility, waiting time and transit path building parameters are assigned to each route during this process as well.

The Gainesville Urbanized Area Transportation Study 2010 model uses a nested logit model approach for mode choice. Variables that are considered include in-vehicle transit time, out-of-vehicle transit time, cost, bike or walk to transit time, and pedestrian environment variables. The coefficients used in mode choice for these variables are shown in Table 8.1. For home-based work trips, the local bus fare is discounted to 25% since there is an employee pass program. This 25% factor was retained from the 2007 model, and was originally computed to match model computed home-based work ridership to observed ridership. Since the transit fare is covered by tuition fees of University of Florida students the local bus fare for home-based university and dorm-university walk to local transit is discounted to 10%. For home-based university walk and drive to premium transit the bus fare is free. The percentages of each
mode are calculated by using nested logit mode choice. The details of the mode choice model validation are discussed in Section 6.

Table 8.1 Mode Choice Coefficient for Transit Variables

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Home-based Work</th>
<th>Home-based Other</th>
<th>Non home-based</th>
<th>University</th>
</tr>
</thead>
<tbody>
<tr>
<td>civt - In-Vehicle Transit Time Coefficient</td>
<td>-0.025</td>
<td>-0.02</td>
<td>-0.024</td>
<td>-0.024</td>
</tr>
<tr>
<td>covt – Out-of-Vehicle Transit Time Coefficient</td>
<td>-0.049</td>
<td>-0.048</td>
<td>-0.07</td>
<td>-0.048</td>
</tr>
<tr>
<td>ccst - Cost Coefficient</td>
<td>-0.005</td>
<td>-0.011</td>
<td>-0.009</td>
<td>-0.011</td>
</tr>
<tr>
<td>cwt - Walk only Coefficient</td>
<td>-0.042</td>
<td>-0.083</td>
<td>-0.052</td>
<td>-0.083</td>
</tr>
<tr>
<td>cbt - Bike Coefficient</td>
<td>-0.109</td>
<td>-0.117</td>
<td>-0.096</td>
<td>-0.117</td>
</tr>
<tr>
<td>pti - Walk to Transit Pedestrian Enhancement Variable Coefficient</td>
<td>1.15</td>
<td>0.6</td>
<td>0.45</td>
<td>0.25</td>
</tr>
<tr>
<td>pwi - Walk Pedestrian Enhancement Variable Coefficient Origin</td>
<td>0.35</td>
<td>0.175</td>
<td>0.22</td>
<td>0.4</td>
</tr>
<tr>
<td>pwj - Walk Pedestrian Enhancement Variable Coefficient Destination</td>
<td>0.3</td>
<td>0.164</td>
<td>0.164</td>
<td>0.35</td>
</tr>
<tr>
<td>pbi - Bike Pedestrian Enhancement Variable Coefficient Origin</td>
<td>0.47</td>
<td>0.07</td>
<td>0.066</td>
<td>0.3</td>
</tr>
<tr>
<td>pbj - Bike Pedestrian Enhancement Variable Coefficient Destination</td>
<td>0.006</td>
<td>0</td>
<td>0.006</td>
<td>0.006</td>
</tr>
</tbody>
</table>

The last step in the process is to assign the transit trips to the routes. Separate loads are conducted by mode and period as allocated in the mode choice model. Home-based work trips are assigned to the peak period network while home-based other, non-home-based, home-based university, and dorm-university trips are assigned to the off-peak network. Within each period there are three transit mode choices available: walk to local bus, walk to express bus, and drive to best available transit. The transit assignment flow is relatively simple in this model. It loads the transit trips from mode choice onto the transit paths computed during path building. There are no parameters used in this procedure. The red dotted square in Figure 8.1 shows the transit assignment module.
Figure 8.1 Transit Assignment Module

HIGHWAY AND TRANSIT ASSIGNMENT

[Diagram showing the process of highway and transit assignment]

[Flowchart with steps for peak season weekday, peak period walk to transit assignment, off-peak period walk to transit assignment, and off-peak period auto to transit assignment]

Highway Evaluation Scripts
- Script File
- NETWORK
- HEVAL
- Output Network
- Link File
- SMS

Additional Highway Evaluation Scripts
- Script File
- NETWORK
- HEVAL 2
- Output Network
- Print File
- Print Data 2

[Attributes used in COMBINEDLOADED.net]
- MOTORIZEDVOL = 2010 Model Volume (directional)
- COUNT.THR = 2010 Directional Peak Season Volume VC = 2010 Volume-to-Count Ratios
8.1 Transit Assignment Validation Results

Not nearly as much effort is typically expended on transit assignment validation as is devoted to highway assignment validation, in part due to transit assignment not impacting roadway congestion or capacity, as well as a general lack of post-processing programs and scripts to summarize transit assignment accuracy. However, transit is a key component of the Gainesville Urbanized Area Transportation Study 2010 model and it is important to validate the transit assignment reasonably well at the system level at a minimum. The transit assignment step in the model chain loads person trips onto the transit network. Separate loadings are conducted by mode and period: home-based work trips are assigned to the peak period network while home-based other, non-home-based, home-based university, and dorm-university trips are assigned to the off-peak network.

Overall validation for transit assignment was based on an analysis of the transit ridership not only as a system but also on a route by route basis. Transit assignment loadings were compared to average weekday “unlinked” route ridership data provided by Gainesville Regional Transit System. Unlinked trips are equal to bus boardings and count all the times that a person boards a bus, both initially and during transfers.

Table 8.2 shows route level comparison of observed ridership and ridership estimated by the model. The 2010 Gainesville Regional Transit System weekday ridership was calculated using the average of the months when school is in session during 2010 (excluding summer months of June, July, and August). The average weekday ridership during the summer months is much lower than the rest of the year. Since the model forecasts include university trips, it is reasonable to use average weekday ridership from September to May for comparison purposes.

The Gainesville Urbanized Area Transportation Study 2010 model trips are the number of transit trips loaded to each route by the transit assignment module. System-wide model estimates of unlinked trips are very close to Gainesville Regional Transit System observed numbers. As shown on Table 8.2, the 2010 transit assignment model estimated 40,403 “unlinked” riders system wide, while the Gainesville Regional Transit System reported 41,535 average weekday ridership system wide for 2010 while school was in session.

At the system-wide level, ridership for the Gainesville Urbanized Area Transportation Study 2010 model differs by only 2.7 percent from data provided by Gainesville Regional Transit System. This is within the preferred 3-9 percent Florida Department of Transportation validation standard. On a route-by-route basis, the 2010 transit assignment model validates
well with 26 out of 36 bus routes performing within the preferred accuracy range, and 29 of 36 within the acceptable range. In general, regional models do not validate well on a route by route basis. More rigorous validation is conducted for specific transit studies (New Starts/ Small Starts projects for example) or other ridership forecasting tools are applied (for example, TBEST is typically applied for transit development plans). Nonetheless, the Gainesville Urbanized Area Transportation Study 2010 model performs well in transit assignment at the system-wide level and performs reasonably well on a route-by-route basis.

Appendix D contains the model script files and flowchart for transit assignment. These show the processes, input parameters, and output file locations.
<table>
<thead>
<tr>
<th>Route</th>
<th>Route Description</th>
<th>2010 Gainesville Regional Transit System Weekday Ridership</th>
<th>Gainesville Urbanized Area Transportation Study 2010 Model</th>
<th>Percent Error</th>
<th>Acceptable</th>
<th>Preferable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Downtown to Butler Plaza via Archer Road</td>
<td>2,008</td>
<td>3,538</td>
<td>-76%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>2</td>
<td>Downtown to Health Department via SE 15th Street</td>
<td>369</td>
<td>26</td>
<td>-93%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>5</td>
<td>Downtown to Oaks Mall via University Avenue</td>
<td>1,648</td>
<td>2,180</td>
<td>32%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>6</td>
<td>Downtown to Gainesville Mall via 6th Street</td>
<td>391</td>
<td>241</td>
<td>-38%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>7</td>
<td>Downtown to Eastwood Meadows</td>
<td>425</td>
<td>46</td>
<td>-89%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>8</td>
<td>Shands to Northwood Village via NW 13th Street</td>
<td>1,199</td>
<td>1,794</td>
<td>50%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>9</td>
<td>Mccarty to Hunters Run</td>
<td>3,076</td>
<td>2,013</td>
<td>-35%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>10</td>
<td>Downtown to Santa Fe via NW 16th Ave</td>
<td>386</td>
<td>114</td>
<td>-70%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>11</td>
<td>Downtown to Eastwood Meadows via University Ave.</td>
<td>491</td>
<td>211</td>
<td>-57%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>12</td>
<td>Mccarty to Butler Plaza</td>
<td>3,189</td>
<td>1,855</td>
<td>-42%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>13</td>
<td>Shands to Florida Works via SW 13th Street</td>
<td>1,800</td>
<td>977</td>
<td>-46%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>15</td>
<td>Downtown to Gainesville Mall</td>
<td>1,072</td>
<td>487</td>
<td>-55%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>16</td>
<td>Shands to Sugar Hill via SW 16th Avenue</td>
<td>833</td>
<td>467</td>
<td>-44%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>17</td>
<td>Shands to Downtown</td>
<td>764</td>
<td>209</td>
<td>-73%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>20</td>
<td>Mccarty to Oaks Mall via SW 20th Avenue</td>
<td>4,528</td>
<td>3,193</td>
<td>-29%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>21</td>
<td>Mccarty to Cabana Beach</td>
<td>1,907</td>
<td>888</td>
<td>-53%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>22</td>
<td>Mccarty to SW 43rd St at SW 24th Avenue</td>
<td>333</td>
<td>472</td>
<td>42%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>23</td>
<td>Oaks Mall to Santa Fe</td>
<td>183</td>
<td>65</td>
<td>-64%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>24</td>
<td>Downtown to Job Corps via SR 24 (Waldo Road)</td>
<td>361</td>
<td>241</td>
<td>-33%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>25</td>
<td>Mccarty to Airport</td>
<td>55</td>
<td>528</td>
<td>857%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>29</td>
<td>Beaty Towers to Cobblestone</td>
<td>159</td>
<td>568</td>
<td>256%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>34</td>
<td>HUB to Lexington Crossing</td>
<td>2,050</td>
<td>1,384</td>
<td>-32%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>35</td>
<td>Mccarty to Homestead Apartments</td>
<td>2,949</td>
<td>4,124</td>
<td>40%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>36</td>
<td>Mccarty to Williston Plaza</td>
<td>584</td>
<td>1,898</td>
<td>225%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>38</td>
<td>HUB to Gainesville Place</td>
<td>532</td>
<td>357</td>
<td>-33%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>43</td>
<td>Downtown to Santa Fe via 43rd Street</td>
<td>801</td>
<td>2,246</td>
<td>180%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>Route</td>
<td>Route Description</td>
<td>2010 Gainesville Regional Transit System Weekday Ridership</td>
<td>Gainesville Urbanized Area Transportation Study 2010 Model</td>
<td>Percent Error</td>
<td>Acceptable</td>
<td>Preferable</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>----------------------------------------------------------</td>
<td>---------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>75</td>
<td>Oaks Mall to Butler Plaza via 75th Street</td>
<td>1,033</td>
<td>271</td>
<td>-74%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>117</td>
<td>Park-N-Ride 2 (SW 34th Street)</td>
<td>681</td>
<td>1,191</td>
<td>75%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>118</td>
<td>Park-N-Ride 1 (Harn Museum)</td>
<td>2,107</td>
<td>1,896</td>
<td>-10%</td>
<td>65%</td>
<td>35%</td>
</tr>
<tr>
<td>119</td>
<td>Family Housing</td>
<td>338</td>
<td>394</td>
<td>17%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>120</td>
<td>West Circulator (Fraternity Row)</td>
<td>1,638</td>
<td>1,805</td>
<td>10%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>121</td>
<td>Commuter Lot</td>
<td>1,096</td>
<td>1,095</td>
<td>0%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td>122</td>
<td>University of Florida North/South Circulator</td>
<td>150</td>
<td>421</td>
<td>181%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>125</td>
<td>Lakeside</td>
<td>918</td>
<td>1,037</td>
<td>13%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>126</td>
<td>University of Florida East/West Circulator</td>
<td>313</td>
<td>947</td>
<td>202%</td>
<td>150%</td>
<td>100%</td>
</tr>
<tr>
<td>127</td>
<td>East Circulator (Sorority Row)</td>
<td>1,167</td>
<td>1,224</td>
<td>5%</td>
<td>100%</td>
<td>65%</td>
</tr>
<tr>
<td></td>
<td>Total Unlinked Riders (Boardings)</td>
<td>41,535</td>
<td>40,403</td>
<td>-2.7%</td>
<td>9%</td>
<td>3%</td>
</tr>
</tbody>
</table>
9 Summary and Conclusions

The model validation phase of the Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update began with data development and review which was documented in Technical Reports 2 and 3. Data review, adjustment, and correction were an iterative process throughout the model validation effort, reflecting identification of data issues based on model results.

Once the input data were initially deemed sufficient to proceed, work continued on validating each component of the Gainesville Urbanized Area Transportation Study 2010 travel demand model. An activity log was created to record all the modifications made to the model and a validation spreadsheet was prepared and used to summarize each validation run and model performance against prior versions of the Gainesville Urbanized Area Transportation Study model as well as other comparable travel demand models. This updated validation worksheet incorporates the latest Florida Department of Transportation model validation accuracy standards and allows for a complete record of all model adjustments made during the validation effort and the resulting impacts. The highway side of the model validated reasonably well from the start, largely a reflection of the efforts put forth by the study team in data development, review, disaggregation, and refinement.

The Gainesville Urbanized Area Transportation Study 2010 base year model meets most established Florida Standard Urban Transportation Model Structure standards for model accuracy and reasonableness. The validated model will be used in subsequent phases of the Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update to develop and test transportation alternatives.
Appendix A: Socioeconomic Data Format

**ZONEDATA{YEAR}.DBF**

Notes: Please note that ZONEDATA{YEAR}.DBF is part of GIS TAZ shape database of ZONEDATA.SHP, together with ZONEDATA.SHX. When editing this ZoneData file, Cube or GIS software must be used.

**Attribute List for Population and Household Data**
- **TAZ_2010** – Traffic Analysis Zone (TAZ) number in the Gainesville MTPO 2010 model.
- **SFDU** – Number of single-family dwellings units
- **SPOP** – Population in single-family dwellings units
- **MFDU** – Number of multifamily dwellings units (MFDUS)
- **MPOP** – Population in MFDUS
- **TOTPOP10** – Total population for year 2010 (this attribute is not used by model scripts, instead SPOP and MPOP are used for base year and future year scenarios)
- **HMDU** – Total hotel–motel units
- **HMPOP** – Total population in occupied hotel-motel units

**Attribute List for Employment Data**
- **OIEMP** – Other industrial employment
- **MFGEMP** – Manufacturing industrial employment
- **COMEMP** – Commercial employment
- **SERVEMP** – Service employment
- **TOTEEMP** – Total employment
- **SCHENR** – School enrollment by school location (this excludes any UF or Santa Fe College enrollment)

**Attribute List for University of Florida Data**
- **UF_EMP** – Number of UF place-of-work employees by TAZ (this variable also is used to reallocate service employment on UF Campus)
- **UF_DORM_ST** – Number of on-campus UF student residents
- **UF_PARKING** – UF commuting parking spaces, excluding on-campus student long-term
- **CLASSROOMS** – Number of UF classrooms (model scripts do not use this)
- **CLASSSQFT** – Square feet of UF classrooms (model scripts do not use this)
- **SEATS** – Number of UF classroom seats
- **UF-OC-ST** – Number of UF off-campus students, estimated from student address records provided by UF
- **SUB_AREA** – Name of city or incorporated area or Alachua if a zone is within the unincorporated area
- **UFZONES** – Identifier that indicates that a zone is on UF Campus when the value is one

**Attribute List for Transit PEV (Pedestrian Environment Variable) Data**
- **SIDEWALK** – Sidewalk availability (values vary from 0 to 3)
- **CROSSING** – Ease of street crossing (values vary from 0 to 3)
- **NONMTR_CNN** – Non-motorized connections (values vary from 0 to 3)
- **SETBACK** – Building setbacks (values vary from 0 to 3)
- **SUM** – Sum of four variable values above: SIDEWALK, CROSSING, NONMTR_CNN and SETBACK (SUM needs to be updated manually when any of four variables has been modified)
- **COMPOSIT** – composite PEV value (model scripts do not use this)
- **SELECTZONE** – Identifier that indicates that a zone is selected for select zone analysis when the value is one (the model will load selected trips that end at the selected zones, and it will be reported
in the attribute SELZONE_MOTOR in the final highway assignment output network, 
COMBINEDLOADED.NET)
HOTEL – Identifier used in the previous model (model scripts do not use this)

**Attribute List for Parking Data (previously found in ZDATA2 file in 2000 model)**
SHORTPARK – Short-term (3 hour) parking cost (cents)
LONGPARK – Long-term (8 hour) parking cost (cents)
STUDENTPAR – Student (8 hour) parking cost (cents) at UF

**Attribute List for Population and Household Variable Data (previously found in ZDATA1 file in 2000 model)**
SF_SEA – Percent SFDU not occupied by permanent residents
SF_0V – Percent households having no vehicles in SFDU occupied by permanent residents
SF_1V – Percent households having one vehicle in SFDU occupied by permanent residents
SF_2V – Percent households having two vehicles in SFDU occupied by permanent residents
SF_3V – Percent households having three or more vehicles in SFDU occupied by permanent residents
SF_VAC – Percent SFDU vacant
MF_SEA – Percent MFDU not occupied by permanent residents
MF_0V – Percent households having no vehicles in MFDU occupied by permanent residents
MF_1V – Percent households having one vehicle in MFDU occupied by permanent residents
MF_2V – Percent households having two vehicles in MFDU occupied by permanent residents
MF_3V – Percent households having three or more vehicles in MFDU occupied by permanent residents
MF_VAC – Percent MFDU vacant
HM_POC – Percent hotel-motel units occupied
Appendix B: Turn Penalties (TURN.PEN)

<table>
<thead>
<tr>
<th>Origin Node A</th>
<th>Intersection Node B</th>
<th>Destination Node C</th>
<th>Penalty Set</th>
<th>Penalty Value*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1211</td>
<td>1207</td>
<td>1221</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1207</td>
<td>1211</td>
<td>1221</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1214</td>
<td>1211</td>
<td>1218</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1214</td>
<td>1211</td>
<td>1221</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1214</td>
<td>1218</td>
<td>1211</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1240</td>
<td>1239</td>
<td>1241</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1240</td>
<td>1241</td>
<td>1238</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1240</td>
<td>1241</td>
<td>1239</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1242</td>
<td>1241</td>
<td>1238</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1241</td>
<td>1242</td>
<td>1238</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1326</td>
<td>1320</td>
<td>1324</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1326</td>
<td>1324</td>
<td>1320</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1326</td>
<td>1324</td>
<td>1325</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1328</td>
<td>1324</td>
<td>1325</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1324</td>
<td>1328</td>
<td>1325</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1338</td>
<td>1333</td>
<td>1337</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1338</td>
<td>1337</td>
<td>1333</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1338</td>
<td>1337</td>
<td>1339</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1340</td>
<td>1337</td>
<td>1339</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1337</td>
<td>1340</td>
<td>1339</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1468</td>
<td>1467</td>
<td>1472</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1468</td>
<td>1467</td>
<td>1474</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1468</td>
<td>1467</td>
<td>5404</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1472</td>
<td>1467</td>
<td>1474</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1472</td>
<td>1467</td>
<td>5404</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1467</td>
<td>1472</td>
<td>1474</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1468</td>
<td>1472</td>
<td>1467</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1468</td>
<td>1472</td>
<td>1474</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1485</td>
<td>1484</td>
<td>1483</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1485</td>
<td>1484</td>
<td>1486</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1486</td>
<td>1484</td>
<td>1483</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1485</td>
<td>1486</td>
<td>1483</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1588</td>
<td>1581</td>
<td>1585</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1588</td>
<td>1588</td>
<td>1585</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1589</td>
<td>1588</td>
<td>1593</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1589</td>
<td>1593</td>
<td>1588</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1599</td>
<td>1597</td>
<td>1601</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1599</td>
<td>1601</td>
<td>1597</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1599</td>
<td>1601</td>
<td>1602</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1599</td>
<td>1603</td>
<td>1602</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1601</td>
<td>1603</td>
<td>1602</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>1737</td>
<td>1733</td>
<td>1740</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>
**Origin Node A** | **Intersection Node B** | **Destination Node C** | **Penalty Set** | **Penalty Value**
--- | --- | --- | --- | ---
1737 | 1733 | 1744 | 1 | -1
1740 | 1733 | 1744 | 1 | -1
1733 | 1740 | 1744 | 1 | -1
1737 | 1740 | 1733 | 1 | -1
1737 | 1740 | 1744 | 1 | -1
1752 | 1750 | 1749 | 1 | -1
1752 | 1750 | 1757 | 1 | -1
1757 | 1750 | 1749 | 1 | -1
1750 | 1757 | 1749 | 1 | -1
1752 | 1757 | 1749 | 1 | -1
1752 | 1757 | 1750 | 1 | -1
1828 | 1825 | 1830 | 1 | -1
1828 | 1830 | 1825 | 1 | -1
1828 | 1830 | 1829 | 1 | -1
1831 | 1830 | 1829 | 1 | -1
1831 | 1835 | 1829 | 1 | -1
1842 | 1841 | 1846 | 1 | -1
1842 | 1846 | 1841 | 1 | -1
1842 | 1846 | 1843 | 1 | -1
1858 | 1846 | 1843 | 1 | -1
1846 | 1858 | 1843 | 1 | -1
2842 | 2841 | 2844 | 1 | -1
2842 | 2841 | 2846 | 1 | -1
2844 | 2841 | 2846 | 1 | -1
2841 | 2844 | 2846 | 1 | -1
2842 | 2844 | 2841 | 1 | -1
2842 | 2844 | 2846 | 1 | -1
2858 | 2856 | 2855 | 1 | -1
2858 | 2856 | 2859 | 1 | -1
2859 | 2856 | 2855 | 1 | -1
2856 | 2859 | 2857 | 1 | -1
2858 | 2859 | 2856 | 1 | -1
2858 | 2859 | 2857 | 1 | -1
1472 | 5356 | 5409 | 1 | -1

*Penalty value of -1 indicates a movement that is prohibited*
Appendix C: Friction Factors
<table>
<thead>
<tr>
<th>TIME</th>
<th>HBWFF</th>
<th>HBSHFF</th>
<th>HBSRFF</th>
<th>HBOFF</th>
<th>NHbff</th>
<th>TK4FF</th>
<th>TKSGFF</th>
<th>TKTRLFF</th>
<th>SOVIEFF</th>
<th>HOVIEFF</th>
<th>TKLIEFF</th>
<th>TKHTIEFF</th>
<th>HBUFF</th>
<th>HDORMUFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25208</td>
<td>126687</td>
<td>126687</td>
<td>198262</td>
<td>9231</td>
<td>9048</td>
<td>9704</td>
<td>222</td>
<td>222</td>
<td>222</td>
<td>222</td>
<td>126687</td>
<td>126687</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>21983</td>
<td>47324</td>
<td>47324</td>
<td>71259</td>
<td>8521</td>
<td>8187</td>
<td>9418</td>
<td>333</td>
<td>333</td>
<td>333</td>
<td>333</td>
<td>47324</td>
<td>47324</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>19282</td>
<td>25585</td>
<td>25585</td>
<td>37571</td>
<td>7866</td>
<td>7408</td>
<td>9139</td>
<td>444</td>
<td>444</td>
<td>444</td>
<td>444</td>
<td>25585</td>
<td>25585</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>16953</td>
<td>16092</td>
<td>16092</td>
<td>23174</td>
<td>7261</td>
<td>6703</td>
<td>8869</td>
<td>555</td>
<td>555</td>
<td>555</td>
<td>555</td>
<td>16092</td>
<td>16092</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>14924</td>
<td>10997</td>
<td>10997</td>
<td>15577</td>
<td>6703</td>
<td>6065</td>
<td>8607</td>
<td>666</td>
<td>666</td>
<td>666</td>
<td>666</td>
<td>10997</td>
<td>10997</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>13149</td>
<td>7919</td>
<td>7919</td>
<td>11056</td>
<td>6188</td>
<td>5488</td>
<td>8353</td>
<td>777</td>
<td>777</td>
<td>777</td>
<td>777</td>
<td>7919</td>
<td>7919</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>11591</td>
<td>5913</td>
<td>5913</td>
<td>8147</td>
<td>5712</td>
<td>4966</td>
<td>8106</td>
<td>888</td>
<td>888</td>
<td>888</td>
<td>888</td>
<td>5913</td>
<td>5913</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>10222</td>
<td>4534</td>
<td>4534</td>
<td>6170</td>
<td>5273</td>
<td>4493</td>
<td>7866</td>
<td>1333</td>
<td>1333</td>
<td>1333</td>
<td>1333</td>
<td>4534</td>
<td>4534</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9018</td>
<td>3548</td>
<td>3548</td>
<td>4773</td>
<td>4868</td>
<td>4066</td>
<td>7634</td>
<td>1666</td>
<td>1666</td>
<td>1666</td>
<td>1666</td>
<td>3548</td>
<td>3548</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>7957</td>
<td>2820</td>
<td>2820</td>
<td>3753</td>
<td>4493</td>
<td>3679</td>
<td>7408</td>
<td>3333</td>
<td>3333</td>
<td>3333</td>
<td>3333</td>
<td>2820</td>
<td>2820</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>7023</td>
<td>2271</td>
<td>2271</td>
<td>2991</td>
<td>4148</td>
<td>3329</td>
<td>7189</td>
<td>6666</td>
<td>6666</td>
<td>6666</td>
<td>6666</td>
<td>2271</td>
<td>2271</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>6199</td>
<td>1849</td>
<td>1849</td>
<td>2410</td>
<td>3829</td>
<td>3012</td>
<td>6977</td>
<td>7777</td>
<td>7777</td>
<td>7777</td>
<td>7777</td>
<td>1849</td>
<td>1849</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>5473</td>
<td>1519</td>
<td>1519</td>
<td>1960</td>
<td>3535</td>
<td>2725</td>
<td>6771</td>
<td>8888</td>
<td>8888</td>
<td>8888</td>
<td>8888</td>
<td>1519</td>
<td>1519</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>4833</td>
<td>1257</td>
<td>1257</td>
<td>1607</td>
<td>3263</td>
<td>2466</td>
<td>6570</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>1257</td>
<td>1257</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>4267</td>
<td>1047</td>
<td>1047</td>
<td>1326</td>
<td>3012</td>
<td>2231</td>
<td>6376</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>1047</td>
<td>1047</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>3769</td>
<td>877</td>
<td>877</td>
<td>1101</td>
<td>2780</td>
<td>2019</td>
<td>6188</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>877</td>
<td>877</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>3328</td>
<td>739</td>
<td>739</td>
<td>919</td>
<td>2567</td>
<td>1827</td>
<td>6005</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>739</td>
<td>739</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>2940</td>
<td>625</td>
<td>625</td>
<td>771</td>
<td>2369</td>
<td>1653</td>
<td>5827</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>625</td>
<td>625</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>2597</td>
<td>531</td>
<td>531</td>
<td>649</td>
<td>2187</td>
<td>1496</td>
<td>5655</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>9999</td>
<td>531</td>
<td>531</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2294</td>
<td>452</td>
<td>452</td>
<td>548</td>
<td>2019</td>
<td>1353</td>
<td>5488</td>
<td>6666</td>
<td>6666</td>
<td>6666</td>
<td>6666</td>
<td>452</td>
<td>452</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>2026</td>
<td>387</td>
<td>387</td>
<td>465</td>
<td>1864</td>
<td>1225</td>
<td>5326</td>
<td>3333</td>
<td>3333</td>
<td>3333</td>
<td>3333</td>
<td>387</td>
<td>387</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>1790</td>
<td>331</td>
<td>331</td>
<td>395</td>
<td>1720</td>
<td>1108</td>
<td>5169</td>
<td>1111</td>
<td>1111</td>
<td>1111</td>
<td>1111</td>
<td>331</td>
<td>331</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>1582</td>
<td>285</td>
<td>285</td>
<td>337</td>
<td>1588</td>
<td>1003</td>
<td>5016</td>
<td>444</td>
<td>444</td>
<td>444</td>
<td>444</td>
<td>285</td>
<td>285</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>1397</td>
<td>246</td>
<td>246</td>
<td>288</td>
<td>1466</td>
<td>907</td>
<td>4868</td>
<td>222</td>
<td>222</td>
<td>222</td>
<td>222</td>
<td>246</td>
<td>246</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>1235</td>
<td>212</td>
<td>212</td>
<td>247</td>
<td>1353</td>
<td>821</td>
<td>4724</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>111</td>
<td>212</td>
<td>212</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>1091</td>
<td>184</td>
<td>184</td>
<td>212</td>
<td>1249</td>
<td>743</td>
<td>4584</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>66</td>
<td>184</td>
<td>184</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>964</td>
<td>159</td>
<td>159</td>
<td>183</td>
<td>1153</td>
<td>672</td>
<td>4449</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>22</td>
<td>159</td>
<td>159</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>852</td>
<td>138</td>
<td>138</td>
<td>157</td>
<td>1065</td>
<td>608</td>
<td>4317</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>138</td>
<td>138</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>HBWFF</td>
<td>HBSHFF</td>
<td>HBSRFF</td>
<td>HBOFF</td>
<td>NHBFF</td>
<td>TK4FF</td>
<td>TKGLFF</td>
<td>TKR4FF</td>
<td>SOVIEFF</td>
<td>HOVIEFF</td>
<td>TKL4FF</td>
<td>TKH4FF</td>
<td>HBUFF</td>
<td>HDRMUFF</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>---------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>---------</td>
</tr>
<tr>
<td>29</td>
<td>753</td>
<td>120</td>
<td>120</td>
<td>136</td>
<td>983</td>
<td>550</td>
<td>4190</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>120</td>
<td>120</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>665</td>
<td>105</td>
<td>105</td>
<td>118</td>
<td>907</td>
<td>498</td>
<td>4066</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>105</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>588</td>
<td>92</td>
<td>92</td>
<td>102</td>
<td>837</td>
<td>450</td>
<td>3946</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>92</td>
<td>92</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>519</td>
<td>80</td>
<td>80</td>
<td>88</td>
<td>773</td>
<td>408</td>
<td>3829</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>80</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>459</td>
<td>70</td>
<td>70</td>
<td>77</td>
<td>714</td>
<td>369</td>
<td>3716</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>70</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>406</td>
<td>61</td>
<td>61</td>
<td>67</td>
<td>659</td>
<td>334</td>
<td>3606</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>61</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>358</td>
<td>54</td>
<td>54</td>
<td>58</td>
<td>608</td>
<td>302</td>
<td>3499</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>54</td>
<td>54</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>317</td>
<td>47</td>
<td>47</td>
<td>51</td>
<td>561</td>
<td>273</td>
<td>3396</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>47</td>
<td>47</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>280</td>
<td>41</td>
<td>41</td>
<td>44</td>
<td>518</td>
<td>247</td>
<td>3296</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>41</td>
<td>41</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>247</td>
<td>36</td>
<td>36</td>
<td>39</td>
<td>478</td>
<td>224</td>
<td>3198</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>36</td>
<td>36</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>219</td>
<td>32</td>
<td>32</td>
<td>34</td>
<td>442</td>
<td>202</td>
<td>3104</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>32</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>193</td>
<td>28</td>
<td>28</td>
<td>29</td>
<td>408</td>
<td>183</td>
<td>3012</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>28</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>171</td>
<td>25</td>
<td>25</td>
<td>26</td>
<td>376</td>
<td>166</td>
<td>2923</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>25</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>151</td>
<td>22</td>
<td>22</td>
<td>23</td>
<td>347</td>
<td>150</td>
<td>2837</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>22</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>133</td>
<td>19</td>
<td>19</td>
<td>20</td>
<td>321</td>
<td>136</td>
<td>2753</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>19</td>
<td>19</td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>118</td>
<td>17</td>
<td>17</td>
<td>17</td>
<td>296</td>
<td>123</td>
<td>2671</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>17</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>45</td>
<td>104</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>273</td>
<td>111</td>
<td>2592</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>15</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>92</td>
<td>13</td>
<td>13</td>
<td>13</td>
<td>252</td>
<td>101</td>
<td>2516</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>13</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>47</td>
<td>81</td>
<td>12</td>
<td>12</td>
<td>12</td>
<td>233</td>
<td>91</td>
<td>2441</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>12</td>
<td>12</td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>72</td>
<td>11</td>
<td>11</td>
<td>10</td>
<td>215</td>
<td>82</td>
<td>2369</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>11</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>49</td>
<td>64</td>
<td>9</td>
<td>9</td>
<td>9</td>
<td>198</td>
<td>74</td>
<td>2299</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>9</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>56</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>183</td>
<td>67</td>
<td>2231</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>8</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>51</td>
<td>50</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>169</td>
<td>61</td>
<td>2165</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>44</td>
<td>7</td>
<td>7</td>
<td>6</td>
<td>156</td>
<td>55</td>
<td>2101</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>7</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>39</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>144</td>
<td>50</td>
<td>2039</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>34</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>133</td>
<td>45</td>
<td>1979</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>30</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>123</td>
<td>41</td>
<td>1920</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>27</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>113</td>
<td>37</td>
<td>1864</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>TIME</td>
<td>HBWFF</td>
<td>HBSHFF</td>
<td>HBSRFF</td>
<td>HBOFF</td>
<td>NHBFF</td>
<td>TK4FF</td>
<td>TKSLFF</td>
<td>TKTRLRFF</td>
<td>SOVIEFF</td>
<td>HOVIEFF</td>
<td>TKLIEFF</td>
<td>TKHTIEFF</td>
<td>HBUFF</td>
<td>HDORMUFF</td>
</tr>
<tr>
<td>------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>-------</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>----------</td>
<td>---------</td>
<td>---------</td>
<td>---------</td>
<td>-----------</td>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td>57</td>
<td>24</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>105</td>
<td>33</td>
<td>1809</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>58</td>
<td>21</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>97</td>
<td>30</td>
<td>1755</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>59</td>
<td>19</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>89</td>
<td>27</td>
<td>1703</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>60</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>82</td>
<td>25</td>
<td>1653</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>120</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
### Appendix D: Model Flowchart, Scripts and File Locations

#### Gainesville 2010 Base Year Model Structure

<table>
<thead>
<tr>
<th>Model Step</th>
<th>File Name</th>
<th>File Format/Extension</th>
<th>File Type</th>
<th>Folder Location</th>
<th>Initial Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trip Generation</td>
<td>ATTRRATES</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Olympus</td>
</tr>
<tr>
<td></td>
<td>DUWEIGHTS</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Census 2000</td>
</tr>
<tr>
<td></td>
<td>EETARGETS</td>
<td>DBF</td>
<td>Input</td>
<td>\Base\Input</td>
<td>FTI</td>
</tr>
<tr>
<td></td>
<td>EETRIPS</td>
<td>DBF</td>
<td>Input</td>
<td>\Base\Input</td>
<td>FTI</td>
</tr>
<tr>
<td></td>
<td>INTEXT</td>
<td>DBF</td>
<td>Input</td>
<td>\Base\Input</td>
<td>I-75 Master Plan</td>
</tr>
<tr>
<td></td>
<td>PRODRATES</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>SPECGEN</td>
<td>DBF</td>
<td>Input</td>
<td>\Base\Input</td>
<td>I-75 Master Plan</td>
</tr>
<tr>
<td></td>
<td>UFData</td>
<td>DBF</td>
<td>Input</td>
<td>\Base\Input</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>ZONEDATA</td>
<td>DBF</td>
<td>Input</td>
<td>\Base\Input</td>
<td>MTPo Staff</td>
</tr>
<tr>
<td>Highway Network</td>
<td>HNET</td>
<td>NET</td>
<td>Input</td>
<td>\Base\Input</td>
<td>FDOT road characters</td>
</tr>
<tr>
<td></td>
<td>VFACTORS</td>
<td>CSV</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Olympus</td>
</tr>
<tr>
<td></td>
<td>SPDCAP</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Olympus</td>
</tr>
<tr>
<td></td>
<td>TURN</td>
<td>PEN</td>
<td>Input</td>
<td>\Base\Input</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td>Trip Distribution</td>
<td>FF</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td>Transit Network</td>
<td>AMPNR</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>TFACWKLB</td>
<td>FAC</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>TFACWKPR</td>
<td>FAC</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>TFACPNR</td>
<td>FAC</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>TFARES</td>
<td>FAR</td>
<td>Input</td>
<td>\Base\Input</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>TRANSPD</td>
<td>DBF</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Olympus</td>
</tr>
<tr>
<td></td>
<td>TROUTE</td>
<td>LIN</td>
<td>Input</td>
<td>\Base\Input</td>
<td>Gainesville 2007</td>
</tr>
<tr>
<td></td>
<td>TSYS</td>
<td>PTS</td>
<td>Parameters</td>
<td>\Parameters</td>
<td>Gainesville 2007</td>
</tr>
</tbody>
</table>
Trip Generation

GGEN000A.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
; Cube/Application Manager.
RUN PGM=GENERATION PRNFILE="\{SCENARIO_DIR\}\output\GGGEN00A.PRN" MSG='Trip Generation'
FILEI LOOKUPI[3] = "\{CATALOG_DIR\}\parameters\ARATES.DBF"
FILEI LOOKUPI[2] = "\{CATALOG_DIR\}\parameters\DUWEIGHTS.DBF"
FILEO PRINTO[3] = "\{SCENARIO_DIR\}\output\AE_ERRORS.PRN"
FILEO PRINTO[1] = "\{SCENARIO_DIR\}\output\GEN_SUM.PRN"
FILEO PAO[1] = "\{SCENARIO_DIR\}\output\PANDA_TEM.DBF",
LIST=Z,P[1],P[2],P[3],P[4],P[5],P[6],P[7],P[8],P[9],P[10],P[11],P[12],
FILEI ZDATI[3] = "\{SCENARIO_DIR\}\input\EITRIPS{Year}\alt}.DBF"
FILEI ZDATI[2] = "\{SCENARIO_DIR\}\input\SPGEN_{Year}\alt}.DBF"
FILEI ZDATI[1] = "\{SCENARIO_DIR\}\input\ZoneData\{YEAR\}.dbf",
Z-TAZ_20\{year\}

;---------------------------------------------
; OVERALL PROGRAM CONTROLS
PARAMETERS ZONES={ZONESA}, ZONEMSG=100
ARRAY CAR=4 CELL=999 CELLT=999 SFPAC=12
LOOKUP LOOKUPI=1,
  NAME=PRATE, ;TRIP PRODUCTION RATES
  LOOKUP[1]=PAD, RESULT=RATEHGW,
  LOOKUP[2]=PAD, RESULT=RATEHBSH,
  LOOKUP[3]=PAD, RESULT=RATEHSSR,
  LOOKUP[4]=PAD, RESULT=RATEHBO,

LOOKUP LOOKUPI=2,
  NAME=DUWEIGHT,
  LOOKUP[1]=SIZERANGE, RESULT=PCT1PER, ; %1 PERSON
  LOOKUP[2]=SIZERANGE, RESULT=PCT2PER, ; %2 PERSON
  LOOKUP[3]=SIZERANGE, RESULT=PCT3PER, ; %3 PERSON
  LOOKUP[4]=SIZERANGE, RESULT=PCT4PER, ; %4 PERSON
  LOOKUP[5]=SIZERANGE, RESULT=PCT5PER, ; %5 PERSON
  INTERPOLATE=N, LIST=N

LOOKUP LOOKUPI=3,
  NAME=ARATE, ;TRIP ATTRACTION RATES
  LOOKUP[1]=PURPOSE, RESULT=ARATEOIE,
  LOOKUP[2]=PURPOSE, RESULT=ARATEMFG,
  LOOKUP[3]=PURPOSE, RESULT=ARATECOM,
  LOOKUP[4]=PURPOSE, RESULT=ARATESVC,
LOOKUP[5]=PURPOSE, RESULT=ARATETOTE,
LOOKUP[6]=PURPOSE, RESULT=ARATEDUS,
LOOKUP[7]=PURPOSE, RESULT=ARATESCH,

PROCESS PHASE=ILOOP
; This is the main program loop to calculate initial production
; and attraction values for each zone for each trip purpose
; ---------------------------------------------------------------------------
LOOP HHTYPE=1,3 ; ESTABLISH VARIABLES FOR EACH HOUSEHOLD CLASS
IF (HHTYPE=1) ; SINGLE-FAMILY
  UNITS= ZI.1.SFDU
  VACRATE= ZI.1.SF_SEA+ZI.1.SF_VAC
  PERMVACRATE= ZI.1.SF_VAC
  POP= ZI.1.SPOP
ELSEIF (HHTYPE=2); MULTI-FAMILY
  UNITS= ZI.1.MFDU
  VACRATE= ZI.1.MF_SEA+ZI.1.MF_VAC
  PERMVACRATE= ZI.1.MF_VAC
  POP= ZI.1.MFPOP
ELSEIF (HHTYPE=3) ; HOTEL/MOTEL
  UNITS= ZI.1.HMDU
  VACRATE= 100-ZI.1.HM_POC
  PERMVACRATE= 100-ZI.1.HM_POC
  POP= ZI.1.HMPOP
ENDIF
; From here down, the same equations get applied to each
; household size, auto ownership and dwelling unit type.
; Since it is being run in the HHTYPE loop, the same equations
; will be applied and running totals by zone will be accumulated.
; ---------------------------------------------------------------------------
VAC=UNITS*(VACRATE/100), OCC=UNITS-VAC, GENVAC=UNITS*(PERMVACRATE/100), GENOCC=UNITS-GENVAC
IF (HHTYPE<>3) TOCC=TOCC+OCC ;Keep track of total permanently occupied DUs
TGOC=TGOC+GENOCC ;Keep track of total occupied DUs
IF (OCC<>0)
  POPDU=POP/OCC
ELSE
  POPDU=0
ENDIF
IF (POPU<0) 
ENDIF
IF (PODU<0) 
IF (PODU<1.12) RANGE=1
IF (PODU>1.12) RANGE=2
IF (PODU>1.37) RANGE=3
IF (PODU>1.62) RANGE=4
IF (PODU>1.87) RANGE=5
IF (PODU>2.12) RANGE=6
IF (PODU>2.37) RANGE=7
IF (PODU>2.62) RANGE=8
IF (PODU>2.87) RANGE=9
IF (PODU>3.12) RANGE=10
IF (PODU>3.37) RANGE=11
IF (PODU>3.62) RANGE=12
IF (PODU>3.87) RANGE=13
IF (PODU>4.12) RANGE=14
IF (PODU>4.37) RANGE=15
IF (PODU>4.62) RANGE=16
IF (PODU>5.99) RANGE=17
if (POPDU<1&OCC>0) PRINT LIST="POP/DU ERROR, HHTYPE=",HHTYPE(1.0)," Population=",POP(4.0C)," Occupied Units=",occ(4.0c)," TAZ=",TAZ_2010(4.0c) printo=2
LOOP PR=1,5
  LOOP AU=1,4
    CELL[CL]=GENOCC*DUWEIGHT(PR,RANGE)*CAR[AU]
  ENDLOOP
  LOOP PURP=1,4
    PRODRATE=PRATE(PURP,CL)
    P[PURP]=P[PURP]+PRODRATE(PURP,CL)*CELL[CL]
  ENDLOOP
ENDLOOP

;==============================================================================
; pre-process prior to attractions calculation
; minor employment adjustment using UF_EMP Data
; (1) Subtract UF employment from service employment if UF<Service
COMEMP=ZI.1.COMEMP
SERVEREMP=ZI.1.SERVEREMP
IF(ZI.1.SERVEREMP>ZI.1.UF_EMP)
  SERVEMP=ZI.1.SERVEREMP-ZI.1.UF_EMP
Else
  IF(ZI.1.UF_EMP>0)
  PRINT LIST='EMPLOYMENT ZONE', I
ENDIF
ENDIF
ENDIF
ELSE
; (2) Take the remaining UF service employment from commercial if UF>Service
COMEMP=ZI.1.COMEMP-ZI.1.UF_EMP+ZI.1.SERVEREMP
SERVEREMP = 0
IF (COMEMP<0)
  COMEMP=0
ENDIF
ENDIF

; (3) Compute total UF parking and employment
; replaced hard-coded zone number below
UF_EMP1=0
UF_PRK1=0
LOOP II=1,{ZONESA}
  UF_EMP1=UF_EMP1+ZI.1.UF_EMP[II]
  UF_PRK1=UF_PRK1+ZI.1.UF_PARKING[II]
ENDLOOP
PRINT LIST=UF_EMP1, '   ', UF_PRK1

; (4) Allocate UF employment to parking TAZs based on proportion of parking spaces
SERVEREMP= ZI.1.SERVEREMP*ZI.1.UF_EMP1*(ZI.1.UF_PARKING[II]/UF_PRK1)
TOTALEMP=ZI.1.MFGEMP+ZI.1.OIEMP+COMEMP+SERVEREMP

;==============================================================================
; Now process the trip purposes that are attraction-based
; PURPOSE 1 = HBW
; PURPOSE 2 = HBSH
; PURPOSE 3 = HBSR
; PURPOSE 4 = HBO
; PURPOSE 5 = NHB
; PURPOSE 6 = 4 Tire Truck
; PURPOSE 7 = Single-Unit Truck
; PURPOSE 8 = Tractor-trailer
TOTALDUS=ZI.1.SFDU+ZI.1.MFDU
LOOP WPURP=1,8
  A[WPURP]=ARATE(1,WPURP)*ZI.1.MFGEMP+
             ARATE(2,WPURP)*ZI.1.OIEMP+
             ARATE(3,WPURP)*COMEMP+
             ARATE(4,WPURP)*SERVEREMP+
\[ \text{ARATE}(5, \text{WPURP}) \times \text{TOTALEMP} + \]
\[ \text{ARATE}(6, \text{WPURP}) \times \text{TOTALDUS} + \]
\[ \text{ARATE}(7, \text{WPURP}) \times \text{ZI.1.SCHENR} \]

\text{ENDLOOP}

; .................................................................
; For the next four purposes,
; Attractions are a function of the total attractions to a zone.
; Since totals aren't known until we finish the initial calculations,
; attractions for these purposes will be initially calculated in the
; \text{ADJUST PHASE}.
; .................................................................

\text{PURPOSE 9} = \text{SOV EI}
\text{PURPOSE 10} = \text{HOV EI}
\text{PURPOSE 11} = \text{LDTK EI}
\text{PURPOSE 12} = \text{HDTK EI}

\[ P[9] = \text{ZI.3.TRIPS} \times (\text{ZI.3.LOVPCT}/100) \]
\[ P[10] = \text{ZI.3.TRIPS} \times (\text{ZI.3.HOVPCT}/100) \]
\[ P[11] = \text{ZI.3.TRIPS} \times (\text{ZI.3.LDTPCT}/100) \]
\[ P[12] = \text{ZI.3.TRIPS} \times (\text{ZI.3.HDTPCT}/100) \]

; \text{NOW PROCESS SPECIAL GENERATORS}
\text{SPFRAC}[1] = \text{ZI.2.HBWP}/100
\text{SPFRAC}[2] = \text{ZI.2.HBSHP}/100
\text{SPFRAC}[3] = \text{ZI.2.HBSRP}/100
\text{SPFRAC}[4] = \text{ZI.2.HBOP}/100
\text{SPFRAC}[5] = \text{ZI.2.NHBP}/100
\text{SPFRAC}[6] = \text{ZI.2.TRK4P}/100
\text{SPFRAC}[7] = \text{ZI.2.TRKSUNITP}/100
\text{SPFRAC}[8] = \text{ZI.2.TRKCOMMOP}/100
\text{SPFRAC}[9] = \text{ZI.2.EILOVP}/100
\text{SPFRAC}[10] = \text{ZI.2.EIHOVP}/100
\text{SPFRAC}[11] = \text{ZI.2.EILDTP}/100
\text{SPFRAC}[12] = \text{ZI.2.EIHDTP}/100

\text{LOOP PRP}=1,12
\text{IF} (\text{ZI.2.PROD}='Y', 'y')
\text{IF} (\text{ZI.2.FUNCTIONP}='+'') \ P[PRP]-P[PRP]+VALUEP*SPFRAC[PRP]
\text{IF} (\text{ZI.2.FUNCTIONP}='-'') \ P[PRP]-P[PRP]-VALUEP*SPFRAC[PRP]
\text{ENDIF}
\text{ENDLOOP}

\text{SPFRAC}[1] = \text{ZI.2.HBWA}/100
\text{SPFRAC}[2] = \text{ZI.2.HBSHIA}/100
\text{SPFRAC}[3] = \text{ZI.2.HBSRA}/100
\text{SPFRAC}[4] = \text{ZI.2.HBOA}/100
\text{SPFRAC}[5] = \text{ZI.2.NHBA}/100
\text{SPFRAC}[6] = \text{ZI.2.TRK4A}/100
\text{SPFRAC}[7] = \text{ZI.2.TRKSUNITA}/100
\text{SPFRAC}[8] = \text{ZI.2.TRKCOMMOP}/100
\text{SPFRAC}[9] = \text{ZI.2.EILOVA}/100
\text{SPFRAC}[10] = \text{ZI.2.EIHOVA}/100
\text{SPFRAC}[11] = \text{ZI.2.EILDTP}/100
\text{SPFRAC}[12] = \text{ZI.2.EIHDTA}/100

\text{LOOP PRP}=1,12
\text{IF} (\text{ZI.2.ATTR}='Y', 'y')
\text{IF} (\text{ZI.2.FUNCTIONA}='+'') \ A[PRP]-A[PRP]+VALUEA*SPFRAC[PRP]
\text{IF} (\text{ZI.2.FUNCTIONA}='-'') \ A[PRP]-A[PRP]-VALUEA*SPFRAC[PRP]
ENDIF
ENDLOOP

; ******************************************************
; This portion of the script checks to see if any zones with populations are lacking values for percent automobile ownership. If so, the model crashes and reports the problem zones so that the user can correct the problem. All zones with populations should have values for percent automobile ownership or the model will not generate Home-Based trips for those zones.

IF (I!=1)
PRINT LIST="\nAUTO OWNERSHIP ERRORS WHERE POPULATION EXISTS BUT AUTO OWNERSHIP DOES NOT", PRINTO=3
PRINT LIST="\nCHECK LISTED ZONES IN ZONEDATA{Year} FILES FOR AUTO OWNERSHIP PERCENTAGES!!!", PRINTO=3
SFAOERROR=0
MFAOERROR=0
ENDIF
SFAO=zi.1.SF_0V+zi.1.SF_1V+zi.1.SF_2V+zi.1.SF_3V
MFAO=zi.1.MF_0V+zi.1.MF_1V+zi.1.MF_2V+zi.1.MF_3V
IF ((zi.1.SPOP<>0 & SFAO=0)|(zi.1.MFPOP<>0 & MFAO=0))
PRINT LIST="\n", PRINTO=3
ENDIF
IF (zi.1.SPOP<>0 & SFAO=0)
SFAOERROR=SFAOERROR+1
PRINT LIST="\nAUTO OWNERSHIP = 0 BUT SF POPULATION > 0 ERROR FOR ZONE=",I(5.0),PRINTO=3
ELSE
ENDIF
IF (zi.1.MFPOP<>0 & MFAO=0)
MFAOERROR=MFAOERROR+1
PRINT LIST="\nAUTO OWNERSHIP = 0 BUT MF POPULATION > 0 ERROR FOR ZONE=",I(5.0),PRINTO=3
ELSE
ENDIF
IF (I={ZONESA})
PRINT LIST="\n************Error Report Summary************",
'\nTOTAL AUTO OWNERSHIP ERRORS FOR SINGLE FAMILY=',SFAOERROR(8.0C),
'\nTOTAL AUTO OWNERSHIP ERRORS FOR MULTI FAMILY=',MFAOERROR(8.0C), printo=3
SFAOERROR=0
MFAOERROR=0 ;set both to 0 in order to success executive by Mia
IF (SFAOERROR=0 & MFAOERROR=0) PRINT LIST="\n',
'\nTHERE ARE NO AUTO OWNERSHIP = 0 BUT POPULATION > 0 ERRORS', PRINTO=3
if (SFAOERROR>1) abort
if (MFAOERROR>1) abort
ENDIF

; ******************************************************
ENDPROCESS
PROCESS PHASE=ADJUST
LOOP PURP=1,12
IF (PURP=1) PRINT LIST="TRIP PRODUCTION AND ATTRACTION REPORT BY PURPOSE", PRINTO=1
PRINT LIST=" Purpose="PURP(2.0)," Productions="P[PURP][0](12.0C)," Unbalanced Attractions="A[PURP][0](12.0C), PRINTO=1
ENDLOOP
PRINT LIST="", PRINTO=1
; Balancing attractions as similarly done in Olympus model.
BALANCE A2P=1-4
/ .................................................................
BALANCE A2P=9-12
LOOP PURP=1,12
PRINT LIST=" Purpose"," Productions="P[PURP][0] (12.0C)," Balanced Attractions="A[PURP][0] (12.0C), PRINTO=1
ENDLOOP
PRINT LIST=" Total"," Productions="PTOTAL (12.0C)," Balanced Attractions="ATOTAL (12.0C), PRINTO=1
PRINT LIST="", PRINTO=1
POPTOT=ZI.1.SPOP[0]+ZI.1.MFPOP[0]
ALLPOP=POPTOT+ZI.1.HMPOP[0]
PDUAVG=(POPTOT/TOCC)
ALPDAG=(ALLPOP/TGCC)
TOTSRV=ZI.1.SRVEMP[0]
TOTCON=ZI.1.COMEMP[0]
TOTMFG=ZI.1.MFEMP[0]
TOTIND=ZI.1.OIEMP[0]
TOTEMP=ZI.1.TOTEMP[0]
EMPPOT=TOTEMP/POPTOT
SRVRTE=TOTSRV/TOCC
COMRTE=TOTCON/TOTEMP
MFRGTE=TOTMFG/TOCC
INDRTE=TOTIND/TOCC
IITRIP=P[1][0]+P[2][0]+P[3][0]+P[4][0]+P[5][0]+P[6][0]+P[7][0]+P[8][0]
ITPPRM=IITRIP/TOCC
ITPTOC=IITRIP/TGCC
ITPEMP=IITRIP/TOCC
PRINT LIST=" Permanent Population = ",POPTOT(12.0C), PRINTO=1
PRINT LIST=" Total Population = ",ALLDPOP(12.0C), PRINTO=1
PRINT LIST=" Permenantley Occupied Dwelling Units = ",TOCC(12.0C), PRINTO=1
PRINT LIST=" Transient and Permenantly Occupied Dwelling Units = ",TGCOCC(12.0C), PRINTO=1
PRINT LIST=" Total Service Employment = ",TOTSRV(12.0C), PRINTO=1
PRINT LIST=" Total Commercial Employment = ",TOTCOM(12.0C), PRINTO=1
PRINT LIST=" Total Manufacturing Employment = ",TOTMFG(12.0C), PRINTO=1
PRINT LIST=" Total Other Industrial Employment = ",TOTIND(12.0C), PRINTO=1
PRINT LIST=" Total Employment = ",TOTEPM(12.0C), PRINTO=1
PRINT LIST=" Permanent Population per Permenantly Occupined Dwelling Unit = ",PDUAVG(5.2C), PRINTO=1
PRINT LIST=" Total Population per Total Occupied Dwelling Unit = ",ALPDAG(5.3C), PRINTO=1
PRINT LIST=" Total Employment per Permanent Population = ",EMPPOT(5.3C), PRINTO=1
PRINT LIST=" Service to Total Employment = ",SRVRTE(5.3C), PRINTO=1
PRINT LIST=" Commercial to Total Employment = ",COMRTE(5.3C), PRINTO=1
PRINT LIST=" Manufacturing to Total Employment = ",MFRGTE(5.3C), PRINTO=1
PRINT LIST=" Other Industrial to Total Employment = ",INDRTE(5.3C), PRINTO=1
PRINT LIST=" Internal Person Trips per Permenantly Occupied Dwelling Unit = ",ITPPRM(5.3C), PRINTO=1
PRINT LIST=" Internal Person Trips per Total Occupied Dwelling Units = ",ITPTOC(5.3C), PRINTO=1
PRINT LIST=" Internal Person Trips per Employee = ",ITPEMP(5.3C), PRINTO=1
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

PRINT LIST=" Internal Person Trips per Person - ",ITPPOP(5.3C),
PRINTO=1
ENDPROCESS
ENDRUN

GNMAT00C.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM= MATRIZ PRNFILE="{SCENARIO_DIR}\output\UFGEN.prn" MSG='UF Trip Generation'
FILEI ZDAT[1] = "{SCENARIO_DIR}\input\ZoneData\YEAR.DBF",
Z=TAZ_20(year)
FILEO RECO[1] = "{SCENARIO_DIR}\output\UFPANDA.dbf",
FIELDS=Z,HBUP,HBUA,HDORMUP,HDORMUA,STUPCT,nocarpct,wcarpct
;FILEO PRINTO[1] = "{SCENARIO_DIR}\output\GEN_SUM.PRN"
PAR ZONES=\ZONESA\ ;
; Trip rates from HH survey
; Off-campus student trips
RO.HBUP = (RATE_HBUP)*ZI.1.UF_OC_ST*(HBO-TF) ; home-based university PRODS from off-campus
(students)
RO.HBUA = (RATE_HBUA)*ZI.1.UF_PARKING*(HBO-TF) ; home-based university ATTRS from off-campus
(parking spaces)
; Campus housing student trips
RO.HDORMUP = (RATE_HDORMUP)*ZI.1.UF_DORM_ST*(HBO-TF) ; home-based university PRODS from Campus
housing (students)
RO.HDORMUA = (RATE_HDORMUA)*ZI.1.SEATS*(HBO-TF) ; home-based university ATTRS from classroom
seats
ufpop = ZI.1.UF_OC_ST + ZI.1.UF_DORM_ST ; UF pop is equal to number of off-campus students plus
dorm students
sfpop = ZI.1.SPOP
mfpop = ZI.1.MFFP
tpop=sfpop+mfpop
sf0 = 0.01*ZI.1.SF_0V
mf0 = 0.01*ZI.1.MF_0V

;Student market share
if (tpop>0)
   RO.STUPCT=ufpop/tpop
   t0=(sf0*sfpop + mf0*mfpop)/tpop
else
   RO.STUPCT=0.0
   t0=0.0
endif
if (STUPCT>1.0) STUPCT=1.0 ; make sure fraction students not greater than 1.0
nocarpct = t0*1.0-STUPCT ; fraction without autos
wcarpct = 1.0-nocarpct-STUPCT ; fraction with autos
WRITE RECO=1
ENDRUN
GNMAT00D.S

; Do not change filenames or add or remove FILEI/FILO statements using an editor. Use Cube/Application Manager.
RUN PGM-MATRIX PRNFILE="\{SCENARIO\_DIR\}/output\GNMAT00B.PRN" MSG="Aggregation of Overall PANDA and UF Purpose"
FILEI ZDATI[2] = "\{SCENARIO\_DIR\}/output\UFFANDA.dbf"
FILEI ZDATI[1] = "\{SCENARIO\_DIR\}/output\PANDA\_TEM.DBF"

FILEO RECO[1] = "\{SCENARIO\_DIR\}/output\PANDA.DBF",
FIELDS=Z,HBWP,HBWA,HBSHP,HBSHA,HBSRF,HBSRA,HBOA,NHBF,NHBA,TK4P,TK4A,
SGLUNITP,SGLUNITA,TRKTRLRP,TRKTRLRA,SOVIEP,SOVIEA,HOVIEP,HOVIEA,LDTKIEP,LDTKIEA,HDTKIEP,HDTKIEA,
HBUP,HBUA,HDORMUP,HDORMUA

PAR ZONES=\{ZONESA\}
RO.HBWP=ZI.1.P1
RO.HBSHP=ZI.1.P2
RO.HBSRF=ZI.1.P3
RO.HBOP=ZI.1.P4
RO.NHBF=ZI.1.P5
RO.TK4P=ZI.1.P6
RO.SGLUNITP=ZI.1.P7
RO.TRKTRLRP=ZI.1.P8
RO.SOVIEP=ZI.1.P9
RO.HOVIEP=ZI.1.P10
RO.LDTKIEP=ZI.1.P11
RO.HDTKIEP=ZI.1.P12
RO.HBWA=ZI.1.A1
RO.HBSHA=ZI.1.A2
RO.HBSRA=ZI.1.A3
RO.HBOA=ZI.1.A4
RO.NHBA=ZI.1.A5
RO.TK4A=ZI.1.A6
RO.SGLUNITA=ZI.1.A7
RO.TRKTRLRA=ZI.1.A8
RO.SOVIEA=ZI.1.A9
RO.HOVIEA=ZI.1.A10
RO.LDTKIEA=ZI.1.A11
RO.HDTKIEA=ZI.1.A12
RO.HBUP=ZI.2.hbup
RO.HBUA=ZI.2.hbua
RO.HDORMUP=ZI.2.hdormup
RO.HDORMUA=ZI.2.hdormua
WRITE RECO=1

ENDRUN
EEMAT00C.S

; Do not change filenames or add or remove FILEI/FILIO statements using an editor. Use
Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="(SCENARIO_DIR)\output\EEMAT00c.PRN" MSG='Parameter Base Year EE Trip
Matrix Development (Do Not Edit)''
FILEO MATO(1) = "(CATALOG_DIR)\PARAMETERS\BASEYEAR_EETRIPS_DIST.MAT",
MO=1
FILEI MATI[1] = "(CATALOG_DIR)\PARAMETERS\BASEYEAR_EETRIPS_IJ.DBF",
   pattern-ijm:v, fields-orz,dsz,0,autotrips
PAR zones={ZONESA}
mw(1)=mi.1.1
ENDRUN
**EEFRA00B.S**

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=FRATAR PRNFILE="\{SCENARIO_DIR\}\output\EEFRA00A.PRN" MSG='External External Trip Matrix Development' FILEI MAT[1] - "\{CATALOG_DIR\}\PARAMETERS\BASEYEAR_EETRIPS_DIST.MAT"
FILEO MATO[1] - "\{SCENARIO_DIR\}\output\EETAB.MAT",
MO=1, name=EETRIPS
FILEI ZDAT[1] - "\{SCENARIO_DIR\}\INPUT\eeTARGET\20\{YEAR\}.dbf"
MAXITERS=99
ACOMP=1, PCOMP=1
MARGINS=1
ENDRUN

**Highway Network Step**
HNMAT00B.S

Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.

RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\Output\SPDCAP.OUT" MSG='SPDCAP file'
FILEO PRINTO[2] = "{SCENARIO_DIR}\OUTPUT\SPDCAP.ERR"
FILEO PRINTO[1] = "{SCENARIO_DIR}\output\SPDCAP.CSV"
FILEI RECI = "(CATALOG_DIR)\PARAMETERS\SPDCAP.DBF"

ARRAY SPDLOOKUP=999999 CAPLOOKUP=999999
_LATVAL=RI.LOW_ATYPE
_HATVAL=RI.HIGH_ATYPE
_LFTVAL=RI.LOW_FTYPE
_HFTVAL=RI.HIGH_FTYPE
_LLVAL=RI.LOW_LANES
_HLVAL=RI.HIGH_LANES
_CAPVAL=RI.CAPACITY
_SPDVAL=RI.SPEED
_CAPFUNC=RI.CAP_OPERAN
_SPDFUNC=RI.SPEED_OPER

; PLACE INITIAL CAPACITIES & SPEEDS INTO AN ARRAY
IF (_CAPFUNC=' ')
  LOOP ATYPE=_LATVAL,_HATVAL
    LOOP FTYPE=_LFTVAL,_HFTVAL
      LOOP LANES=_LLVAL,_HLVAL
        INDEXVAL=ATYPE*10000+FTYPE*100+LANES
        CAPLOOKUP[INDEXVAL]=_CAPVAL
      ENDLOOP
    ENDLOOP
  ENDLOOP
ENDIF

IF (_SPDFUNC=' ')
  LOOP ATYPE=_LATVAL,_HATVAL
    LOOP FTYPE=_LFTVAL,_HFTVAL
      LOOP LANES=_LLVAL,_HLVAL
        INDEXVAL=ATYPE*10000+FTYPE*100+LANES
        SPDLOOKUP[INDEXVAL]=_SPDVAL
      ENDLOOP
    ENDLOOP
  ENDLOOP
ENDIF

IF (_CAPFUNC='*')
  LOOP ATYPE=_LATVAL,_HATVAL
    LOOP FTYPE=_LFTVAL,_HFTVAL
      LOOP LANES=_LLVAL,_HLVAL
        INDEXVAL=ATYPE*10000+FTYPE*100+LANES
        CAPLOOKUP[INDEXVAL]=CAPLOOKUP[INDEXVAL]*_CAPVAL
      ENDLOOP
    ENDLOOP
  ENDLOOP
ENDIF

IF (_SPDFUNC='*' | _SPDFUNC='+' | _SPDFUNC='-')
  LOOP ATYPE=_LATVAL,_HATVAL
    LOOP FTYPE=_LFTVAL,_HFTVAL
      LOOP LANES=_LLVAL,_HLVAL
        INDEXVAL=ATYPE*10000+FTYPE*100+LANES
        SPDLOOKUP[INDEXVAL]=SPDLOOKUP[INDEXVAL]+_SPDVAL
      ENDLOOP
    ENDLOOP
  ENDLOOP
ENDIF
INDEXVAL=ATYPE*10000+TYPE*100+LANES
IF (_SPDFUNC='*') SPDLOOKUP[INDEXVAL]=SPDLOOKUP[INDEXVAL]* SPDVAL
IF (_SPDFUNC='*') SPDLOOKUP[INDEXVAL]=SPDLOOKUP[INDEXVAL]+SPDVAL
ENDLOOP
ENDLOOP
ENDIF

PRINT "SPEED OR CAPACITY ERRORS WHERE THE SPD CAP RESULT IS LESS THAN ZERO", PRINTO=2
LOOP IVAL=1,999999
   PRINT CSV=T, LIST=IVAL(6.0),CAPLOOKUP[IVAL],SPDLOOKUP[IVAL],PRINTO=1
   IF (CAPLOOKUP[IVAL]>0) CAPERRCNT=CAPERRCNT+1
   IF (SPDLOOKUP[IVAL]<0) SPFERRCNT=SPFERRCNT+1
ENDLOOP
PRINT CSV=T, LIST='SPD CAP ERROR FOR ATFTLN=',IVAL(6.0),'
CAPACITY=',CAPLOOKUP[IVAL](9.2),PRINTO=2
ENDIF
IF (SPDLOOKUP[IVAL]<0)
   SPFERRCNT=SPFERRCNT+1
   PRINT CSV=T, LIST='SPD CAP ERROR FOR ATFTLN=',IVAL(6.0),'
SPEED=',SPDLOOKUP[IVAL](9.2),PRINTO=2
ENDIF
ENDLOOP
PRINT LIST='\n**********Error Report Summary**********',
   '\nTOTAL LESS THAN ZERO CAPACITY ERRORS=',CAPERRCNT(8.0C),
   '\nTOTAL LESS THAN ZERO SPEED ERRORS =',SPFERRCNT(8.0C),printo=2
ENDIF
ENDRUN

HNNET00B.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM=NETWORK PRNFILE="{SCENARIO_DIR}\Output\HNNET00A.FRN" MSG='Unloaded Network Development'
FILEI LOOKUP[1] = "{CATALOG_DIR}\parameters\VFACTORS.CSV"
FILEO PRINTO(2) = "{SCENARIO_DIR}\Output\NETERRORS.PRN"
FILEI LINK[1] = "{SCENARIO_DIR}\INPUT\HNET20{YEAR}.NET"
FILEI LOOKUP[2] = "{SCENARIO_DIR}\output\SPDCAP.CSV"
FILEO PRINTO[1] = "{SCENARIO_DIR}\output\NODECOORD.CSV"
FILEO NETO = "{SCENARIO_DIR}\OUTPUT\UNLOADED.NET",
EXCLUDE-LINKCNT
PAR LIST ERRS=0 MAX IP ERRS=10000
ARRAY _ATCNT-99, _FTCNT-99
LOOKUP, NAME=VFACTORS,
   LOOKUP[1]=1, RESULT=2,
   LOOKUP[2]=1, RESULT=3,
LOOKUP[3]=1, RESULT=4,
LOOKUP[4]=1, RESULT=5,
INTERPOLATE=N, LOOKUPI=1
LOOKUP, NAME=SPDCAP,
LOOKUP[1]=1, RESULT=2,
LOOKUP[2]=1, RESULT=3,
INTERPOLATE=N, LOOKUPI=2

PROCESS PHASE=INPUT
;Use this phase to modify data as it is read, such as recoding node numbers.
ENDPROCESS

PROCESS PHASE=NODEMERGE
   print csv=t list=N(6.0),X,Y, PRINTO=1
ENDPROCESS

PROCESS PHASE=LINKMERGE
   COMP FTYPE=LI.1.FTYPE
   COMP FTYPE1=INT(LI.1.FTYPE/10)
   COMP ATYPE=LI.1.ATYPE
   COMP ATYPE1=INT(LI.1.ATYPE/10)
   COMP LANES=LI.1.LANES

   IF (DISTANCE<=0)
      DISTANCE=SQRT((A.X-B.X)^2+(A.Y-B.Y)^2)/{UNITS}
   endif
   _MYDIST=SQRT((A.X-B.X)^2+(A.Y-B.Y)^2)/{UNITS}
   _err=(_MYDIST-DISTANCE)/DISTANCE
   if(_err >0.01) print list=A,B,_MYDIST(8.4),DISTANCE(8.4) PRINTO=2

   ; PUT VFACTORS ON NETWORK
   linkcnt=1
   UROADFACTOR=VFACTORS(1,FTYPE)
   CONFAC=VFACTORS(2,FTYPE)
   BPRCOEFFICIENT=VFACTORS(3,FTYPE)
   BPREXponent=VFACTORS(4,FTYPE)
   ; PUT SPEEDS AND CAPACITIES ON NETWORK
   _INDEXVAL=10000*ATYPE+100*FTYPE+LANES
   CAPACITY=SPDCAP(1,_INDEXVAL)*LANES
   IF (CAPACITY=0)
      DAILYCAP=999999
   ELSE
      DAILYCAP=(CAPACITY/CONFAC)*UROADFACTOR
   ENDIF
   SPEED=SPDCAP(2,_INDEXVAL)
   IF (SPEED!=0)
      TIME=60*DISTANCE/SPEED
   endif
   if (time<0.01) time=0.01
   ; PUT WALKTIME ON NETWORK
   WALKTIME=DISTANCE/2.5*60
   _ATCNT[ATYPE]=_ATCNT[ATYPE]+1
   _FTCNT[FTYPE]=_FTCNT[FTYPE]+1
   ; Put Bike Speed and Time on network
   _spd_red=0
   _ln_red=0
   if (SPEED>12)
      _spd_red=(SPEED-12)/18
   endif
   if (LANES=2)
      _ln_red=1
   endif
if (L == 0) DELETE
ENDPROCESS
PROCESS PHASE=SUMMARY
; Use this phase for combining and reporting of working variables.

ENDPROCESS
ENDRUN

HNNHWW00A.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cubi/Application Manager.
RUN FGM=HIGHWAY FRNFILE="{SCENARIO_DIR}\Output\HNHWY00A.FRNAME" MSG='Travel Time Skim'
FILEI NETI = "{SCENARIO_DIR}\Output\UNLOADED.NET"
FILEO MATO[1] = "{SCENARIO_DIR}\OUTPUT\HSKIMS.{ALT}\{YEAR}.MAT",
MO=1,2,10,99,3 NAME=TIME,DISTANCE,TERMINALTIME,WALKDISTANCE,BIKETIME, DEC=4*3
FILEO TURNPENI = "{SCENARIO_DIR}\INPUT\TCARDS.PEN"
ARRAY TERMTIME=59 TERMTIME=(ZONESA)
PAR ZONEMSG=100
TERM[1]=(TERM10)
TERM[2]=(TERM20)
TERM[3]=(TERM30)
TERM[4]=(TERM40)
TERM[5]=(TERM50)

PROCESS PHASE=LINKREAD
IF (A-1=(ZONESA)) TERMTIME(A)=TERM[LI.TYPEI]
IF (LI.FTYPE=10-19,49,70-99) ADDTOGROUP=1 ; no walk on freeways, etc.
ENDPROCESS

PROCESS PHASE=ILOOP
PATHLOAD PATH=LI.TIME,
   MW[1]=PATHTRACE(LI.TIME,1),NOACCESS=99999,
   MW[2]=PATHTRACE(LI.DISTANCE),NOACCESS=99999, PENI=1
PATHLOAD PATH=LI.BK_TIME,EXCLUDEGROUP=1,
   MW[3]=PATHTRACE(LI.BK_TIME),NOACCESS=99999
MW[1][I]=LOWEST(1,2,4) ; INTRAZONAL TIME = 1/2 THE AVERAGE OF THE TWO NEAREST ZONES
MW[2][I]=LOWEST(2,2,4) ; INTRAZONAL DISTANCE = 1/2 THE AVERAGE OF THE TWO NEAREST ZONES
MW[3][I]=LOWEST(3,2,4) ; INTRAZONAL BIKE TIME = 1/2 THE AVERAGE OF THE TWO NEAREST ZONES
MW[10]=TERMTIME(I)+TERMTIME(J) ; BUILDS TERMINAL TIME MATRIX
PATHLOAD PATH=LI.DISTANCE, MW[99]=PATHTRACE(LI.DISTANCE), EXCLUDEGROUP=1
MW[99][I]=ROWMIN(99)
ENDPROCESS
PROCESS PHASE=ADJUST
ENDPROCESS
ENDRUN
Trip Distribution Step

DTMAT00C.S

; Do not change filenames or add or remove FILEI/FILOO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIEX PRNFILE="{SCENARIO_DIR}\Output\DTMAT00A.FRNL MSG='Convert FF File to CSV'
FILEO PRINTO[1] = "{SCENARIO_DIR}\Output\FF.CSV"
FILEI RECI = "{CATALOG_DIR}\PARAMETERS\FF.DBF"

print csv=t,
list=ri.time,ri.hbwwf,ri.hbshff,ri.hbrsff,ri.hboff,ri.hbslff,ri.tk4ff,ri.tksllff,ri.tklslff,
ri.sovieff,ri.hovieff,ri.tkltieff,ri.tkhrlf,ri.hbiff,ri.hdormff, printo-1
ENDRUN

DTOST00A.S

; Do not change filenames or add or remove FILEI/FILOO statements using an editor. Use Cube/Application Manager.
RUN PGM=DISTRIBUTION PRNFILE="{SCENARIO_DIR}\output\DISTRIP.FRN" MSG='Distribution'
FILEO MATO[1] = "{SCENARIO_DIR}\Output\PTRIPS.MAT",
MO=1-14,
NAME=HBB,HBSS,HBSSR,HBO,HBB,TRUK4,TRUKSR,TRUCKTLR,SOVIE,HOVIE,TRUCKDIE,TRUCKHDIE,HBU,HDORMU
FILEI ZDAT[1] = "{SCENARIO_DIR}\output\PANDA.DBF"
FILEI MATI[1] = "{SCENARIO_DIR}\Output\FMSXIMS_{ALT}{YEAR}.MAT"
FILEI LOOKIU[1] = "{SCENARIO_DIR}\Output\FF.CSV"

PAR ZOONMSG=100, MAXRMSE=.001, MAXITERs=50


MW(50)-MI.1.TIME+MI.1.TEMLTIME

LOOKUP, NAME=FF,
    LOOKUP(1)=1, RESULT=2,
    LOOKUP(2)=1, RESULT=3,
    LOOKUP(3)=1, RESULT=4,
    LOOKUP(4)=1, RESULT=5,
    LOOKUP(5)=1, RESULT=6,
    LOOKUP(6)=1, RESULT=7,
    LOOKUP(7)=1, RESULT=8,
    LOOKUP(8)=1, RESULT=9,
    LOOKUP(9)=1, RESULT=10,
    LOOKUP(10)=1, RESULT=11,
    LOOKUP(11)=1, RESULT=12,
    LOOKUP(12)=1, RESULT=13,
    LOOKUP(13)=1, RESULT=14,
    LOOKUP(14)=1, RESULT=15,
    INTERPOLATE=Y, lookup=1

GRAVITY LOS=MW(50), PURPOSE=1, FACTORS=FF ; HBW
GRAVITY LOS=MW(50), PURPOSE=2, FACTORS=FF ; HBSH
GRAVITY LOS=MW(50), PURPOSE=3, FACTORS=FF ; HBSR
GRAVITY LOS=MW(50), PURPOSE=4, FACTORS=FF ; HBO
GRAVITY LOS=MW(50), PURPOSE=5, FACTORS=FF ; NHB
GRAVITY LOS=MW(50), PURPOSE=6, FACTORS=FF ; TK4
GRAVITY LOS=MW(50), PURPOSE=7, FACTORS=FF ; SGLUNIT
GRAVITY LOS=MW(50), PURPOSE=8, FACTORS=FF ; TRKTLR
GRAVITY LOS=MW(50), PURPOSE=9, FACTORS=FF ; SOVIE
GRAVITY LOS=MW(50), PURPOSE=10, FACTORS=FF ; HOSIE
GRAVITY LOS=MW(50), PURPOSE=11, FACTORS=FF ; LDRIE
GRAVITY LOS=MW(50), PURPOSE=12, FACTORS=FF ; HDRIE
GRAVITY LOS=MW(50), PURPOSE=13, FACTORS=FF ; HBO
GRAVITY LOS=MW(50), PURPOSE=14, FACTORS=FF ; HDORMU

FREQUENCY BASEMW=50, ValueMw= 1, RANGE=0-60-5.0, TITLE='HBW TLFD'
FREQUENCY BASEMW=50, ValueMw= 2, RANGE=0-60-5.0, TITLE='HBSH TLFD'
FREQUENCY BASEMW=50, ValueMw= 3, RANGE=0-60-5.0, TITLE='HBSR TLFD'
FREQUENCY BASEMW=50, ValueMw= 4, RANGE=0-60-5.0, TITLE='HBO TLFD'
FREQUENCY BASEMW=50, ValueMw= 5, RANGE=0-60-5.0, TITLE='NHB TLFD'
FREQUENCY BASEMW=50, ValueMw= 6, RANGE=0-60-5.0, TITLE='TK4 TLFD'
FREQUENCY BASEMW=50, ValueMw= 7, RANGE=0-60-5.0, TITLE='SGLUNIT TLFD'
FREQUENCY BASEMW=50, ValueMw= 8, RANGE=0-60-5.0, TITLE='TRKTLR TLFD'
FREQUENCY BASEMW=50, ValueMw= 9, RANGE=0-60-5.0, TITLE='SOVIE TLFD'
FREQUENCY BASEMW=50, ValueMw=10, RANGE=0-60-5.0, TITLE='MOVIE TLFD'
FREQUENCY BASEMW=50, ValueMw=11, RANGE=0-60-5.0, TITLE='LDTKIE TLFD'
FREQUENCY BASEMW=50, ValueMw=12, RANGE=0-60-5.0, TITLE='HDTKIE TLFD'
FREQUENCY BASEMW=50, ValueMw=13, RANGE=0-60-5.0, TITLE='HBU TLFD'
FREQUENCY BASEMW=50, ValueMw=14, RANGE=0-60-5.0, TITLE='HDORMU TLFD'

ENDRUN
FILEI MA[1] = "(SCENARIO_DIR)\output\EETAB.MAT"
FILEI MA[2] = "(SCENARIO_DIR)\output\EETAB.MAT"
FILEM MA[1] = "(SCENARIO_DIR)\output\PTRIPS.MAT"
MO=1, NAME=PRELOADVH

; The MATRIX module does not have any explicit phases. The module does not run within an implied ILOOP
; where I is the origin zones. All user statements in the module are processed once for each origin.
; Matrix computation (MW[#]) are solved for all values of J for each I. Thus for a given origin zone I
; the values for all destination zones J are automatically computed. The user can control the computations
; at each J by using a JLOOP.
PAR ZONEMSG=100
MW[1]=(MI.1.1+MI.1.1.T)*0.5*{AOFAC1}+
(MI.1.2+MI.1.2.T)*0.5*{AOFAC2}+
(MI.1.3+MI.1.3.T)*0.5*{AOFAC3}+
(MI.1.4+MI.1.4.T)*0.5*{AOFAC4}+
(MI.1.5+MI.1.5.T)*0.5*{AOFAC1}+
(MI.1.6+MI.1.6.T)*0.5+
(MI.1.7+MI.1.7.T)*0.5+
(MI.1.8+MI.1.8.T)*0.5+
(MI.1.9+MI.1.9.T)*0.5+
(MI.1.10+MI.1.10.T)*0.5+
(MI.1.11+MI.1.11.T)*0.5+
(MI.1.12+MI.1.12.T)*0.5+
(MI.1.13+MI.1.13.T)*0.5*{AOFACU} ; HBU
; (MI.1.14+MI.1.14.T)*0.5 ; HDORMU - don't include here because these are mostly not auto.

ENDRUN
PROCESS PHASE=LINKREAD
; USE THE USER SUPPLIED ALPHA AND BETA FOR THE BPR CURVE
IF (LI.BPRCOEFFICIENT=0)
  LW.BPRCOEFFICIENT=0.15
ELSE
  LW.BPRCOEFFICIENT=LI.BPRCOEFFICIENT
ENDIF
IF (LI.BPREXONENT=0)
  LW.BPREXONENT=4.0
ELSE
  LW.BPREXONENT=LI.BPREXONENT
ENDIF
IF (LI.CAPACITY=0)
  LW.DAILYCAP=999999
ELSE
  LW.DAILYCAP=(LI.CAPACITY/li.confac)*li.uroadfactor
ENDIF
IF (LI.TIME=0)
  LW.FTIME=0.0001
ELSE
  LW.FTIME=LI.TIME
ENDIF
C=LW.DAILYCAP
T0=LW.FTIME
IF (LI.FTYPE=49) ADDTOGROUP=1
ENDPROCESS

PROCESS PHASE=ILOOP
MW[1]=LI.1.PRELOADVEH
  PATHLOAD PATH=TIME, VOL[1]=MW[1], EXCLUDEGROUP=1, PENI=1
ENDPROCESS

PROCESS PHASE=ADJUST
  FUNCTION TC[1]=T0*(1+LW.BPRCOEFFICIENT*(V/C)^LW.BPREXONENT); congested time equation, no toll model in place
ENDPROCESS

ENDRUN
TERM[4]=TERM[40]
TERM[5]=TERM[50]

PROCESS PHASE=LINKREAD
IF (A1=(ZONESA)) TERM[TIME(A)]=TERM[LI.ATYPE1] ; BUILDS TERMINAL TIME ARRAY (KDK fixed again)
ENDPROCESS

PROCESS PHASE=LOOP
PATHLOAD PATH=LI.TIME,
MW[1]=PATHTRACE(LI.TIME_1,1),NOACCESS=99999,
MW[2]=PATHTRACE(LI.DISTANCE),NOACCESS=99999, PENI=1
MW[1][I]=LOWEST(1,2)/4 ; INTRAZONAL TIME = 1/2 THE AVERAGE OF THE TWO NEAREST ZONES
MW[2][I]=LOWEST(2,2)/4 ; INTRAZONAL DISTANCE = 1/2 THE AVERAGE OF THE TWO NEAREST ZONES
ENDPROCESS

PROCESS PHASE=ADJUST
ENDPROCESS
ENDRUN

DTMAT000DS
; Do not change filenames or add or remove FILEI/FILE0 statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\output\DISTRIB2.FRM" MSG='Second Distribution Report'
FILEI MATI[2] = "{SCENARIO_DIR}\output\PTRIPS.MAT"
FILEI MATI[1] = "{SCENARIO_DIR}\output\FSKIMS.(ALT){YEAR}.MAT"

MW[50]=MI.1.TIME+MI.1.TERMINALTIME
MW[1] = MW[50]*MI.2.HBW
MW[2] = MW[50]*MI.2.HBSH
MW[3] = MW[50]*MI.2.HBSR
MW[4] = MW[50]*MI.2.HBO
MW[5] = MW[50]*MI.2.NHB
MW[6] = MW[50]*MI.2.TRUCK4
MW[7] = MW[50]*MI.2.TRUCKSU
MW[8] = MW[50]*MI.2.TRUCKTRLR
MW[9] = MW[50]*MI.2.SOvie
MW[10]= MW[50]*MI.2.HVIE
MW[12]= MW[50]*MI.2.TRUCKHDIE
MW[13]= MW[50]*MI.2.HBU
MW[14]= MW[50]*MI.2.HDORMU
ENDRUN
TNNET00C.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM-NETWORK PRRFILE="{SCENARIO_DIR}\output\TNNET00A.PRN" MSG='Build Generate Statement for PNR Lots'
FILEI LOOKUP[1] - "{SCENARIO_DIR}\output\NODECOORD.csv"
FILEI LINK[2] - "{SCENARIO_DIR}\output\PRELOAD.NET"
FILEI LINK[1] - "{SCENARIO_DIR}\output\UNLOADED.NET"
FILEO PRINTO[4] - "{SCENARIO_DIR}\output\OPPNRCOST.CSV"
FILEO PRINTO[3] - "{SCENARIO_DIR}\output\PKPNRCOST.CSV"
FILEO PRINTO[2] - "{SCENARIO_DIR}\output\MD_STATDATA.CSV"
FILEO PRINTO[1] - "{SCENARIO_DIR}\output\AM_STATDATA.CSV"
ARRAY STATSTOP=99999 STATNUMB=99999, statspaces=99999, PNRTERM=99999, KNRTERM=99999, nrz=99999

; add in nearest centroid lookup for auto cost to stations HNYOPCOST
PROCESS PHASE-NODEMERGE
; put nodes, x and y coordinates into memory for lookup nearest TAZ question
lookup lookupi-1,name-netcoord, lookupi-1, result-2, lookupi(2)-1, result-3, fail-0
; extract am station info from network for later calculations
IF (AMUSEFLAG=1)
  workstat-N
  workstatx-netcoord(1,workstat,0)
  workstaty-netcoord(2,workstat,0)
  mindist=999.99
  loop _ww-1,ZONESA
    zx=netcoord(1,_ww,0)
    zy=netcoord(2,_ww,0)
    if (_ww!=workstat) dist=sqrt((workstatx-zx)^2+(workstaty-zy)^2)/{units}
  endif
  if (dist<mindist) mindist=dist, nearestzone=_ww
endloop

PRINT form=5.0,list="GENERATE,COST=(li.distance),MINCOST=12*1.0,MAXCOST=12*",pnrsvcarea(5.2L),
",EXTRACTCOST=(li.TIME_1),LIST=F,DIRECTION=1,NTELEMODE=2,FROMNODE=1-
{ZONESA},TONODE="{N},PRINTO=1
PRINT CSV=T, LIST=N(6.0),AMPNRCOST,NEARESTZONE(6.0) PRINTO=3
endif

; extract md station info from network for later calculations
IF (MDUSEFLAG=1)
  workstat-N
  workstatx-netcoord(1,workstat,0)
  workstaty-netcoord(2,workstat,0)
  mindist=999.99
  loop _ww-1,ZONESA
    zx=netcoord(1,_ww,0)
    zy=netcoord(2,_ww,0)
    if (_ww!=workstat) dist=sqrt((workstatx-zx)^2+(workstaty-zy)^2)/{units}
endloop

PRINT form=5.0,list="GENERATE,COST=(li.distance),MINCOST=12*1.0,MAXCOST=12**",pnrvarea(5.2L),
"",EXTRACTCOST=(li.TIME),LIST=T,DIRECTION=1,NTLEGMODE=2,FROMNODE=1-
{ZONESA},TONODE="N",PRINTO=2
PRINT CSV=T, LIST=N(6.0),MDPNRCOST,NEARESTZONE(6.0) PRINTO=4
endif

ENDPROCESS
ENDRUN

TNPTR00G.S
; Script for program PUBLIC TRANSPORT in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNPTR00G.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM-PUBLIC TRANSPORT PRFILE="(SCENARIO_DIR)\output\TNPTR000.DPRN" MSG='AM Walk Access'
FILEI NETI = "(SCENARIO_DIR)\output\PRELOAD.NET"
FILEI LINESI1 = "(SCENARIO_DIR)\input\trout020\(YEAR).lin"
FILEI TURNPENI = "(SCENARIO_DIR)\input\TCARDS.PEN"
FILEI Lookup[1] = "(CATALOG_DIR)\PARAMETERS\SPDCRV.CSV"
FILEO MATO[2] = "(SCENARIO_DIR)\output\WALKPREM.MAT",
MO=1-11, NAME=TIME1M,TIME2M,TIME3M,TIME4M,TIME5M,IWAIT,IXWT,IVTT,OVTT,FARE
FILEO MATO[1] = "(SCENARIO_DIR)\output\WALRAM.MAT",
MO=1-11, NAME=TIME1M,TIME2M,TIME3M,TIME4M,TIME5M,IWAIT,IXWT,IVTT,OVTT,FARE
FILEO REPORTO = "(SCENARIO_DIR)\output\TNPTR000.DPRN"
FILEI FACTORI2 = "(CATALOG_DIR)\Parameters\ALACHUAWKPREM.FAC"
FILEI FACTORI1 = "(CATALOG_DIR)\PARAMETERS\ALACHUAWLB.FAC"
FILEI FAREI = "(CATALOG_DIR)\Parameters\ALACHUA.FAR"
FILEO ROUTEO[2] = "(SCENARIO_DIR)\output\WALKPREM.RTE",
REPORT1=1-{zonesa}, REPORTJ=(cbdzone)
FILEO ROUTEO[1] = "(SCENARIO_DIR)\output\WALKLAM.RTE",
REPORT1=1-{zonesa}, REPORTJ=(cbdzone)
FILEI SYSTEMI = "(CATALOG_DIR)\PARAMETERS\ALACHUA.PTS"
FILEO NETO = "(SCENARIO_DIR)\output\TNETWALKN.NET"
PARAMETERS TRANTIME=(1.0,TIME 1*1.5), \GENERIC TRANSIT VS. AUTO RUN TIME
TRANTIME[4]=LW.LBTIME, \NODE SPECIFIC TRANSIT VS. AUTO RUN TIME
TRANTIME[6]=LW.EBTIME, \CREATED THROUGH A LOOKUP FUNCTION
TRANTIME[8]=LW.EBTIME,
FARE=F, USERCLASSES=1-2, HDWAYPERIOD=1 MAPSCALE=(UNITS)

REPORT LINES=T
;PROCESS PHASE=NODEREAD
; loops over all nodes computes node based scalar and array variables (Optional)
; ENDPROCESS
;
PROCESS PHASE=LINKREAD
LW.WALKTIME=60*LI.DISTANCE/walkspeed
IF (LI.TIME_1>0) LW.SPEED=60*LI.DISTANCE/LI.TIME_1
INTERPOLATE=Y, LOOKUPI=1
IF (LI.FTYPE=10-19,80-99) ; FREE FLOW CONDITIONS FOR ALL
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=10-19) ; BUSSIES HITTING RESISTANCE, RAIL NO CONFLICTS (GRADE SEP)
LW.LBCURVE=1,LW.EBCURVE=1,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=20-29) ; BUSSIES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=3,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=30-39) ; BUSSIES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=40-49) ; BUSSIES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=50-59) ; BUSSIES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=1,LW.RLCURVE=1
ENDIF
LW.LBSPEED=CURVES(LW.LBCURVE,LW.SPEED)
LW.EBSPEED=CURVES(LW.EBCURVE,LW.SPEED)
LW.RLSPEED=CURVES(LW.RLCURVE,LW.SPEED)
LW.LBTIME=60*LI.DISTANCE/LW.LBSPEED
LW.EBTIME=60*LI.DISTANCE/LW.EBSPEED
LW.RLTIME=60*LI.DISTANCE/LW.RLSPEED
TRANTIME[4]=LW.LBTIME
TRANTIME[6]=LW.EBTIME
TRANTIME[8]=LW.RLTIME
ENDPROCESS

PROCESS PHASE=DATAPREP
; WALK ACCESS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*24.0,LIST=T,NTLEGMODE = 1,
DIRECTION=1, FROMNODE=1-{zonesa}, TONODE=1000-99999
; WALK EGRESS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*24.0,LIST=T,NTLEGMODE=101,
DIRECTION=2, FROMNODE=1-{zonesa}, TONODE=1000-99999
; WALK CONNECTORS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*12,LIST=T,NTLEGMEMODE = 3,DIRECTION=3,
FROMNODE=1000-99999, TONODE=1000-99999

ENDPROCESS

PROCESS PHASE=SKIMIJ
MN[1]=TIMEA(0,1,101)
MN[2]=TIMEA(0,2,102)
MN[3]=TIMEA(0,3)
MN[4]=TIMEA(0,4)
MN[5]=TIMEA(0,6)
MN[6]=TIMEA(0,8)
MN[7]=IWAITA(0)
MN[8]=XWAITA(0)
MN[9]=TIMEA(0,100)
MN[10]=TIMEA(0,102)
MN[11]=FAREA(0,ALLMODES)

; VARIOUS THINGS THAT CAN BE SKIMMED
/*
 COMPCOST(RouteSet) Skims Composite Costs
 ValOfChoice(RouteSet) Skims Value of Choice
 IWAITA(RouteSet) Skims Initial Wait Times Actual
 XWAITA(RouteSet) Skims Transfer Wait Times Actual
 IWAIT(RouteSet) Skims Initial Wait Times Perceived
 XWAIT(RouteSet) Skims Initial Transfer Times Perceived
 TIMEA(RouteSet, Mode) Skims Travel Time Actual
 TIMEP(RouteSet, Mode) Skims Travel Time Perceived
*/
XFPENP (RouteSet, Mode) Skims Transfer Penalty Actual
XFPENP (RouteSet, Mode) Skims Transfer Penalty Actual
BRDINGS (RouteSet, Mode) Skims Number of Boardings (xfers+1)
BESTJRNY Skims Best Journey Times
FAREA (RouteSet, Mode) Skims Fares in Monetary units
FAREP (RouteSet, Mode) Skims Fares in Generalized Time units

ENDPROCESS

ENDRUN

TNPTR00H.S

; Script for program PUBLIC TRANSPORT in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNPTR00D.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRNFILE="(SCENARIO_DIR)\output\TNPTR00G.PRN" MSG="AM Auto Access"
FILEI NETI = "(SCENARIO_DIR)\output\PRELOAD.NET"
FILEI LOOKUPI[1] = "(CATALOG_DIR)\PARAMETERS\SPDCRV.CSV"
FILEI TURNJENI = "(SCENARIO_DIR)\input\TCARDS.PEN"
FILEO REPORTO = "(SCENARIO_DIR)\output\TNPTR00F.PRN"
FILEO MATO[1] = "(SCENARIO_DIR)\output\AUTOAM.MAT",
NAME=TIME1M,TIME2M,TIME3M,TIME4M,TIME6M,TIME8M,IWAIT,XWAIT,IVTT,OVTT,FARE
FILEO FACTOR[1] = "(CATALOG_DIR)\PARAMETERS\ALACHUAPNR.FAC"
FILEO FAREI = "(CATALOG_DIR)\parameters\ALACHUA.FAR"
FILEO ROUTEJ[1] = "(SCENARIO_DIR)\output\AUTOALLAM.RTE",
REPORT1-{zonesa}, REPORTJ-{cbdzone}
FILEO SYSTEM = "(CATALOG_DIR)\PARAMETERS\ALACHUA.PTS"
FILEO NETO = "(SCENARIO_DIR)\output\TNMETAUTOAM.NET"
FILEO LINEN[1] = "(SCENARIO_DIR)\input\route20\YEAR).lin"
PARAMETERS TRANTIME=(LI.TIME1*1.5), ; GENERIC TRANSIT VS. AUTO RUN TIME
TRANTIME[4]=LI.LBTIME, ; MODE SPECIFIC TRANSIT VS. AUTO RUN TIME
TRANTIME[6]=LI.EBTIME, ; CREATED THROUGH A LOOKUP FUNCTION
TRANTIME[8]=LI.RLTIME,
FARE=F, USERCLASSES=1, MAPSCALE=(UNITS)

HDWAYPERIOD=1

REPORT LINES=T

;PROCESS PHASE=NODEREAD
; loops over all nodes computes node based scalar and array variables (Optional)

;ENDPROCESS

; PROCESS PHASE=LINKREAD

Page | 90
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

LW.WALKTIME=60*LI.DISTANCE/{walkspeed}
IF (LI.TIME_1>0) LW.SPEED=60*LI.DISTANCE/LI.TIME_1
INTERPOLATE=Y, LOOKUPI=1
IF (LI.FTYPE=10-19,80-99) ; FREE FLOW CONDITIONS FOR ALL
LW.LBCURVE=1,LW.EBCURVE=1,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=10-19) ; BUSSES HITTING RESISTANCE, RAIL NO CONFLICTS (GRADE SEP)
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=20-29) ; BUSSES HITTING RESISTANCE (LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=3,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=30-39) ; BUSSES HITTING RESISTANCE (LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=3,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=40-49) ; BUSSES HITTING RESISTANCE (LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=50-59) ; BUSSES HITTING RESISTANCE (LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ENDIF
LW.LBSPEED=CURVES(LW.LBCURVE,LW.SPEED)
LW.EBSPEED=CURVES(LW.EBCURVE,LW.SPEED)
LW.RLSPEED=CURVES(LW.RLCURVE,LW.SPEED)
LW.LBTIME=60*LI.DISTANCE/LW.LBSPEED
LW.EBTIME=60*LI.DISTANCE/LW.EBSPEED
LW.RLTIME=60*LI.DISTANCE/LW.RLSPEED
TRANTIME[4]=LW.LBTIME
TRANTIME[6]=LW.EBTIME
TRANTIME[8]=LW.RLTIME

ENDPROCESS

PROCESS PHASE=DATAPREP
; AUTO ACCESS
READ,
FILE = "\{SCENARIO_DIR\Output\AM_STATDATA.CSV"
; WALK EGRESS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*24.0,LIST=T,NTLEGMODE=101,
DIRECTION=2, FROMNODE=1-{zonesa}, TONODE=1000-99999
; WALK CONNECTORS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*12,LIST=T,NTLEGMODE = 3,DIRECTION=3,
FROMNODE=1000-99999, TONODE=1000-99999

ENDPROCESS

;PROCESS PHASE=MATI
; manipulates input and work matrices prior to processing each Origin zone, I (Optional)

;ENDPROCESS

;PROCESS PHASE=SELECTIJ
; allows finer selection of zone pairs, IJ, for Route Evaluation, and the setting
; or revising of trips for Loading (Optional)

;ENDPROCESS

PROCESS PHASE=SKIMIJ
MW[1]=TIMEA(0,1,101)
MW[2]=TIMEA(0,2,102)
MW[3]=TIMEA(0,3)
MW[4]=TIMEA(0,4)
MW[5]=TIMEA(0,6)
MW[6]=TIMEA(0,8)
MW[7]=IWAITA(0)
MW[8]=XWAITA(0)
MW[9]=TIMEA(0,NTMODES)
MW[10]=TIMEA(0,NTMODES)
MW[11]=FAREA(0,ALLMODES)
VARIOUS THINGS THAT CAN BE SKIMMED

*/

COMP Costs (RouteSet) Skims Composite Costs
ValChoice(RouteSet) Skims Value of Choice
IWAITA(RouteSet) Skims Initial Wait Times Actual
XWAITA(RouteSet) Skims Transfer Wait Times Actual
IWAITP(RouteSet) Skims Initial Wait Times Perceived
XWAITP(RouteSet) Skims Initial Transfer Times Perceived
TIMEA(RouteSet, Mode) Skims Travel Time Actual
TIMEP(RouteSet, Mode) Skims Travel Time Perceived
XFERPENA(RouteSet, Mode) Skims Transfer Penalty Actual
XFERPENP(RouteSet, Mode) Skims Transfer Penalty Actual
DIST(RouteSet, Mode) Skims Distance

ENDPROCESS

;PROCESS PHASE=MATO
; allows processing of work matrices prior to them being written to the MATO files
; at the end of each Origin zone (Optional)

;ENDPROCESS

ENDRUN

TNPTR001.S

; Script for program PUBLIC TRANSPORT in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNPTR00E.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN FILE=PUBLIC TRANSPORT PRNFILE="{(SCENARIO_DIR)\output\TNPTR001.PRN" MSG='Create AM Stop to Stop File'
FILEI REPORTO = "{(SCENARIO_DIR)\output\TNPTR00J.PRN"
FILEO STOP2STOPO = "{(SCENARIO_DIR)\output\AMPNR.DBF",
ACCUMULATE="FIRSTLAST, NODES=1-99999
FILEI NETI = "{(SCENARIO_DIR)\output\TNETAUTOAM.NET"
FILEI ROUTE[1] = "{(SCENARIO_DIR)\output\AUTOALLAM.RTE"
PARAMETERS HDWAYPERIOD=1,
TRIPSIJ[1]=100,
NOROUTEERRS=999999999

ENDRUN
TNMAT00G.S
; Script for program MATRIX in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNMAT00E.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\output\TNMAT00E.FRN" MSG='Lookup AM Multi-path Parking Cost'
FILEO RECO[1] - "{SCENARIO_DIR}\output\AMPNR.DBF",
  FIELDS-ORZ,DSZ,MA,MEANCOST,MB,STNZONE,MC,CNT
FILEI RECI - "{SCENARIO_DIR}\output\AMPNR.DBF"
FILEI LOOKUPI[1] - "{SCENARIO_DIR}\output\PKPNRCOST.CSV"
  RO.ORZ=RI.I
  RO.DSZ=RI.J
  statnode=ri.fromnode
  RO.MA=1
  RO.MB=2
  RO.MC=3
  RO.CNT=1
  PCOST=STATIONS(1,statnode)
  RO.STNZONE=STATIONS(2,STATNODE)
  meancost=PCOST*ri.vol/100
WRITE RECO=1
ENDRUN

TNMAT00H.S
; Script for program MATRIX in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNMAT00C.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\output\TNMAT00C.FRN" MSG='Build AM PNR Cost Matrix'
FILEI MATI[2] - "{SCENARIO_DIR}\output\RHSKIMS.MAT"
FILEI MATI[1] - "{SCENARIO_DIR}\output\AMPNR.DBF"
FILEO MATO[1] - "{SCENARIO_DIR}\output\PKPNRCOST.MAT",
  MO=1-4,13,14 NAME=PKPNRCOST,STNZONE,STNTIME,STNDIST,FREQUENCY,TERMTIME
PAR ZONEMSG=100 ZONES={ZONESA}
MW[1]=MI.1.1 ; PNR COST
MW[2]=MI.1.2 ; STNZONE
MW[13]=MI.1.3 ; FREQUENCY COUNT
MW[14]=MI.2.TERMINALTIME
MW[10]=MI.2.TIME
jloop
  IF (MW[13]>0)
    STNZONE=MW[2]
    TIME=MW[10]
    DISTANCE=MW[11]
  ENDIF
endjloop
ENDRUN

TNMAT00LS
; Script for program MATRIX in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNMAT00A.S"
; Do not change filenames or add or remove FILEI/FILOE statements using an editor. Use
Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="|SCENARIO_DIR|\output\TNMAT00A.PRN" MSG='Compile AM LOS for Mode Choice'
FILEI MATT[4] = "|SCENARIO_DIR|\output\PKPFCOST.MAT"
FILEO MATO[1] = "|SCENARIO_DIR|\output\PKPKCOST.MAT",
MO=1-5,11-15,21-25,
NAME=PKWTIMEMLB,PKWTIMEMLB,PKIVTIMEMLB,PKPKCOSTLB,PKPFCOSTLB,
PKWTIMEMEX,PKWTIMEEMEX,PKIVTIMEEMEX,PKPKCOSTEX,PKPFCOSTEX,
PKWKTIMEMLB,PKWTIMEMLB,PKIVTIMEMLB,PKPKCOSTBA,PKPFCOSTBA dec=15*d
FILEI MATT[3] = "|SCENARIO_DIR|\output\AUTOAM.MAT"
FILEI MATT[2] = "|SCENARIO_DIR|\output\WALKFRENAM.MAT"
FILEI MATT[1] = "|SCENARIO_DIR|\output\WALKAM.MAT"
par zonemsg=100
jloop
  ; FIRST PROCESS THE WALK TO LOCAL BUS
  IF (MI.1.TIMEM=0&MI.1.TIMEMH=0)
    MW[001]=mi.1.ovtt+mi.4.termtime
    MW[002]=mi.1.iwait+mi.1.xwait
    MW[003]=mi.1.ltt
    MW[004]=0 ; no parking cost for walk modes
    MW[005]=mi.1.fare
  ELSE
    MW[013]=999999
  ENDIF
  ; NEXT PROCESS WALK TO EXPRESS SERVICE
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update

Technical Report No. 4: 2010 Model Update and Validation

IF (MI.1.TIME&6M>0|MI.1.TIME8M>0)
MW[011]=mi.2.ovtt+mi.4.termtime
MW[012]=mi.2.iwait+mi.2.xwait
MW[013]=mi.2.ivtt
MW[014]=0; no parking cost for walk modes
MW[015]=mi.1.fare
ENDIF

; NEXT PROCESS DRIVE TO BEST AVAILABLE
IF (mi.3.ivtt>0)
MW[021]=mi.3.ovtt+mi.4.termtime
MW[022]=mi.3.iwait+mi.3.xwait
MW[023]=mi.3.ivtt+mi.4.statime
MW[024]=mi.4.pknpcost+(mi.4.atndist*{hwyopcost})
MW[025]=mi.3.fare
ENDIF
MW[023]=999999
ENDLOOP
ENDRUN

TNPTR00JS.S

; Script for program PUBLIC TRANSPORT in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNPTR00A.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRNFILE="(SCENARIO_DIR)\output\TNPTR00B.PRN" MSG="MD Walk Access"
FILEI NETI = "(SCENARIO_DIR)\output\UNLOADED.NET"
FILEI LOOKUP{[1] = "(CATALOG_DIR)\Parameters\SPDCRV.CSV"
FILEI MATO{[2] = "(SCENARIO_DIR)\output\WALKPREMMD.MAT",
MODE 1-11, NAME=TIME1M,TIME2M,TIME3M,TIME4M,TIME6M,TIME8M,IWAIT,XWAIT,IVTT,OVT,FARE
FILEI MATO{[1] = "(SCENARIO_DIR)\output\WALKMD.MAT",
MODE 1-11, NAME=TIME1M,TIME2M,TIME3M,TIME4M,TIME6M,TIME8M,IWAIT,XWAIT,IVTT,OVT,FARE
FILEI FACTOR{[2] = "(SCENARIO_DIR)\Parameters\ALACHUAWKPREM.FAC"
FILEI FACTOR{[1] = "(SCENARIO_DIR)\Parameters\ALACHUAWLB.FAC"
FILEI FAREI = "(CATALOG_DIR)\Parameters\ALACHUA.FAR"
FILEI ROUTE{[2] = "(SCENARIO_DIR)\output\WALKPREMMD.RTE",
REPORTI=1-{zones}, REPORTJ={cbdzone}
FILEI ROUTE{[1] = "(SCENARIO_DIR)\output\WALKBLMD.RTE",
REPORTI=1-{zones}, REPORTJ={cbdzone}
FILEI SYSTEMI = "(CATALOG_DIR)\Parameters\ALACHUA.PTS"
FILEI REPORTO = "(SCENARIO_DIR)\output\TNPTR00A.PRN"
FILEI NETO = "(SCENARIO_DIR)\output\TNETWALKMD.NET"
FILEI LINE{[1] = "(SCENARIO_DIR)\input\route20\{YEAR}.lin"
FILEI TURN{[1] = "(SCENARIO_DIR)\input\TURNCARDS.PEN"
PARAMETERS TRANTIME{[1-5], GENERIC TRANSIT VS. AUTO RUN TIME
TRANTIME{[4]=LW.LBTIME, MODE SPECIFIC TRANSIT VS. AUTO RUN TIME
TRANTIME[6]=LW.EBTIME,  ; CREATED THROUGH A LOOKUP FUNCTION
TRANTIME[8]=LW.RLTIME,
FARE=F, USERCLASSES=1-2, HDWAYPERIOD=2 MAPSCALE={UNITS}

REPORT LINES=T

;PROCESS PHASE=NODEREAD
; loops over all nodes computes node based scalar and array variables (Optional)

;ENDPROCESS

;PROCESS PHASE=LINKREAD
LW.WALKTIME=60*LI.DISTANCE/{walkspeed}
IF (LI.TIME>0) LW.SPEED=60*LI.DISTANCE/LI.TIME
INTERPOLATE=Y, LOOKUPI=1
IF (LI.FTYPE=10-19,80-99) ; FREE FLOW CONDITIONS FOR ALL
LW.LBCURVE=1,LW.EBCURVE=1,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=10-19) ; BUSSES HITTING RESISTANCE, RAIL NO CONFLICTS (GRADE SEP)
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=20-29) ; BUSSES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=3,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=30-39) ; BUSSES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=3,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=40-49) ; BUSSES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=2,LW.RLCURVE=1
ELSEIF (LI.FTYPE=20-79&LI.ATYPE=50-59) ; BUSSES HITTING RESISTANCE(LB MORE), RAIL NO CONFLICTS
LW.LBCURVE=2,LW.EBCURVE=1,LW.RLCURVE=1
ENDIF
LW.LBSPEED=CURVES(LW.LBCURVE,LW.SPEED)
LW.EBSPEED=CURVES(LW.EBCURVE,LW.SPEED)
LW.RLSPEED=CURVES(LW.RLCURVE,LW.SPEED)
LW.LBTIME=60*LI.DISTANCE/LW.LBSPEED
LW.EBTIME=60*LI.DISTANCE/LW.EBSPEED
LW.RLTIME=60*LI.DISTANCE/LW.RLSPEED
TRANTIME[4]=LW.LBTIME
TRANTIME[6]=LW.EBTIME
TRANTIME[8]=LW.RLTIME

ENDPROCESS

PROCESS PHASE=DATAPREP
; WALK ACCESS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*24.0,LIST=T,NTLEGMODE=1,
DIRECTION=1, FROMNODE=1-{zonesa}, TONODE=1000-99999
; WALK ERCESS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*24.0,LIST=T,NTLEGMODE=101,
DIRECTION=2, FROMNODE=1-{zonesa}, TONODE=1000-99999
; WALK CONNECTORS
GENERATE, COST=(LW.WALKTIME),MAXCOST=103*12,LIST=T,NTLEGMODE=3,DIRECTION=3,
FROMNODE=(1000-99999, TONODE=1000-99999

ENDPROCESS

;PROCESS PHASE=MATI
; manipulates input and work matrices prior to processing each Origin zone, I (Optional)

;ENDPROCESS

;PROCESS PHASE=SELECTIJ
; allows finer selection of zone pairs, IJ, for Route Evaluation, and the setting
; or revising of trips for Loading (Optional)

;ENDPROCESS

PROCESS PHASE=SKIMIJ
MN[1]=TIMEA(0,1,101)
MO=1-11,
FILEO MA
FILEO RE
FILEI LO
FILEI NE
RUN PGM=
Cube/App;
Do not
Script
TNPTR00
ENDRUN

;VARIOUS THINGS THAT CAN BE SKIMMED
/*
COMPCOST(RouteSet) Skims Composite Costs
ValOfChoice(RouteSet) Skims Value of Choice
IWAITA(RouteSet) Skims Initial Wait Times Actual
XWAITA(RouteSet) Skims Transfer Wait Times Actual
IWAITP(RouteSet) Skims Initial Wait Times Perceived
XWAITP(RouteSet) Skims Initial Transfer Times Perceived
TIMEA(RouteSet, Mode) Skims Travel Time Actual
TIMEP(RouteSet, Mode) Skims Travel Time Perceived
XFERPEN(RouteSet, Mode) Skims Transfer Penalty Actual
XFERPEN(RouteSet, Mode) Skims Transfer Penalty Actual
DRT(RouteSet, Mode) Skims Distance
BRDINGS(RouteSet, Mode) Skims Number of Boardings (xFers+1)
BESTJRNY Skims Best Journey Times
FAREA(RouteSet, Mode) Skims Fares in Monetary units
FAREP(RouteSet, Mode) Skims Fares in Generalized Time units
*/
ENDPROCESS
;PROCESS PHASE=MATO
; allows processing of work matrices prior to them being written to the MATO files
; at the end of each Origin zone (Optional)
ENDPROCESS
ENDRUN

TNPTR00K.S
; Script for program PUBLIC TRANSPORT in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNPTR00B.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN FGM=PUBLIC TRANSPORT PRNFILE="\n(SCENARIO_DIR)\out\TNPTR00H.PRN" MSG="MD Auto Access"
FILEI NETI = "\(SCENARIO_DIR)\out\UNLOADED.NET"
FILEI LOOKUP[1] = "\(CATALOG_DIR)\Parameters\SPDCSV.CSV"
FILEO ReportO = "\(SCENARIO_DIR)\out\TNPTR00E.PRN"
FILEO MATO[1] = "\(SCENARIO_DIR)\out\AUTOMD.MAT",
MO=1-11, NAME=TIME1M,TIME2M,TIME3M,TIME4M,TIME6M,TIME8M,IWAIT,XWAIT,IVTT,OVTT,FARE
FILEI FACTORI[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"
FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEO ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"

FILEI FACTOR[1] = "{CATALOG_DIR}\Parameters\ALACHUAPNR.FAC"
FILEI FAREI = "{CATALOG_DIR}\parameters\ALACHUA.FAR"
FILEI ROUTE(1) = "{SCENARIO_DIR}\output\AUTOALLMD.RTE",
REPORT I-{zonesa}, REPORT J-{cbdzone}
FILEI SYSTEM = "{CATALOG_DIR}\Parameters\ALACHUA.FTS"
FILEI LINEI[1] = "{SCENARIO_DIR}\input\troute20{YEAR}.lin"
FILEI TURNPEN = "{SCENARIO_DIR}\input\TCARDS.PEN"
FILEO NETO = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"
;ENDPROCESS

;PROCESS PHASE=SELECTI
; allows finer selection of zone pairs, IJ, for Route Evaluation, and the setting
; or revising of trips for Loading (Optional)

;ENDPROCESS

PROCESS PHASE=SKIMIJ
MW[1]=TIMEA(0,1,101)
MW[2]=TIMEA(0,2,102)
MW[3]=TIMEA(0,3)
MW[4]=TIMEA(0,4)
MW[5]=TIMEA(0,6)
MW[6]=TIMEA(0,8)
MW[7]=IWAITA(0)
MW[8]=XWAITA(0)
MW[9]=TIMEA(0,TMODES)
MW[10]=TIMEA(0,NTMODES)
MW[11]=FAREA(0,ALLMODES)

;VARIOUS THINGS THAT CAN BE SKIMMED
/
COMPCOST(RouteSet) Skims Composite Costs
ValOfChoice(RouteSet) Skims Value of Choice
IWAITA(RouteSet) Skims Initial Wait Times Actual
XWAITA(RouteSet) Skims Transfer Wait Times Actual
IWAITP(RouteSet) Skims Initial Wait Times Perceived
XWAITP(RouteSet) Skims Travel Time Perceived
TIMEA(RouteSet, Mode) Skims Travel Time Actual
TIMEP(RouteSet, Mode) Skims Travel Time Perceived
XFERPE(RouteSet, Mode) Skims Transfer Penalty Actual
XFERPEN(RouteSet, Mode) Skims Transfer Penalty Perceived
DIST(RouteSet, Mode) Skims Distance
BRDINGS(RouteSet, Mode) Skims Number of Boardings (xfers+1)
BESTJRNY Skims Best Journey Times
FAREA(RouteSet, Mode) Skims Fares in Monetary units
FAREP(RouteSet, Mode) Skims Fares in Generalized Time units
*/

ENDPROCESS
;PROCESS PHASE=MATO
; allows processing of work matrices prior to them being written to the MATO files
; at the end of each Origin zone (Optional)

;ENDPROCESS

ENDRUN

TNPTR00L_S
; Script for program PUBLIC TRANSPORT in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNPTR00F.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRRFILE="{SCENARIO_DIR}\output\TNPTR00K.PRN" MSG=`Create MD Stop to Stop File`
FILEI ROUTEi[1] = "{SCENARIO_DIR}\output\AUTOALLMD.RTE"
FILEI NETI = "{SCENARIO_DIR}\output\TNETAUTO4D.NET"
FILEO REPORTO = "{SCENARIO_DIR}\output\TNPTR00L.PRN"
FILEO STOPSTOPO = "{SCENARIO_DIR}\output\TNPTR00C.DBF",
ACCUMUL=FIRSTLAST, NODES=1-99999
PARAMETERS HDWAYPERIOD=2,
TRIPSJ[1]=100,
NOROUTER=999999999
ENDRUN

**TNMAT00J.S**
;; Script for program MATRIX in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNMAT00D.S"
;; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRRFILE="{SCENARIO_DIR}\output\TNMAT00D.PRN" MSG=`Lookup MD Multi-path Parking Cost`
FILEI RECI = "{SCENARIO_DIR}\output\TNPTR00C.DBF"
FILEO RECO[1] = "{SCENARIO_DIR}\output\MDPCOST.DBF",
FIELD=ORZ,DSZ,MA,MEANCO,MB,STNZONE,MC,CNT
FILEI LOOKUPI[1] = "{SCENARIO_DIR}\output\OPPNRCOST.CSV"
FAIL[2]=0, LOOKUPI=1
RO.ORZ=RI.i
RO.DSZ=RI.J
statnode=ri.fromnode
RO.MA=1
RO.MB=2
RO.MC=3
RO.CNT=1
PCOST=STATIONS(1,statnode)
RO.STNZONE=STATIONS(2,STATNODE)
MEANCO=PCOST*ri.vol/100
WRITE RECO=1
ENDRUN

**TNMAT00K.S**
; Script for program MATRIX in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNMAT00F.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\output\TNMAT00F.PRN" MSG='Build MD FNR Cost Matrix'
FILEI MATI[2] = "{SCENARIO_DIR}\output\FHSKIMS.(ALT\{YEAR\}.MAT"
FILEI MATI[1] = "{SCENARIO_DIR}\output\MDPCOST.DBF",
PATTERN=IJ:MV, FILED=ORZ,DSZ,MA,MEANCOST,MB,STNZONE,MC,CNT
FILEO MATO[1] = "{SCENARIO_DIR}\output\OPNRNCOST.MAT",
MO=1-4,15,14 NAME=PNRNCOST,STNZONE,STNTIME,STNDIST,FREQUENCY,TERTIME

PAR ZONEMSG=100 ZONES={ZONESA}
MW(1)=MI.1.1 ; FNR COST
MW(2)=MI.1.2 ; STNZONE
MW(3)=MI.1.3 ; FREQUENCY COUNT
MW(4)=MI.2.TERMINALTIME
MW(10)=MI.2.TIME
MW(11)=MI.2.DISTANCE
jloop
    IF (MW(13)>0)
        MW(2)=MW(2)/MW(13)
STNZONE=0=0
TIME=MW(10)
DISTANCE=MW(11)
    MW(3)=TIME, MW(4)=DISTANCE
ENDIF
endjloop
PAR ZONES={ZONESA}
MW(1)=MI.1.1
ENDRUN

TNMAT00S
; Script for program MATRIX in file "C:\FSUTMS\DISTRICT2\ALACHUA\TNMAT00B.S"
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\output\TNMAT00B.PRN" MSG='Compile MD LOS for Mode Choice'
FILEI MATI[4] = "{SCENARIO_DIR}\output\OPPNRCOST.MAT"
FILEI MATI[3] = "{SCENARIO_DIR}\output\AUTOMD.MAT"
FILEI MATI[2] = "{SCENARIO_DIR}\output\WALKPREM.MAT"
FILEI MATI[1] = "{SCENARIO_DIR}\output\WALMD.MAT"
FILEO MATO[1] = "{SCENARIO_DIR}\output\OP TRN LOS.MAT",
MO=1-5,11-15,21-25,
NAME=OPKTIMELB,OPWTIMELB,OPIVTIMELB,OPFKCOSTLB,OPFCOSTLB,
OPKTIMEEX,OPWTIMEEX,OPIVTIMEEX,OPFKCOSTEX,OPFCOSTEX,
OPKTIMEBA,OPWTIMEBA,OPIVTIMEBA,OPFKCOSTBA,OPFCOSTBA DEC=15*D

par zonemsg=100
JLOOP
    IF (MI.1.TIME5M=0&MI.1.TIME8M=0)
        MW(001)=mi.1.otvit+mi.4.termtime
        MW(002)=mi.1.iwait+mi.1.xwait
        MW(003)=mi.1.ivtt
        MW(004)=0 ; no parking cost for walk modes
    ENDIF
endjloop
MW[005]=mi.1.fare
ELSE
MW[003]=999999
ENDIF

; NEXT PROCESS WALK TO EXPRESS SERVICE
IF (MI.1.TIME6M>0||MI.1.TIME8M>0)
MW[011]=mi.2.overt+mi.4.termtime
MW[012]=mi.2.iwait+mi.2.xwait
MW[013]=mi.2.ivtt
MW[014]=0 ; no parking cost for walk modes
MW[015]=mi.1.fare
ELSE
MW[013]=999999
ENDIF

; NEXT PROCESS DRIVE TO BEST AVAILABLE
IF (mi.3.ivtt>0)
MW[021]=mi.3.overt+mi.4.termtime
MW[022]=mi.3.iwait+mi.3.xwait
MW[023]=mi.3.ivtt+mi.4.stntime
MW[024]=mi.4.pkpcost+{mi.4.atndist*{hwyopcost}}
MW[025]=mi.3.fare
ELSE
MW[023]=999999
ENDIF
ENDLOOP
ENDRUN
Gainesville Urbanized Area 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

MCMAT00A.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="{SCENARIO_DIR}\output\MCMAT00A.PRN" MSG="Calculates Utilities & Runs Choice"
FILEI MATI[1] = "{SCENARIO_DIR}\output\FHSKIMS.(ALT\{YEAR\}.MAT"
FILEI ZDATI[3] = "{SCENARIO_DIR}\output\UFFPANDA.DBF"
FILEI MATI[5] = "{SCENARIO_DIR}\output\RHSKIMS.MAT"
FILEI MATI[4] = "{SCENARIO_DIR}\output\OP TRN LOS.MAT"
FILEI MATI[3] = "{SCENARIO_DIR}\output\PEAK TRN LOS.MAT"
FILEI MATI[2] = "{SCENARIO_DIR}\output\TRIPS.MAT"
FILEO PRINTO[4] = "{SCENARIO_DIR}\output\MODE SUM.CSV"
FILEI ZDATI[1] = "{SCENARIO_DIR}\input\ZoneData\{YEAR\}.DBF", Z=TAZ_20\{year\}
FILEI LOOKUPI[2] = "{SCENARIO_DIR}\output\MCLOO00C.CSV"
FILEO PRINTO[3] = "{SCENARIO_DIR}\output\NEWK.CSV"
FILEO PRINTO[2] = "{SCENARIO_DIR}\output\REV_MODE_CONST.CSV"
FILEO PRINTO[1] = "{SCENARIO_DIR}\output\MODE SUMMARY.PRN"
    name=NBDA,HBWCP,HBWCK,HBWB,HBWK,HBWBK,
    HBODA,HBOCP,HBOCX,HBOWB,HBOWK,HBOWBk,HBOWR,
    NHBA,HBNCF,HBNCX,HBWB,HBWK,HBWBK,HBWBK,
    HBODA,HBOCP,HBOCX,HBOWB,HBOWK,HBOWBk,HBOWR,
    HDORMUB,HDORMUW,HDORMUBk, DEC=24*S
FILEI LOOKUPI[1] = "{CATALOG_DIR}\parameters\MC_COEFFICIENTS.CSV"
par zonemsg=100

; THE JOB OF THIS SCRIPT IS TO TURN THE COMPONENTS OF UTILITY FOR EACH MODE IN THE MODE CHOICE
; INTO A COMPOSITE UTILITY.  BECAUSE THE MODEL IS NESTED, WITH NESTING COEFFICIENTS APPLIED
; IN THE MODE CHOICE MODE, THE INPUT UTILITIES SHOULD BE DIVIDED BY THE PRODUCT OF THE NESTING
; COEFFICIENTS.

; MARKET SEGMENTS ARE:
; 0 CAR HOUSEHOLDS
; 1 OR MORE CAR HOUSEHOLDS
; UNIVERSITY STUDENTS

; TRIP PURPOSES ARE:
; 1 HBW (AM PEAK LOS MATRICES)
; 2 HBO (MD OFF-PEAK LOS)
; 3 NHB (MD OFF-PEAK LOS)
; 4 HBU (MD OFF-PEAK LOS)
; 5 HDORMU (choice set: walk, bike, MD walk-local BUS)
MW[1]=MI.2.HBW
MW[2]=MI.2.HBSH+MI.2.HBSR+MI.2.HBO
MW[3]=MI.2.NHB
MW[4]=MI.2.HBU
MW[5]=MI.2.HDORMU

; THE AUTO DIVISOR IS NESTCMOTOR*NESTCAUTO
NESTMOTOR=\{NESTCMOTOR\}*\{NESTCAUTO\}
; THE TRANSIT DIVISOR IS NESTCMOTOR*NESTCTRANSIT
NESTTRANSIT=\{NESTCMOTOR\}*(\{NESTCTRANSIT\}
NESTNONMOTOR=\{NESTNONMOTOR\}

;Coefficients
lookup, name=coefficients,
interpolate=n, LIST=Y, lookupi=1
; cvt-IN VEHICLE TIME COEFFICIENT
HBWCVT=COEFFICIENTS(1,1), HBOCVT=COEFFICIENTS(2,1), NHBCVT=COEFFICIENTS(3,1),
UNICVT=COEFFICIENTS(4,1)
; cvt-OUT OF VEHICLE TIME COEFFICIENT
HBWCVT=COEFFICIENTS(1,2), HBOCVT=COEFFICIENTS(2,2), NHBCVT=COEFFICIENTS(3,2),
UNICVT=COEFFICIENTS(4,2)
; ccst-COST COEFFICIENT (cents)
HBWCCST=COEFFICIENTS(1,3), HBOCCST=COEFFICIENTS(2,3), NHBCST=COEFFICIENTS(3,3),
UNICST=COEFFICIENTS(4,3)
; cwt-WALK ONLY COEFFICIENT

Page | 103
HBWCT=COEFFICIENTS(1,4), HBOCT=COEFFICIENTS(2,4), NHBCWT=COEFFICIENTS(3,4), UNICT=COEFFICIENTS(4,4),
; BIKE ONLY COEFFICIENT
HBWCBT=COEFFICIENTS(1,5), HBOCTBT=COEFFICIENTS(2,5), NHBCBT=COEFFICIENTS(3,5),
UNIPT=COEFFICIENTS(4,5);
p㎝-Walk PEV J
HBWPJ=COEFFICIENTS(1,7), HBOPT=COEFFICIENTS(2,7), NHBPJ=COEFFICIENTS(3,7),
UNIPT=COEFFICIENTS(4,7);
p㎝-Walk PEV J
HBWPJ=COEFFICIENTS(1,8), HBOPT=COEFFICIENTS(2,8), NHBPJ=COEFFICIENTS(3,8),
UNIPT=COEFFICIENTS(4,8);
p㎝-BIKE PEV J
HBWPJ=COEFFICIENTS(1,9), HBOPT=COEFFICIENTS(2,9), NHBPJ=COEFFICIENTS(3,9),
UNIPT=COEFFICIENTS(4,9);
p㎝-BIKE PEV J
HBWPJ=COEFFICIENTS(1,10), HBOPT=COEFFICIENTS(2,10), NHBPJ=COEFFICIENTS(3,10),
UNIPT=COEFFICIENTS(4,10);

; Constants
; K=COSTANT, FOLLOWED BY TRIP PURPOSE FOLLOWED BY MODE
; DRIVE ONLY
K1_NC_DA=CONSTANTS(1,1), K2_NC_DA=CONSTANTS(4,1), K3_NC_DA=CONSTANTS(7,1),
K4_NC_DA=CONSTANTS(8,1),
K1_WC_DA=CONSTANTS(2,1), K2_WC_DA=CONSTANTS(5,1),
K3_WC_DA=CONSTANTS(6,1),
K4_WC_DA=CONSTANTS(7,1),
K5_WC_DA=CONSTANTS(8,1),

; CARPOOL
K1_NC_CP=CONSTANTS(1,2), K2_NC_CP=CONSTANTS(4,2), K3_NC_CP=CONSTANTS(7,2),
K4_NC_CP=CONSTANTS(8,2),
K1_WC_CP=CONSTANTS(2,2), K2_WC_CP=CONSTANTS(5,2),
K3_WC_CP=CONSTANTS(6,2),
K4_WC_CP=CONSTANTS(7,2),
K5_WC_CP=CONSTANTS(8,2),

; WALK TO BUS
K1_NC_WB=CONSTANTS(1,4), K2_NC_WB=CONSTANTS(4,4), K3_NC_WB=CONSTANTS(7,4),
K4_NC_WB=CONSTANTS(8,4),
K1_WC_WB=CONSTANTS(2,4), K2_WC_WB=CONSTANTS(5,4),
K3_WC_WB=CONSTANTS(6,4),
K4_WC_WB=CONSTANTS(7,4),
K5_WC_WB=CONSTANTS(8,4),

; WALK TO PREMIUM TRANSIT
K1_NC_WX=CONSTANTS(1,5), K2_NC_WX=CONSTANTS(4,5), K3_NC_WX=CONSTANTS(7,5),
K4_NC_WX=CONSTANTS(8,5),
K1_WC_WX=CONSTANTS(2,5), K2_WC_WX=CONSTANTS(5,5),
K3_WC_WX=CONSTANTS(6,5),
K4_WC_WX=CONSTANTS(7,5),
K5_WC_WX=CONSTANTS(8,5),

; AUTO TO TRANSIT
K1_NC_BA=CONSTANTS(1,6), K2_NC_BA=CONSTANTS(4,6), K3_NC_BA=CONSTANTS(7,6),
K4_NC_BA=CONSTANTS(8,6),
K1_WC_BA=CONSTANTS(2,6), K2_WC_BA=CONSTANTS(5,6),
K3_WC_BA=CONSTANTS(6,6),
K4_WC_BA=CONSTANTS(7,6),
K5_WC_BA=CONSTANTS(8,6),

; WALK ONLY
K1_NC_WK=CONSTANTS(1,7), K2_NC_WK=CONSTANTS(4,7), K3_NC_WK=CONSTANTS(7,7),
K4_NC_WK=CONSTANTS(8,7),
K1_WC_WK=CONSTANTS(2,7), K2_WC_WK=CONSTANTS(5,7),
K3_WC_WK=CONSTANTS(6,7),
K4_WC_WK=CONSTANTS(7,7),
K5_WC_WK=CONSTANTS(8,7),

; Constants
; 3*HBW,HBO,NHB,Constants for 0 car, 1+car and student, rows=mode=da,cp,cx,wl,wx,ab,wk,bk
lookup, name=constants,
lookup[7]=1, result=8, lookup[8]=1, result=9, lookup[9]=1, result=10,
interpolate=n, LIST=Y, lookupi=2
K=CONSTANTS, FOLLOWED BY TRIP PURPOSE FOLLOWED BY MODE
; DRIVE ALONE
K1_NC_DA=CONSTANTS(1,1), K2_NC_DA=CONSTANTS(4,1), K3_NC_DA=CONSTANTS(7,1),
K4_NC_DA=CONSTANTS(8,1),
K1_WC_DA=CONSTANTS(2,1), K2_WC_DA=CONSTANTS(5,1),
K3_WC_DA=CONSTANTS(6,1),
K4_WC_DA=CONSTANTS(7,1),
K5_WC_DA=CONSTANTS(8,1),

; CARPOOL
K1_NC_CP=CONSTANTS(1,2), K2_NC_CP=CONSTANTS(4,2), K3_NC_CP=CONSTANTS(7,2),
K4_NC_CP=CONSTANTS(8,2),
K1_WC_CP=CONSTANTS(2,2), K2_WC_CP=CONSTANTS(5,2),
K3_WC_CP=CONSTANTS(6,2),
K4_WC_CP=CONSTANTS(7,2),
K5_WC_CP=CONSTANTS(8,2),
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

K1_ST_WK=CONSTANTS(3,7), K2_ST_WK=CONSTANTS(6,7)
; BIKE ONLY
K1_NC_BK=CONSTANTS(1,8), K2_NC_BK=CONSTANTS(4,8), K3_NC_BK=CONSTANTS(7,8),
K4_NC_BK=CONSTANTS(8,8)
K1_WC_BK=CONSTANTS(2,8), K2_WC_BK=CONSTANTS(5,8),
K3_WC_BK=CONSTANTS(9,8)
K1_ST_BK=CONSTANTS(3,8), K2_ST_BK=CONSTANTS(6,8)

;TARGETS
;*HBW, HBO, NBH, Targets for 0 car, 1+car and student, rows=mode=da,cp,cx,wx,ab,wk,bk
lookup, name=targ, lookup[1]=1, result=2, lookup[2]=1, result=3, lookup[3]=1, result=4,
lookup[7]=1, result=8, lookup[8]=1, result=9, lookup[9]=1, result=10,
interpolate=n, LIST=y, lookupi=3
;t=Target, FOLLOWED BY TRIP PURPOSE FOLLOWED BY MODE
; DRIVE ALONE
t1_NC_DA=targ(1,1), t2_NC_DA=targ(4,1), t3_NC_DA=targ(7,1), t4_NC_DA=targ(8,1)
t1_WC_DA=targ(2,1), t2_WC_DA=targ(5,1), t3_WC_DA=targ(6,1)                         , t5_NC_DA=targ(9,1)
t1_ST_DA=targ(3,1), t2_ST_DA=targ(6,1)  
; 2+ CARPOOL
t1_NC_CP=targ(1,2), t2_NC_CP=targ(4,2), t3_NC_CP=targ(7,2), t4_NC_CP=targ(8,2)
t1_WC_CP=targ(2,2), t2_WC_CP=targ(5,2), t3_WC_CP=targ(6,2)                    , t5_NC_CP=targ(9,2)
t1_ST_CP=targ(3,2), t2_ST_CP=targ(6,2)
; 3+ CARPOOL
t1_NC_CX=targ(1,3), t2_NC_CX=targ(4,3), t3_NC_CX=targ(7,3), t4_NC_CX=targ(8,3)
t1_WC_CX=targ(2,3), t2_WC_CX=targ(5,3), t3_WC_CX=targ(6,3)                    , t5_NC_CX=targ(9,3)
t1_ST_CX=targ(3,3), t2_ST_CX=targ(6,3)  
; WALK TO BUS
t1_NC_WB=targ(1,4), t2_NC_WB=targ(4,4), t3_NC_WB=targ(7,4), t4_NC_WB=targ(8,4)
t1_WC_WB=targ(2,4), t2_WC_WB=targ(5,4), t3_WC_WB=targ(6,4)                    , t5_NC_WB=targ(9,4)
t1_ST_WB=targ(3,4), t2_ST_WB=targ(6,4)  
; WALK TO PREMIUM TRANSIT
t1_NC_WX=targ(1,5), t2_NC_WX=targ(4,5), t3_NC_WX=targ(7,5), t4_NC_WX=targ(8,5)
t1_WC_WX=targ(2,5), t2_WC_WX=targ(5,5), t3_WC_WX=targ(6,5)                    , t5_NC_WX=targ(9,5)
t1_ST_WX=targ(3,5), t2_ST_WX=targ(6,5)
; AUTO TO TRANSIT
t1_NC_BA=targ(1,6), t2_NC_BA=targ(4,6), t3_NC_BA=targ(7,6), t4_NC_BA=targ(8,6)
t1_WC_BA=targ(2,6), t2_WC_BA=targ(5,6), t3_WC_BA=targ(6,6)                    , t5_NC_BA=targ(9,6)
t1_ST_BA=targ(3,6), t2_ST_BA=targ(6,6)
; WALK ONLY
t1_NC_WK=targ(1,7), t2_NC_WK=targ(4,7), t3_NC_WK=targ(7,7), t4_NC_WK=targ(8,7)
t1_WC_WK=targ(2,7), t2_WC_WK=targ(5,7), t3_WC_WK=targ(6,7)                    , t5_NC_WK=targ(9,7)
t1_ST_WK=targ(3,7), t2_ST_WK=targ(6,7)
; BIKE ONLY
t1_NC_BK=targ(1,8), t2_NC_BK=targ(4,8), t3_NC_BK=targ(7,8), t4_NC_BK=targ(8,8)
t1_WC_BK=targ(2,8), t2_WC_BK=targ(5,8), t3_WC_BK=targ(6,8)                    , t5_NC_BK=targ(9,8)
t1_ST_BK=targ(3,8), t2_ST_BK=targ(6,8)

; COST UNITS
; assume parking costs are in cents, both for auto and PnR lots
; assume fares are in dollars, so multiply by 100.
; assume auto operating costs are in dollars, so multiply by 100.
;
; Bus fare factor
; difference of bus fare factors between year 2007 (1.0) and
; future scenarios (1.5) will be reduced to 10% of actual difference amount
; due to significant impact from this bus fare increase since year 2007 ($1.00 to $1.50).
busfarefac=1+({BUSFAREFAC}-1)*0.10)

JLOOP
;=================================================================================
; HBW (PEAK) TRIP PURPOSE
;=================================================================================

; PEAK PERIOD DRIVE ALONE ELEMENTS OF UTILITY ARE:
; WALK TIME
; MW[11]=(MI.5.TERMINALTIME)*HBWCOVT
; WAIT TIME
; MW[12]=(0)*HBWCOVT
; IVTT
MW[13] = (MI.5.TIME) * HBWCIVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION
MW[14] = 0.5 * ZI.1.LONGPARK(J) * HBWCST
; OTHER COST
MW[15] = MI.5.DISTANCE * (HWYOPCOST) * 100 * HBWCST
; COMPOSITE UTILITY

; PEAK PERIOD CARPOOL2 ELEMENTS OF UTILITY ARE:
; WALK TIME
; WAIT TIME
MW[12] = (0) * HBWCOT
; IVTT
MW[13] = (MI.5.TIME) * HBWCIVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION, SHARED BY 2 = 0.25
MW[14] = 0.25 * ZI.1.LONGPARK(J) * HBWCST
; OTHER COST
MW[15] = 0.25 * ZI.1.LONGPARK(J) * HBWCST
; COMPOSITE UTILITY

; PEAK PERIOD CARPOOL3 ALONE ELEMENTS OF UTILITY ARE:
; WALK TIME
; WAIT TIME
MW[12] = (0) * HBWCOT
; IVTT
MW[13] = (MI.5.TIME) * HBWCIVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION, SHARED BY {hbw3p}
MW[14] = 0.5 * ZI.1.LONGPARK(J) * HBWCST/{hbw3p}
; OTHER COST
MW[15] = MI.5.DISTANCE * (HWYOPCOST) * 100 * HBWCST/{hbw3p}
; COMPOSITE UTILITY

; PEAK PERIOD WALK TO LOCAL BUS ELEMENTS OF UTILITY ARE:
; WALK TIME
MW[11] = (mi.3.pkwtimelb) * HBWCOT
; WAIT TIME
MW[12] = (mi.3.pkwtimelb) * HBWCOT
; IVTT
MW[13] = (mi.3.pkivtimelb) * HBWCIVT
; PARKING COST
MW[14] = (mi.3.pkpkcostlb) * HBWCST
; OTHER COST = t
MW[15] = (mi.3.pkopcostlb * 100 * 0.25 * busfarefac) * HBWCST
; CS applied 25% (discounted) bus fare due to employee pass program
; PEDESTRIAN ENVIRONMENT
MW[16] = HBWPTI * ZI.1.SUM[1] * 0.25
; COMPOSITE UTILITY

; PEAK PERIOD WALK TO PREMIUM TRANSIT ELEMENTS OF UTILITY ARE:
; WALK TIME
MW[11] = (mi.3.pkwtimex) * HBWCOT
; WAIT TIME
MW[12] = (mi.3.pkwtimex) * HBWCOT
; IVTT
MW[13] = (mi.3.pkwtimeba) * HBWCOVT

; WALK TIME
MW[12] = (mi.3.pkwtimeba) * HBWCOVT

; IVTT
MW[13] = (mi.3.pkwtimeba) * HBWCOVT

; PARKING COST
MW[14] = (mi.3.pckcostba) * HBWCCST

; OTHER COST - FARE
MW[15] = (mi.3.pkopcostba * 100) * HBWCCST

; PEDESTRIAN ENVIRONMENT
MW[16] = HBWPTI * ZI.1.SUM[1] * 0.25

; COMPOSITE UTILITY

; PEAK PERIOD AUTO TO BEST AVAILABLE TRANSIT ELEMENTS OF UTILITY ARE:
; WALK TIME
MW[11] = (mi.3.pkwtimeex) * HBWCI VT

; IVTT
MW[13] = (mi.3.pkwtimeex) * HBWCI VT

; PARKING COST
MW[14] = (mi.3.pckcostex) * HBWCCST

; OTHER COST - FARE
MW[15] = (mi.3.pkopcostex * 100) * HBWCCST

; PEDESTRIAN ENVIRONMENT
MW[16] = HBWPTI * ZI.1.SUM[1] * 0.25

; COMPOSITE UTILITY

; OFF-PEAK PERIOD DRIVE ALONE ELEMENTS OF UTILITY ARE:
; HBO
; WALK TIME

; IVTT
MW[13] = (MI.1.TIME) * HBOCIVT

; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION
MW[14] = 0.5 * (ZI.1.SHORTPARK[1] + ZI.1.SHORTPARK[2]) * HBOCCST

; OTHER COST
MW[15] = MI.1.DISTANCE * {HWYOPCOST} * 100 * HBOCCST

; COMPOSITE UTILITY

; NHB
; WALK TIME

; IVTT
MW[13] = (MI.1.TIME) * NHBCIVT

; PARKING COST - AVG ORIGIN AND DESTINATION
MW[14] = 0.5 * (ZI.1.SHORTPARK[1] + ZI.1.SHORTPARK[2]) * NHBCCST

; OTHER COST
MW[15] = MI.1.DISTANCE * {HWYOPCOST} * 100 * NHBCCST

; COMPOSITE UTILITY

; HBU
; WALK TIME

; IVTT
MW[13] = (MI.1.TIME) * UNICIVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION
MW[14]=0.5 * ZI.1.STUDENTPAR[J] * UNICCST ; university gets long term cost
; OTHER COST
MW[15]=MI.1.DISTANCE * (HWYOPCOST) * 100 * UNICCST
; COMPOSITE UTILITY

; OFF-PEAK PERIOD CARPOOL2 ELEMENTS OF UTILITY ARE:
-- HBO --
WALK TIME
WAIT TIME
;MW[12]=0 * HBOCOVT
IVTT
MW[13]=(MI.1.TIME) * HBOCVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION
MW[14]=0.25 * ZI.1.SHORTPARK[J] * HBOCCST
; OTHER COST
MW[15]=0.5 * MI.1.DISTANCE * (HWYOPCOST) * 100 * HBOCCST
; COMPOSITE UTILITY

-- NHB --
WALK TIME
WAIT TIME
;MW[12]=0 * NHBCOVT
IVTT
MW[13]=(MI.1.TIME) * NHBCVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION
MW[14]=0.25 * (ZI.1.SHORTPARK[I]+ZI.1.SHORTPARK[J]) * NHBCST
; OTHER COST
MW[15]=0.5 * MI.1.DISTANCE * (HWYOPCOST) * 100 * NHBCST
; COMPOSITE UTILITY

-- HBU --
WALK TIME
WAIT TIME
;MW[12]=0 * UNICOVT
IVTT
MW[13]=(MI.1.TIME) * UNICIVT
; PARKING COST - ONLY AT DESTINATION (J), HALF IN EACH DIRECTION
MW[14]=0.25 * ZI.1.STUDENTPAR[J] * UNICCST ; university gets long term cost
; OTHER COST
MW[15]=0.5 * MI.1.DISTANCE * (HWYOPCOST) * 100 * UNICCST
; COMPOSITE UTILITY

; OFF-PEAK PERIOD CARPOOL3 ELEMENTS OF UTILITY ARE:
-- HBO --
WALK TIME
WAIT TIME
;MW[12]=0 * HBOCOVT
IVTT
MW[13]=(MI.1.TIME) * HBOCVT
; PARKING COST
MW[14]=0.50 * ZI.1.SHORTPARK[J] * HBOCCST/(hbo3p)
; OTHER COST
MW[15]=(MI.1.DISTANCE* (HWYOPCOST) * 100)* HBOCCST/(hbo3p)
; COMPOSITE UTILITY
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

-- NHB --
WALK TIME
WAIT TIME
MW[12] = (0)*NHBCOVT
IVTT
MW[13] = (MI.1.TIME)*NHBCIVT
PARKING COST
MW[14] = 0.50 * (ZI.1.SHORTPARK[1]+ZI.1.SHORTPARK[1]) * NHBCST/NHB3P
OTHER COST
MW[15] = (MI.1.DISTANCE*{HWYOPCOST}*100)*NHBCST/NHB3P
COMPOSITE UTILITY

-- HBU --
WALK TIME
WAIT TIME
MW[12] = (0)*UNICOVT
IVTT
MW[13] = (MI.1.TIME)*UNICIVT
PARKING COST
MW[14] = 0.50 * ZI.1.STUDENTPAR[J] * UNICCST/hbw3p; assume 3+ occ like work & Long park
cost
OTHER COST
MW[15] = (MI.1.DISTANCE*{HWYOPCOST}*100)*UNICCST/hbw3p; assume 3+ occ like work
COMPOSITE UTILITY

OFF-PEAK PERIOD WALK TO LOCAL BUS ELEMENTS OF UTILITY ARE:
-- HBO --
WALK TIME
MW[11] = (mi.4.opwktimelb)*HBOCOVT
WAIT TIME
MW[12] = (mi.4.opwttimelb)*HBOCOVT
IVTT
MW[13] = (mi.4.opivtimelb)*HBOCIVT
PARKING COST
MW[14] = (mi.4.oppkcostlb)*HBOCCST
OTHER COST - FARE
MW[15] = (mi.4.opopcostlb*100*busfarefac)*HBOCCST
PEDESTRIAN ENVIRONMENT
MW[16] = HBOPTI * ZI.1.SUM[I]*0.25
COMPOSITE UTILITY

-- NHB --
WALK TIME
MW[11] = (mi.4.opwktimelb)*NHBCOVT
WAIT TIME
MW[12] = (mi.4.opwttimelb)*NHBCOVT
IVTT
MW[13] = (mi.4.opivtimelb)*NHBCIVT
PARKING COST
MW[14] = (mi.4.oppkcostlb)*NHBCCST
OTHER COST - FARE
MW[15] = (mi.4.opopcostlb*100*busfarefac)*NHBCCST
PEDESTRIAN ENVIRONMENT
MW[16] = NHBPTI * ZI.1.SUM[I]*0.25
COMPOSITE UTILITY

-- HBU --
WALK TIME
MW[11] = (mi.4.opwktimelb)*UNICOVT
WAIT TIME
MW[12] = (mi.4.opwttimelb)*UNICOVT
; IVTT
MW[13]=(mi.4.opivtimelb)*UNICIVT
; PARKING COST
MW[14]=(mi.4.oppkcostlb)*UNICCST
; OTHER COST - FARE
MW[15]=(mi.4.opopcostlb * 100 * 0.10*busfarefac)*UNICCST
; CS applied 10% (discounted) bus fare
; due to transit fare payed in tuition

; MW[15]=0 ; UF fare free - previous model
; PEDESTRIAN ENVIRONMENT
MW[16]=UNIPTI * ZI.1.SUM[I]*0.25
; COMPOSITE UTILITY

; -- HDORMI --
; WALK TIME
MW[11]=(mi.4.opwktimelb)*UNICOVT
; WAIT TIME
MW[12]=(mi.4.opwttimelb)*UNICOVT
; IVTT
MW[13]=(mi.4.opivtimelb)*UNICIVT
; PARKING COST
MW[14]=(mi.4.oppkcostlb)*UNICCST
; OTHER COST - FARE
MW[15]=(mi.4.opopcostlb * 100 * 0.10*busfarefac)*UNICCST
; CS applied 10% (discounted) bus fare
; due to transit fare payed in tuition

; MW[15]=0 ; UF fare free - previous model
; PEDESTRIAN ENVIRONMENT
MW[16]=UNIPTI * ZI.1.SUM[I]*0.25
; COMPOSITE UTILITY

; OFF-PEAK PERIOD WALK TO PREMIUM TRANSIT ELEMENTS OF UTILITY ARE:

; -- HBO --
; WALK TIME
MW[11]=(mi.4.opwktimelb)*HBOCOV T
; WAIT TIME
MW[12]=(mi.4.opwttimelb)*HBOCOV T
; IVTT
MW[13]=(mi.4.opivtimelb)*HBOCIVT
; PARKING COST
MW[14]=(mi.4.oppkcostex)*HBCCST
; OTHER COST
MW[15]=(mi.4.opopcostex * 100)*HBCCST
; PEDESTRIAN ENVIRONMENT
MW[16]=HBPTI * ZI.1.SUM[I]*0.25
; COMPOSITE UTILITY

; -- NHB --
; WALK TIME
MW[11]=(mi.4.opwktimelb)*NHBCOV T
; WAIT TIME
MW[12]=(mi.4.opwttimelb)*NHBCOV T
; IVTT
MW[13]=(mi.4.opivtimelb)*NHBCIVT
; PARKING COST
MW[14]=(mi.4.oppkcostex)*NHCCST
; OTHER COST
MW[15]=(mi.4.opopcostex * 100)*NHCCST
; PEDESTRIAN ENVIRONMENT
MW[16]=NHPTI * ZI.1.SUM[I]*0.25
; COMPOSITE UTILITY
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation


; -- HBU --
; WALK TIME
MW[11]= (mi.4.opwktimeex)*UNICOVT
; WAIT TIME
MW[12]= (mi.4.opwttimeex)*UNICOVT
; IVTT
MW[13]= (mi.4.opivtimeex)*UNICIVT
; PARKING COST
MW[14]= (mi.4.oppkcostex)*UNICCST
; OTHER COST
;MW[15]= (mi.4.opopcostex)*UNICCST
MW[15]= 0 ; UF fare free
; PEDESTRIAN ENVIRONMENT
MW[16]= UNIPTI * ZI.1.SUM[I]*0.25
; COMPOSITE UTILITY

; OFF-PEAK PERIOD AUTO TO BEST AVAILABLE TRANSIT ELEMENTS OF UTILITY ARE:
; -- HBO --
; WALK TIME
MW[11]= (mi.4.opwktimeba)*HBOCOVT
; WAIT TIME
MW[12]= (mi.4.opwttimeba)*HBOCOVT
; IVTT
MW[13]= (mi.4.opivtimeba)*HBOCIVT
; PARKING COST
MW[14]= (mi.4.oppkcostba)*HBOCCST
; OTHER COST
MW[15]= (mi.4.opopcostba * 100)*HBOCCST
; COMPOSITE UTILITY

; -- NHB --
; WALK TIME
MW[11]= (mi.4.opwktimeba)*NHBCOVT
; WAIT TIME
MW[12]= (mi.4.opwttimeba)*NHBCOVT
; IVTT
MW[13]= (mi.4.opivtimeba)*NHBCIVT
; PARKING COST
MW[14]= (mi.4.oppkcostba)*NHBCCST
; OTHER COST
MW[15]= (mi.4.opopcostba * 100)*NHBCCST
; COMPOSITE UTILITY

; -- HBU --
; WALK TIME
MW[11]= (mi.4.opwktimeba)*UNICOVT
; WAIT TIME
MW[12]= (mi.4.opwttimeba)*UNICOVT
; IVTT
MW[13]= (mi.4.opivtimeba)*UNICIVT
; PARKING COST
MW[14]= (mi.4.oppkcostba)*UNICCST
; OTHER COST
;MW[15]= (mi.4.opopcostba)*UNICCST
MW[15]= 0 ; UF fare free
; COMPOSITE UTILITY
; WALK ONLY (NON-MOTORIZED) ELEMENTS OF UTILITY ARE:
; WALK AND BIKE TIMES
mw[8]=60*M.I.1.WALDDISTANCE/(WALKSPEED) ; all walk
mw[9]=M.I.1.BIKETIME ; all bike

; HBW
; WALK TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*HBWPWI + ZI.1.SUM[J]*HBWPWJ)
; 0.25 because we are using sum, not composite
; UTILITIES

; HBO
; WALK TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*HBOPWI + ZI.1.SUM[J]*HBOPWJ)
; UTILITIES

; NHB
; WALK TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*NHBPWI + ZI.1.SUM[J]*NHBPWJ)
; UTILITIES

; UNIVERSITY
; WALK TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*UNIPWI + ZI.1.SUM[J]*UNIPWJ)
; UTILITIES

; BIKE ONLY (NON-MOTORIZED) ELEMENTS OF UTILITY ARE:

; HBW
; BIKE TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*HBWPBI + ZI.1.SUM[J]*HBWPBJ)
; UTILITIES

; HBO
; BIKE TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*HBOPBI + ZI.1.SUM[J]*HBOPBJ)
; UTILITIES

; NHB
; BIKE TIME
; PEDESTRIAN ENVIRONMENT
MW[12]=0.25*(ZI.1.SUM[I]*NHBPBI + ZI.1.SUM[J]*NHBPBJ)
; UTILITIES

; UNIVERSITY
; BIKE TIME
; PEDESTRIAN ENVIRONMENT
\[ \text{MW}[12] = 0.25 \times (ZI.1 \times \text{SUM}[I] \times \text{UNIPBI} + ZI.1 \times \text{SUM}[J] \times \text{UNIPBJ}) \]

; UTILITIES
\[ \text{MW}[091] = \left( \text{MW}[11] + MW[12] + K4_{NC \_BK} \right) / \{ \text{NESTCNONMOTOR} \} \]
\[ \text{MW}[092] = \left( \text{MW}[11] + MW[12] + K5_{NC \_BK} \right) / \{ \text{NESTCNONMOTOR} \} \]
\[ \text{endjloop} \]

; MARKET SEGMENTATION: car, no car student
\[ \text{MW}[301] = \text{MW}[1] \times ZI.3 \times \text{NOCARPCT} ; 0 \text{ car} \]
\[ \text{MW}[302] = \text{MW}[1] \times ZI.3 \times \text{WCARPCT} ; \text{ with car} \]
\[ \text{MW}[303] = \text{MW}[1] \times ZI.3 \times \text{STUPCT} ; \text{ student} \]
\[ \text{MW}[304] = \text{MW}[2] \times ZI.3 \times \text{NOCARPCT} ; 0 \text{ car} \]
\[ \text{MW}[305] = \text{MW}[2] \times ZI.3 \times \text{WCARPCT} ; \text{ with car} \]
\[ \text{MW}[306] = \text{MW}[2] \times ZI.3 \times \text{STUPCT} ; \text{ student} \]

; HBW (USE 0 CAR)
; DA, BA not in market
CHOICE ALTERNATIVES=CP,CX,WB,WX,WK,BK,
DEMAND=\text{MW}[301],
UTILITIES=\text{MW}[022],\text{MW}[023],\text{MW}[024],\text{MW}[025],\text{MW}[027],\text{MW}[028],
ODEMAND=402,403,404,405,407,408,
STARTMW=500,
SPLIT=TOTAL, \{ \text{NESTCNONMOTOR} \} \text{ NONMOTOR}, \{ \text{NESTCMOTOR} \} \text{ MOTOR},
SPLIT=MOTOR, \{ \text{NESTCAUTO} \} \text{ AUTO}, \{ \text{NESTCTRANSIT} \} \text{ TRANSIT},
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=AUTO, 1.0 DA, 1.0 CP, 1.0 CX,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX

\[ \text{MW}[401]=0; \text{ no drive alone} \]
\[ \text{MW}[406]=0; \text{ no auto access} \]

; HBW (USE 1+ CAR)
CHOOSE ALTERNATIVES=DA,CP,CX,WB,WX,WK,BK,
DEMAND=\text{MW}[302],
UTILITIES=\text{MW}[031],\text{MW}[032],\text{MW}[033],\text{MW}[034],\text{MW}[035],\text{MW}[036],\text{MW}[037],\text{MW}[038],
ODEMAND=411,412,413,414,415,416,417,418,
STARTMW=500,
SPLIT=TOTAL, \{ \text{NESTCNONMOTOR} \} \text{ NONMOTOR}, \{ \text{NESTCMOTOR} \} \text{ MOTOR},
SPLIT=MOTOR, \{ \text{NESTCAUTO} \} \text{ AUTO}, \{ \text{NESTCTRANSIT} \} \text{ TRANSIT},
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=AUTO, 1.0 DA, 1.0 CP, 1.0 CX,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX, 1.0 BA

; HBW (USE STUDENT)
CHOICE ALTERNATIVES=DA,CP,CX,WB,WX,WK,BK,
DEMAND=\text{MW}[303],
UTILITIES=\text{MW}[041],\text{MW}[042],\text{MW}[043],\text{MW}[044],\text{MW}[045],\text{MW}[046],\text{MW}[047],\text{MW}[048],
ODEMAND=421,422,423,424,425,426,427,428,
STARTMW=500,
SPLIT=TOTAL, \{ \text{NESTCNONMOTOR} \} \text{ NONMOTOR}, \{ \text{NESTCMOTOR} \} \text{ MOTOR},
SPLIT=MOTOR, \{ \text{NESTCAUTO} \} \text{ AUTO}, \{ \text{NESTCTRANSIT} \} \text{ TRANSIT},
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=AUTO, 1.0 DA, 1.0 CP, 1.0 CX,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX, 1.0 BA

\[ \text{MW}[151] = \text{MW}[401] + \text{MW}[407] + \text{MW}[413] + \text{MW}[421] \]
\[ \text{MW}[152] = \text{MW}[402] + \text{MW}[412] + \text{MW}[422] \]
\[ \text{MW}[153] = \text{MW}[403] + \text{MW}[413] + \text{MW}[423] \]
\[ \text{MW}[154] = \text{MW}[404] + \text{MW}[414] + \text{MW}[424] \]
\[ \text{MW}[155] = \text{MW}[405] + \text{MW}[415] + \text{MW}[425] \]
\[ \text{MW}[156] = \text{MW}[406] + \text{MW}[416] + \text{MW}[426] \]
\[ \text{MW}[157] = \text{MW}[407] + \text{MW}[417] + \text{MW}[427] \]
\[ \text{MW}[158] = \text{MW}[408] + \text{MW}[418] + \text{MW}[428] \]

; HBO (USE 0 CAR)
; DA, BA not in market
CHOICE ALTERNATIVES=CP,CX,WB,WK,BK,
DEMAND=\text{MW}[304],
UTILITIES=\text{MW}[052],\text{MW}[053],\text{MW}[054],\text{MW}[055],\text{MW}[057],\text{MW}[058],
ODEMAND=432,433,434,435,437,438,
STARTMW=500,
SPLIT=TOTAL, (NESTCNONMOTOR) NONMOTOR, (NESTCMOTOR) MOTOR,
SPLIT=MOTOR, (NESTCAUTO) AUTO, (NESTCTRANSIT) TRANSIT,
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX

MW[431]=0; no drive alone
MW[436]=0; no auto access

; HBO (USE 1+ CAR)
CHOICE ALTERNATIVES=DA,CP,CX,WB,WX,BA,WK,BK,
DEMAND=MW[305],
UTILITIES=MW[061],MW[062],MW[063],MW[064],MW[065],MW[066],MW[067],MW[068],
ODEMAND=441,442,443,444,445,446,447,448,
STARTMW=500,
SPLIT=TOTAL, (NESTCNONMOTOR) NONMOTOR, (NESTCMOTOR) MOTOR,
SPLIT=MOTOR, (NESTCAUTO) AUTO, (NESTCTRANSIT) TRANSIT,
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX, 1.0 BA

; HBO (USE STUDENT)
CHOICE ALTERNATIVES=DA,CP,CX,WB,WX,BA,WK,BK,
DEMAND=MW[306],
UTILITIES=MW[071],MW[072],MW[073],MW[074],MW[075],MW[076],MW[077],MW[078],
ODEMAND=451,452,453,454,455,456,457,458,
STARTMW=500,
SPLIT=TOTAL, (NESTCNONMOTOR) NONMOTOR, (NESTCMOTOR) MOTOR,
SPLIT=MOTOR, (NESTCAUTO) AUTO, (NESTCTRANSIT) TRANSIT,
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX, 1.0 BA

; NHB (USE 0 CAR CONSTANTS, NO MARKET SEGMENTATION IS NEEDED FOR THIS TRIP PURPOSE)
CHOICE ALTERNATIVES=DA,CP,CX,WB,WX,BA,WK,BK,
DEMAND=MW[003],
UTILITIES=MW[081],MW[082],MW[083],MW[084],MW[085],MW[086],MW[087],MW[088],
ODEMAND=171,172,173,174,175,176,177,178,
STARTMW=500,
SPLIT=TOTAL, (NESTCNONMOTOR) NONMOTOR, (NESTCMOTOR) MOTOR,
SPLIT=MOTOR, (NESTCAUTO) AUTO, (NESTCTRANSIT) TRANSIT,
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX, 1.0 BA

; HBU (FULL CHOICE SET)
CHOICE ALTERNATIVES=DA,CP,CX,WB,WX,BA,WK,BK,
DEMAND=MW[004],
UTILITIES=MW[101],MW[102],MW[103],MW[104],MW[105],MW[106],MW[089],MW[091],
ODEMAND=181,182,183,184,185,186,187,188,
STARTMW=500,
SPLIT=TOTAL, (NESTCNONMOTOR) NONMOTOR, (NESTCMOTOR) MOTOR,
SPLIT=MOTOR, (NESTCAUTO) AUTO, (NESTCTRANSIT) TRANSIT,
SPLIT=NONMOTOR, 1.0 WK, 1.0 BK,
SPLIT=TRANSIT, 1.0 WB, 1.0 WX, 1.0 BA

; CAMPUS HOUSING -HDORMU- (PARTIAL CHOICE SET)
CHOICE ALTERNATIVES=WB,WX,BK,
DEMAND=MW[5],
UTILITIES=MW[110],MW[090],MW[092],
DEMAND=191,192,193,
STARTMW=600,
SPLIT=TOTAL, 1.0 WB, 1.0 WK, 1.0 BK

JLOOP
;
HBWDA=HBWDA+MW[151]
HBWCP=HBWCP+MW[152]
HBWCX=HBWCX+MW[153]
HBWNB=HBWNB+MW[154]
HBWNX=HBWNX+MW[155]
HBWBA=HBWBA+MW[156]
HBWKR=HBWKR+MW[157]
HBWKB=HBWKB+MW[158]

HBODA=HBODA+MW[161]
HBOCP=HBOCP+MW[162]
HBOCX=HBOCX+MW[163]
HBOWB=HBOWB+MW[164]
HBOWNX=HBOWNX+MW[165]
HBOB=HBOB+A+MW[166]
HBORK=HBORK+MW[167]
HBOBK=HBOBK+MW[168]

NHBD=NHBD+MW[171]
NHBCP=NHBCP+MW[172]
NHBCX=NHBCX+MW[173]
NHBB=NHBB+MW[174]
NHBNX=NHBNX+MW[175]
NHBB=NHBB+MW[176]
NHBNK=NHBNK+MW[177]
NHBBK=NHBBK+MW[178]

HBUDA=HBUDA+MW[181]
HBUCP=HBUCP+MW[182]
HBUCX=HBUCX+MW[183]
HBUNB=HBUNB+MW[184]
HBUNX=HBUNX+MW[185]
HBUBA=HBUBA+MW[186]
HBUNK=HBUNK+MW[187]
HBUBK=HBUBK+MW[188]

UNIWB=UNIWB+MW[191]
UNINK=UNINK+MW[192]
UNIBK=UNIBK+MW[193]

MW401=MW401+MW[401]
MW411=MW411+MW[411]
MW421=MW421+MW[421]
MW431=MW431+MW[431]
MW441=MW441+MW[441]
MW451=MW451+MW[451]
MW171=MW171+MW[171]
MW181=MW181+MW[181]
MW402=MW402+MW[402]
MW412=MW412+MW[412]
MW422=MW422+MW[422]
MW432=MW432+MW[432]
MW442=MW442+MW[442]
MW452=MW452+MW[452]
MW172=MW172+MW[172]
MW182=MW182+MW[182]
MW403=MW403+MW[403]
MW413=MW413+MW[413]
MW423=MW423+MW[423]
MW433=MW433+MW[433]
MW443=MW443+MW[443]
MW453=MW453+MW[453]
MW173=MW173+MW[173]
MW183=MW183+MW[183]
if (i=_zones)

SUMHBW=HBWDA+HBWCP+HBWCX+HBWWB+HBWWX+HBWBA+HBWBK+HBWBK
SUMHBO=HBODA+HBOCX+HBOWB+HBOWX+HBOBA+HBOWK+HBOBK
SUMNHB=NHBDA+NHBCP+NHBGX+NHBWB+NHBWX+NHBBA+NHBWK+NHBWK
SUMNBO=NHBDA+NHBCP+NHBGX+NHBWB+NHBWX+NHBBA+NHBWK+NHBWK
SUMUNI=UNIWB+UNIWK+UNIBK

;Total trips by purpose
mx1=MW401+MW402+MW403+MW404+MW405+MW406+MW407+MW408
mx2=MW411+MW412+MW413+MW414+MW415+MW416+MW417+MW418
mx3=MW421+MW422+MW423+MW424+MW425+MW426+MW427+MW428
mx4=MW431+MW432+MW433+MW434+MW435+MW436+MW437+MW438
mx5=MW441+MW442+MW443+MW444+MW445+MW446+MW447+MW448
mx6=MW451+MW452+MW453+MW454+MW455+MW456+MW457+MW458
mx7=MW171+MW172+MW173+MW174+MW175+MW176+MW177+MW178
mx8=MW181+MW182+MW183+MW184+MW185+MW186+MW187+MW188
mx9=MW191+MW192+MW193

;mode shares
MW401=MW401/mx1
MW402=MW402/mx1
MW403=MW403/mx1
MW404=MW404/mx1
MW405=MW405/mx1
<table>
<thead>
<tr>
<th>Mode Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MW406</td>
<td></td>
</tr>
<tr>
<td>MW407</td>
<td></td>
</tr>
<tr>
<td>MW408</td>
<td></td>
</tr>
<tr>
<td>MW409</td>
<td></td>
</tr>
<tr>
<td>MW411</td>
<td></td>
</tr>
<tr>
<td>MW412</td>
<td></td>
</tr>
<tr>
<td>MW413</td>
<td></td>
</tr>
<tr>
<td>MW414</td>
<td></td>
</tr>
<tr>
<td>MW415</td>
<td></td>
</tr>
<tr>
<td>MW416</td>
<td></td>
</tr>
<tr>
<td>MW417</td>
<td></td>
</tr>
<tr>
<td>MW418</td>
<td></td>
</tr>
<tr>
<td>MW421</td>
<td></td>
</tr>
<tr>
<td>MW422</td>
<td></td>
</tr>
<tr>
<td>MW423</td>
<td></td>
</tr>
<tr>
<td>MW424</td>
<td></td>
</tr>
<tr>
<td>MW425</td>
<td></td>
</tr>
<tr>
<td>MW426</td>
<td></td>
</tr>
<tr>
<td>MW427</td>
<td></td>
</tr>
<tr>
<td>MW428</td>
<td></td>
</tr>
<tr>
<td>MW431</td>
<td></td>
</tr>
<tr>
<td>MW432</td>
<td></td>
</tr>
<tr>
<td>MW433</td>
<td></td>
</tr>
<tr>
<td>MW434</td>
<td></td>
</tr>
<tr>
<td>MW435</td>
<td></td>
</tr>
<tr>
<td>MW436</td>
<td></td>
</tr>
<tr>
<td>MW437</td>
<td></td>
</tr>
<tr>
<td>MW438</td>
<td></td>
</tr>
<tr>
<td>MW441</td>
<td></td>
</tr>
<tr>
<td>MW442</td>
<td></td>
</tr>
<tr>
<td>MW443</td>
<td></td>
</tr>
<tr>
<td>MW444</td>
<td></td>
</tr>
<tr>
<td>MW445</td>
<td></td>
</tr>
<tr>
<td>MW446</td>
<td></td>
</tr>
<tr>
<td>MW447</td>
<td></td>
</tr>
<tr>
<td>MW448</td>
<td></td>
</tr>
<tr>
<td>MW451</td>
<td></td>
</tr>
<tr>
<td>MW452</td>
<td></td>
</tr>
<tr>
<td>MW453</td>
<td></td>
</tr>
<tr>
<td>MW454</td>
<td></td>
</tr>
<tr>
<td>MW455</td>
<td></td>
</tr>
<tr>
<td>MW456</td>
<td></td>
</tr>
<tr>
<td>MW457</td>
<td></td>
</tr>
<tr>
<td>MW458</td>
<td></td>
</tr>
<tr>
<td>MW171</td>
<td></td>
</tr>
<tr>
<td>MW172</td>
<td></td>
</tr>
<tr>
<td>MW173</td>
<td></td>
</tr>
<tr>
<td>MW174</td>
<td></td>
</tr>
<tr>
<td>MW175</td>
<td></td>
</tr>
<tr>
<td>MW176</td>
<td></td>
</tr>
<tr>
<td>MW177</td>
<td></td>
</tr>
<tr>
<td>MW178</td>
<td></td>
</tr>
<tr>
<td>MW181</td>
<td></td>
</tr>
<tr>
<td>MW182</td>
<td></td>
</tr>
<tr>
<td>MW183</td>
<td></td>
</tr>
<tr>
<td>MW184</td>
<td></td>
</tr>
<tr>
<td>MW185</td>
<td></td>
</tr>
<tr>
<td>MW186</td>
<td></td>
</tr>
<tr>
<td>MW187</td>
<td></td>
</tr>
<tr>
<td>MW188</td>
<td></td>
</tr>
<tr>
<td>MW189</td>
<td></td>
</tr>
<tr>
<td>MW190</td>
<td></td>
</tr>
<tr>
<td>MW191</td>
<td></td>
</tr>
<tr>
<td>MW192</td>
<td></td>
</tr>
<tr>
<td>MW193</td>
<td></td>
</tr>
<tr>
<td>MW194</td>
<td></td>
</tr>
</tbody>
</table>

PRINT LIST="\n+++++ MODE CHOICE SUMMARY ++++
PRINTO=1 PRINT LIST="\{(DESC)\" PRINTO=1
### Mode Choice Results

#### Home-Based Work Mode Choice Results

<table>
<thead>
<tr>
<th>Mode Choice</th>
<th>Total</th>
<th>Drive Alone</th>
<th>Carpool 2</th>
<th>Carpool 3+</th>
<th>Walk to Local Transit</th>
<th>Walk to Premium Transit</th>
<th>Drive to Best Available Transit</th>
<th>Non-Motorized Walk</th>
<th>Non-Motorized Bicycle</th>
<th>Average Auto Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBW TOTAL</td>
<td>SUMHBW</td>
<td>HBWDA</td>
<td>HBWCP</td>
<td>HBWCX</td>
<td>HBWBA</td>
<td>HBWWB</td>
<td>HBWBA</td>
<td>HBWWK</td>
<td>HBWBK</td>
<td>(HBWDA+HBWCP+HBWCX)/(HBWDA+HBWCP+HBWCX)</td>
</tr>
<tr>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA</td>
<td>HBWDA/(HBWDA+HBWCP+HBWCX)</td>
</tr>
<tr>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP</td>
<td>HBWCP/(HBWDA+HBWCP+HBWCX)</td>
</tr>
<tr>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX</td>
<td>HBWCX/(HBWDA+HBWCP+HBWCX)</td>
</tr>
</tbody>
</table>

#### Home-Based Other Mode Choice Results

<table>
<thead>
<tr>
<th>Mode Choice</th>
<th>Total</th>
<th>Drive Alone</th>
<th>Carpool 2</th>
<th>Carpool 3+</th>
<th>Walk to Local Transit</th>
<th>Walk to Premium Transit</th>
<th>Drive to Best Available Transit</th>
<th>Non-Motorized Walk</th>
<th>Non-Motorized Bicycle</th>
<th>Average Auto Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBO TOTAL</td>
<td>SUMHBO</td>
<td>HBODA</td>
<td>HBOPC</td>
<td>HBOCX</td>
<td>HBOWA</td>
<td>HBOWB</td>
<td>HBOWA</td>
<td>HBOWK</td>
<td>HBOK</td>
<td>(HBODA+HBOPC+HBOCX)/(HBODA+HBOPC+HBOCX)</td>
</tr>
<tr>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA</td>
<td>HBODA/(HBODA+HBOPC+HBOCX)</td>
</tr>
<tr>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC</td>
<td>HBOPC/(HBODA+HBOPC+HBOCX)</td>
</tr>
<tr>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX</td>
<td>HBOCX/(HBODA+HBOPC+HBOCX)</td>
</tr>
</tbody>
</table>

#### Non-Home Based Mode Choice Results

<table>
<thead>
<tr>
<th>Mode Choice</th>
<th>Total</th>
<th>Drive Alone</th>
<th>Carpool 2</th>
<th>Carpool 3+</th>
<th>Walk to Local Transit</th>
<th>Walk to Premium Transit</th>
<th>Drive to Best Available Transit</th>
<th>Non-Motorized Walk</th>
<th>Non-Motorized Bicycle</th>
<th>Average Auto Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>NHB TOTAL</td>
<td>SUMNHB</td>
<td>NHBDA</td>
<td>NHBCP</td>
<td>NHBCX</td>
<td>NHBWA</td>
<td>NHBWB</td>
<td>NHBWA</td>
<td>NHBWK</td>
<td>NHBB</td>
<td>(NHBDA+NHBCP+NHBCX)/(NHBDA+NHBCP+NHBCX)</td>
</tr>
<tr>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA</td>
<td>NHBDA/(NHBDA+NHBCP+NHBCX)</td>
</tr>
<tr>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP</td>
<td>NHBCP/(NHBDA+NHBCP+NHBCX)</td>
</tr>
<tr>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX</td>
<td>NHBCX/(NHBDA+NHBCP+NHBCX)</td>
</tr>
</tbody>
</table>

#### Home Based University Mode Choice Results

<table>
<thead>
<tr>
<th>Mode Choice</th>
<th>Total</th>
<th>Drive Alone</th>
<th>Carpool 2</th>
<th>Carpool 3+</th>
<th>Walk to Local Transit</th>
<th>Walk to Premium Transit</th>
<th>Drive to Best Available Transit</th>
<th>Non-Motorized Walk</th>
<th>Non-Motorized Bicycle</th>
<th>Average Auto Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HBU TOTAL</td>
<td>SUMHBU</td>
<td>HBUDA</td>
<td>HBUCP</td>
<td>HBUCX</td>
<td>HBUBA</td>
<td>HBUBW</td>
<td>HBUBA</td>
<td>HBUBK</td>
<td>HBUBK</td>
<td>(HBUDA+HBUCP+HBUCX)/(HBUDA+HBUCP+HBUCX)</td>
</tr>
<tr>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA</td>
<td>HBUDA/(HBUDA+HBUCP+HBUCX)</td>
</tr>
<tr>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP</td>
<td>HBUCP/(HBUDA+HBUCP+HBUCX)</td>
</tr>
<tr>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX</td>
<td>HBUCX/(HBUDA+HBUCP+HBUCX)</td>
</tr>
</tbody>
</table>

#### Campus University Mode Choice Results

<table>
<thead>
<tr>
<th>Mode Choice</th>
<th>Total</th>
<th>Walk to Local Transit</th>
<th>Non-Motorized Walk</th>
<th>Non-Motorized Bicycle</th>
<th>Average Auto Occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td>DORM TOTAL</td>
<td>SUMUNI</td>
<td>UNIBW</td>
<td>UNIBK</td>
<td>UNIBK</td>
<td>UNIBK/(UNIBW+UNIBK)</td>
</tr>
<tr>
<td>UNIBW</td>
<td>UNIBW</td>
<td>UNIBW</td>
<td>UNIBK</td>
<td>UNIBK</td>
<td>UNIBK/(UNIBW+UNIBK)</td>
</tr>
<tr>
<td>UNIBK</td>
<td>UNIBK</td>
<td>UNIBK</td>
<td>UNIBK</td>
<td>UNIBK</td>
<td>UNIBK/(UNIBW+UNIBK)</td>
</tr>
</tbody>
</table>

### Mode Summary Table as CSV

```
PURPOSE, TOTAL, DA, SR2, SR3+, WALKBUS, WALKPREM, DRIVETRN, WALK, BIKE
```

```python
PRINT CSV=T, LIST='HOME BASED WORK MODE CHOICE RESULTS',
PRINT LIST='HBW TOTAL = SUMHBW, HBWDA, HBWCP, HBWCX, HBWWB, HBWBA, HBWWK, HBWBK, PRINTO=4

PRINT CSV=T, LIST='HOME BASED OTHER MODE CHOICE RESULTS',
PRINT LIST='HBO TOTAL = SUMHBO, HBODA, HBOPC, HBOCX, HBOWA, HBOWB, HBOWK, HBOK, PRINTO=4

PRINT CSV=T, LIST='NON-HOME BASED MODE CHOICE RESULTS',
PRINT LIST='NHB TOTAL = SUMNHB, NHBCP, NHBCX, NHBWA, NHBWB, NHBWK, NHBBK, PRINTO=4

PRINT CSV=T, LIST='HOME BASED UNIVERSITY MODE CHOICE RESULTS',
PRINT LIST='HBU TOTAL = SUMHBU, HBUDA, HBUCP, HBUCX, HBUBA, HBUBW, HBUBK, PRINTO=4

PRINT CSV=T, LIST='CAMPUS UNIVERSITY MODE CHOICE RESULTS',
PRINT LIST='DORM TOTAL = SUMUNI, UNIBW, UNIBK, PRINTO=4
```

### Targets

Page | 118
PRINT CSV=T, LIST= 'TARGETS', PRINTO=2
PRINT CSV=T, LIST= 1.t1_NC_DA(7.5),t1_ST_DA(7.5),t2_NC_DA(7.5),t2_ST_DA(7.5),t3_NC_DA(7.5),t4_NC_DA(7.5),t5_NC_DA(7.5), PRINTO=2
PRINT CSV=T, LIST= 2.t1_NC_CP(7.5),t1_ST_CP(7.5),t2_NC_CP(7.5),t2_ST_CP(7.5),t3_NC_CP(7.5),t4_NC_CP(7.5),t5_NC_CP(7.5), PRINTO=2
PRINT CSV=T, LIST= 3.t1_NC_CX(7.5),t1_ST_CX(7.5),t2_NC_CX(7.5),t2_ST_CX(7.5),t3_NC_CX(7.5),t4_NC_CX(7.5),t5_NC_CX(7.5), PRINTO=2
PRINT CSV=T, LIST= 4.t1_NC_WB(7.5),t1_ST_WB(7.5),t2_NC_WB(7.5),t2_ST_WB(7.5),t3_NC_WB(7.5),t4_NC_WB(7.5),t5_NC_WB(7.5), PRINTO=2
PRINT CSV=T, LIST= 5.t1_NC_WX(7.5),t1_ST_WX(7.5),t2_NC_WX(7.5),t2_ST_WX(7.5),t3_NC_WX(7.5),t4_NC_WX(7.5),t5_NC_WX(7.5), PRINTO=2
PRINT CSV=T, LIST= 6.t1_NC_BA(7.5),t1_ST_BA(7.5),t2_NC_BA(7.5),t2_ST_BA(7.5),t3_NC_BA(7.5),t4_NC_BA(7.5),t5_NC_BA(7.5), PRINTO=2
PRINT CSV=T, LIST= 7.t1_NC_WK(7.5),t1_ST_WK(7.5),t2_NC_WK(7.5),t2_ST_WK(7.5),t3_NC_WK(7.5),t4_NC_WK(7.5),t5_NC_WK(7.5), PRINTO=2
PRINT CSV=T, LIST= 8.t1_NC_BK(7.5),t1_ST_BK(7.5),t2_NC_BK(7.5),t2_ST_BK(7.5),t3_NC_BK(7.5),t4_NC_BK(7.5),t5_NC_BK(7.5), PRINTO=2

; Shares
PRINT CSV=T, LIST= 'MODAL SHARES', PRINTO=2
PRINT CSV=T, LIST= 1.mW401(7.5),mW411(7.5),mW421(7.5),mW431(7.5),mW441(7.5),mW451(7.5),mW171(7.5),mW181(7.5),0(7.5), PRINTO=2
PRINT CSV=T, LIST= 2.mW402(7.5),mW412(7.5),mW422(7.5),mW432(7.5),mW442(7.5),mW452(7.5),mW172(7.5),mW182(7.5),0(7.5), PRINTO=2
PRINT CSV=T, LIST= 3.mW403(7.5),mW413(7.5),mW423(7.5),mW433(7.5),mW443(7.5),mW453(7.5),mW173(7.5),mW183(7.5),0(7.5), PRINTO=2
PRINT CSV=T, LIST= 4.mW404(7.5),mW414(7.5),mW424(7.5),mW434(7.5),mW444(7.5),mW454(7.5),mW174(7.5),mW184(7.5),mW191(7.5), PRINTO=2
PRINT CSV=T, LIST= 5.mW405(7.5),mW415(7.5),mW425(7.5),mW435(7.5),mW445(7.5),mW455(7.5),mW175(7.5),mW185(7.5),0(7.5), PRINTO=2
PRINT CSV=T, LIST= 6.mW406(7.5),mW416(7.5),mW426(7.5),mW436(7.5),mW446(7.5),mW456(7.5),mW176(7.5),mW186(7.5),0(7.5), PRINTO=2
PRINT CSV=T, LIST= 7.mW407(7.5),mW417(7.5),mW427(7.5),mW437(7.5),mW447(7.5),mW457(7.5),mW177(7.5),mW187(7.5),mW192(7.5), PRINTO=2
PRINT CSV=T, LIST= 8.mW408(7.5),mW418(7.5),mW428(7.5),mW438(7.5),mW448(7.5),mW458(7.5),mW178(7.5),mW188(7.5),mW193(7.5), PRINTO=2
PRINT CSV=T, LIST= 'T',mx1,mx2,mx3,mx4,mx5,mx6,mx7,mx8,mx9, PRINTO=2

; print INPUT modal constants
PRINT CSV=T, LIST= 'INPUT CONSTANTS', PRINTO=2
PRINT CSV=T, LIST= 1.0,K1_NC_DA,K1_WC_DA,K1_ST_DA,K2_NC_DA,K2_WC_DA,K2_ST_DA,K3_NC_DA,K4_NC_DA,K5_NC_DA, PRINTO=2
PRINT CSV=T, LIST= 2.0,K1_NC_CP,K1_WC_CP,K1_ST_CP,K2_NC_CP,K2_WC_CP,K2_ST_CP,K3_NC_CP,K4_NC_CP,K5_NC_CP, PRINTO=2
PRINT CSV=T, LIST= 3.0,K1_NC_CX,K1_WC_CX,K1_ST_CX,K2_NC_CX,K2_WC_CX,K2_ST_CX,K3_NC_CX,K4_NC_CX,K5_NC_CX, PRINTO=2
PRINT CSV=T, LIST= 4.0,K1_NC_WB,K1_WC_WB,K1_ST_WB,K2_NC_WB,K2_WC_WB,K2_ST_WB,K3_NC_WB,K4_NC_WB,K5_NC_WB, PRINTO=2
PRINT CSV=T, LIST= 5.0,K1_NC_WX,K1_WC_WX,K1_ST_WX,K2_NC_WX,K2_WC_WX,K2_ST_WX,K3_NC_WX,K4_NC_WX,K5_NC_WX, PRINTO=2
PRINT CSV=T, LIST= 6.0,K1_NC_BA,K1_WC_BA,K1_ST_BA,K2_NC_BA,K2_WC_BA,K2_ST_BA,K3_NC_BA,K4_NC_BA,K5_NC_BA, PRINTO=2
PRINT CSV=T, LIST= 7.0,K1_NC_WK,K1_WC_WK,K1_ST_WK,K2_NC_WK,K2_WC_WK,K2_ST_WK,K3_NC_WK,K4_NC_WK,K5_NC_WK, PRINTO=2
if ((MC_Cal)>1) ; Calibrate?
; -- Revised constants
D=+1.0 ; Damping factor
; HBW No+Car: DA(1) and BA(6) not in set, No WX present so omit from calibration
R1_NC_DA=0
/*
NESTMOTOR = 1.0
NESTTRANSIT = 1.0
NESTNONMOTOR = 1.0
*/

PRINT CSV=T, LIST=
8.0,K1_NC_BK,K1_WC_BK,K1_ST_BK,K2_NC_BK,K2_WC_BK,K2_ST_BK,K3_NC_BK,K4_NC_BK,K5_NC_BK, PRINTO=2

if ({MC_Cal}>1) ; Calibrate?
; -- Revised constants
D=+1.0 ; Damping factor
; HBW No+Car: DA(1) and BA(6) not in set, No WX present so omit from calibration
R1_NC_DA=0
/*
NESTMOTOR = 1.0
NESTTRANSIT = 1.0
NESTNONMOTOR = 1.0
*/

PRINT CSV=T, LIST=
8.0,K1_NC_BK,K1_WC_BK,K1_ST_BK,K2_NC_BK,K2_WC_BK,K2_ST_BK,K3_NC_BK,K4_NC_BK,K5_NC_BK, PRINTO=2

LCP=Ln(MW402/t1_NC_CP)*NESTMOTOR
LCX=Ln(MW403/t1_NC_CX)*NESTMOTOR
LWB=Ln(MW404/t1_NC_WB)*NESTTRANSIT
LWK=Ln(MW407/t1_NC_WK)*NESTNONMOTOR
LBK=Ln(MW408/t1_NC_BK)*NESTNONMOTOR
R1_NC_CP=K1_NC_CP
R1_NC_CX=K1_NC_CX+D*(+LCP-LCX)
R1_NC_WB=K1_NC_WB+D*(+LCP-LWB)
R1_NC_WX=R1_NC_WB ; express same as local
R1_NC_BA=K1_NC_BA
R1_NC_WK=K1_NC_WK+D*(+LCP-LWK)
R1_NC_BK=K1_NC_BK+D*(+LCP-LBK)
/*
NESTMOTOR = 1.0
NESTTRANSIT = 1.0
NESTNONMOTOR = 1.0
*/

LDA=Ln(MW411/t1_WC_DA)*NESTMOTOR
LCP=Ln(MW412/t1_WC_CP)*NESTMOTOR
LCX=Ln(MW413/t1_WC_CX)*NESTMOTOR
LWB=Ln(MW414/t1_WC_WB)*NESTTRANSIT
LBA=Ln(MW416/t1_WC_BA)*NESTTRANSIT
LWK=Ln(MW417/t1_WC_WK)*NESTNONMOTOR
LBK=Ln(MW418/t1_WC_BK)*NESTNONMOTOR
R1_WC_CP=K1_WC_CP+D*(+LDA-LCP)
R1_WC_CX=K1_WC_CX+D*(+LDA-LCX)
R1_WC_WB=K1_WC_WB+D*(+LDA-LWB)
R1_WC_WX=R1_WC_WB ; express same as local
R1_WC_BA=K1_WC_BA+D*(+LDA-LBA)
R1_WC_WK=K1_WC_WK+D*(+LDA-LWK)
R1_WC_BK=K1_WC_BK+D*(+LDA-LBK)
/*
NESTMOTOR = 1.0
NESTTRANSIT = 1.0
NESTNONMOTOR = 1.0
*/

LDA=Ln(MW421/t1_ST_DA)*NESTMOTOR
LCP=Ln(MW422/t1_ST_CP)*NESTMOTOR
LCX=Ln(MW423/t1_ST_CX)*NESTMOTOR
LBA=Ln(MW423/t1_ST_BA)*NESTTRANSIT
LWK=Ln(MW424/t1_ST_WK)*NESTNONMOTOR
LBK=Ln(MW425/t1_ST_BK)*NESTNONMOTOR
R1_ST_CP=K1_ST_CP+D*(+LDA-LCP)
R1_ST_CX=K1_ST_CX+D*(+LDA-LCX)
R1_ST_WB=K1_ST_WB+D*(+LDA-LWB)
R1_ST_WX=R1_ST_WB ; express same as local
R1_ST_BA=K1_ST_BA+D*(+LDA-LBA)
R1_ST_WK=K1_ST_WK+D*(+LDA-LWK)
R1_ST_BK=K1_ST_BK+D*(+LDA-LBK)
; HBO No+Car: DA(1) and BA(6) not in set, No WX present so omit from calibration
;R2_NC_DA=K2_NC_DA
R2_NC_DA=0

LCP=Ln(MW432/t2_NC_CP)*NESTMOTOR
LCX=Ln(MW433/t2_NC_CX)*NESTMOTOR
LWB=Ln(MW434/t2_NC_WB)*NESTTRANSIT
LWK=Ln(MW437/t2_NC_WK)*NESTNONMOTOR
LBK=Ln(MW438/t2_NC_BK)*NESTNONMOTOR

R2_NC_CP=K2_NC_CP
R2_NC_CX=K2_NC_CX+D*(+LCP-LCX)
R2_NC_WB=K2_NC_WB+D*(+LCP-LWB)

R2_NC_WK=R2_NC_WB ; express same as local
R2_NC_BA=K2_NC_BA
R2_NC_WK=K2_NC_WK+D*(+LCP-LWK)
R2_NC_BK=K2_NC_BK+D*(+LCP-LBK)

; HBO With+Car: No WX present so omit from calibration
;R2_WC_DA=K2_WC_DA-
Ln(MW441/t2_WC_DA)+Ln(MW442/t2_WC_CP)+Ln(MW444/t2_WC_CX)+Ln(MW444/t2_WC_WB)+Ln(MW446/t2_WC_BA)+Ln
(MW447/t2_WC_WK)+Ln(MW448/t2_WC_BK)
R2_WC_DA=0

LDA=Ln(MW441/t2_WC_DA)*NESTMOTOR
LCX=Ln(MW442/t2_WC_CX)*NESTMOTOR
LWB=Ln(MW444/t2_WC_WB)*NESTTRANSIT
LBA=Ln(MW446/t2_WC_BA)*NESTNONMOTOR
LWK=Ln(MW447/t2_WC_WK)*NESTNONMOTOR
LBK=Ln(MW448/t2_WC_BK)*NESTNONMOTOR

R2_WC_CP=K2_WC_CP+D*(+LDA-LCP)
R2_WC_CX=K2_WC_CX+D*(+LDA-LCX)
R2_WC_WB=K2_WC_WB+D*(+LDA-LWB)
R2_WC_WK=R2_WC_WB ; express same as local
R2_WC_BA=K2_WC_BA+D*(+LDA-LBA)
R2_WC_WK=K2_WC_WK+D*(+LDA-LWK)
R2_WC_BK=K2_WC_BK+D*(+LDA-LBK)

; HBO Student: No WX present so omit from calibration
;R2_ST_DA=K2_ST_DA-
Ln(MW451/t2_ST_DA)+Ln(MW452/t2_ST_CP)+Ln(MW453/t2_ST_CX)+Ln(MW454/t2_ST_WB)+Ln(MW456/t2_ST_BA)+Ln
(MW457/t2_ST_WK)+Ln(MW458/t2_ST_BK)
R2_ST_DA=0

LDA=Ln(MW451/t2_ST_DA)*NESTMOTOR
LCX=Ln(MW452/t2_ST_CX)*NESTMOTOR
LWB=Ln(MW454/t2_ST_WB)*NESTTRANSIT
LBA=Ln(MW456/t2_ST_BA)*NESTNONMOTOR
LWK=Ln(MW457/t2_ST_WK)*NESTNONMOTOR
LBK=Ln(MW458/t2_ST_BK)*NESTNONMOTOR

R2_ST_CP=K2_ST_CP+D*(+LDA-LCP)
R2_ST_CX=K2_ST_CX+D*(+LDA-LCX)
R2_ST_WB=K2_ST_WB+D*(+LDA-LWB)
R2_ST_WK=R2_ST_WB ; express same as local
R2_ST_BA=K2_ST_BA+D*(+LDA-LBA)
R2_ST_WK=K2_ST_WK+D*(+LDA-LWK)
R2_ST_BK=K2_ST_BK+D*(+LDA-LBK)

; NHB: No WX present so omit from calibration
;R3_NC_DA=K3_NC_DA-
Ln(MW171/t3_NC_DA)+Ln(MW172/t3_NC_CP)+Ln(MW173/t3_NC_CX)+Ln(MW174/t3_NC_WB)+Ln(MW176/t3_NC_BA)+Ln
(MW177/t3_NC_WK)+Ln(MW178/t3_NC_BK)
R3_NC_DA=0

LDA=Ln(MW171/t3_NC_DA)*NESTMOTOR
LCX=Ln(MW173/t3_NC_CX)*NESTMOTOR
LWB=Ln(MW174/t3_NC_WB)*NESTTRANSIT
LBA = Ln(MW176/t3_NC_BA) * NESTTRANSIT
LWK = Ln(MW177/t3_NC_WK) * NESTNONMOTOR
LBK = Ln(MW178/t3_NC_BK) * NESTNONMOTOR

R3_NC_CP = K3_NC_CP + D*(+LDA-LCP)
R3_NC_CX = K3 NC_CX + D*(+LDA-LCX)
R3_NC_WB = K3 NC_WB + D*(+LDA-LWB)
R3_NC_WX = R3_NC_WB; express same as local
R3_NC_BA = K3_NC_BA + D*(+LDA-LBA)
R3_NC_WK = K3_NC_WK + D*(+LDA-LWK)
R3_NC_BK = K3_NC_BK + D*(+LDA-LBK)

; HBU: No WX present so omit from calibration
R4_NC_DA = K4_NC_DA - Ln(MW181/t4_NC_DA) + Ln(MW182/t4 NC_CP) + Ln(MW183/t4 NC_CX) + Ln(MW184/t4 NC_WB) + Ln(MW186/t4 NC_BA) + Ln(MW187/t4 NC_WK) + Ln(MW188/t4 NC_BK)
R4 NC_DA = 0
LDA = Ln(MW181/t4 NC_DA) * NESTMOTOR
LCP = Ln(MW182/t4 NC_CP) * NESTMOTOR
LCX = Ln(MW183/t4 NC_CX) * NESTMOTOR
LWB = Ln(MW184/t4 NC_WB) * NESTTRANSIT
LWK = Ln(MW186/t4 NC_BA) * NESTNONMOTOR
LBK = Ln(MW188/t4 NC_BK) * NESTNONMOTOR

R4_NC_CP = K4_NC_CP + D*(+LDA-LCP)
R4_NC_CX = K4_NC_CX + D*(+LDA-LCX)
R4_NC_WB = K4_NC_WB + D*(+LDA-LWB)
R4_NC_WX = R4_NC_WB; express same as local
R4_NC_BA = K4_NC_BA + D*(+LDA-LBA)
R4_NC_WK = K4_NC_WK + D*(+LDA-LWK)
R4_NC_BK = K4_NC_BK + D*(+LDA-LBK)

; DORM: AUTO, PNR and WX NOT IN CHOICE SET. No WX present so omit from calibration
R5_NC_DA = 0
LWB = Ln(MW191/t5 NC_WB) * NESTTRANSIT
LWK = Ln(MW192/t5 NC_WK) * NESTNONMOTOR
LBK = Ln(MW193/t5 NC_BK) * NESTNONMOTOR

R5_NC_CP = 0
R5_NC_CX = 0
R5_NC_WB = K5_NC_WB
R5_NC_WX = 0
R5_NC_BA = 0
R5_NC_WK = K5_NC_WK + D*(+LWB-LWK)
R5_NC_BK = K5_NC_BK + D*(+LWB-LBK)

; print REVISED modal constants
PRINT CSV=T, LIST=", REVISED CONSTANTS", PRINTO=2
PRINT CSV=T, LIST=", 1.0, R1 NC DA(10.5), R1 NC DA(10.5), R1 ST DA(10.5), R2 NC DA(10.5), R2 ST DA(10.5), R3
NC DA(10.5), R4 NC DA(10.5), R5 NC DA(10.5), PRINTO=2
PRINT CSV=T, LIST=", 2.0, R1 NC CP(10.5), R1 NC CP(10.5), R1 ST CP(10.5), R2 NC CP(10.5), R2 ST CP(10.5), R3
NC CP(10.5), R4 NC CP(10.5), R5 NC CP(10.5), PRINTO=2
PRINT CSV=T, LIST=", 3.0, R1 NC CX(10.5), R1 NC CX(10.5), R1 ST CX(10.5), R2 NC CX(10.5), R2 ST CX(10.5), R3
NC CX(10.5), R4 NC CX(10.5), R5 NC CX(10.5), PRINTO=2
PRINT CSV=T, LIST=", 4.0, R1 NC WB(10.5), R1 NC WB(10.5), R1 ST WB(10.5), R2 NC WB(10.5), R2 ST WB(10.5), R3
NC WB(10.5), R4 NC WB(10.5), R5 NC WB(10.5), PRINTO=2
PRINT CSV=T, LIST=", 5.0, R1 NC WX(10.5), R1 NC WX(10.5), R1 ST WX(10.5), R2 NC WX(10.5), R2 ST WX(10.5), R3
NC WX(10.5), R4 NC WX(10.5), R5 NC WX(10.5), PRINTO=2
PRINT CSV=T, LIST=", 6.0, R1 NC BA(10.5), R1 NC BA(10.5), R1 ST BA(10.5), R2 NC BA(10.5), R2 ST BA(10.5), R3
NC BA(10.5), R4 NC BA(10.5), R5 NC BA(10.5), PRINTO=2
PRINT CSV=T, LIST=", 7.0, R1 NC WK(10.5), R1 NC WK(10.5), R1 ST WK(10.5), R2 NC WK(10.5), R2 ST WK(10.5), R3
NC WK(10.5), R4 NC WK(10.5), R5 NC WK(10.5), PRINTO=2"
PRINT CSV=T, LIST=
8.0,R1 NC BK(10.5), R1 WC BK(10.5), R1 ST BK(10.5), R2 NC BK(10.5), R2 WC BK(10.5), R2 ST BK(10.5), R3 NC BK(10.5), R4 NC BK(10.5), R5 NC BK(10.5), PRINTO=2

; print REVISED modal constants
PRINT CSV=T, LIST=
1.0,R1 NC DA(10.5), R1 WC DA(10.5), R1 ST DA(10.5), R2 NC DA(10.5), R2 WC DA(10.5), R2 ST DA(10.5), R3 NC DA(10.5), R4 NC DA(10.5), R5 NC DA(10.5), PRINTO=3
PRINT CSV=T, LIST=
2.0,R1 NC CP(10.5), R1 WC CP(10.5), R1 ST CP(10.5), R2 NC CP(10.5), R2 WC CP(10.5), R2 ST CP(10.5), R3 NC CP(10.5), R4 NC CP(10.5), R5 NC CP(10.5), PRINTO=3
PRINT CSV=T, LIST=
3.0,R1 NC CX(10.5), R1 WC CX(10.5), R1 ST CX(10.5), R2 NC CX(10.5), R2 WC CX(10.5), R2 ST CX(10.5), R3 NC CX(10.5), R4 NC CX(10.5), R5 NC CX(10.5), PRINTO=3
PRINT CSV=T, LIST=
4.0,R1 NC WB(10.5), R1 WC WB(10.5), R1 ST WB(10.5), R2 NC WB(10.5), R2 WC WB(10.5), R2 ST WB(10.5), R3 NC WB(10.5), R4 NC WB(10.5), R5 NC WB(10.5), PRINTO=3
PRINT CSV=T, LIST=
5.0,R1 NC WX(10.5), R1 WC WX(10.5), R1 ST WX(10.5), R2 NC WX(10.5), R2 WC WX(10.5), R2 ST WX(10.5), R3 NC WX(10.5), R4 NC WX(10.5), R5 NC WX(10.5), PRINTO=3
PRINT CSV=T, LIST=
6.0,R1 NC BA(10.5), R1 WC BA(10.5), R1 ST BA(10.5), R2 NC BA(10.5), R2 WC BA(10.5), R2 ST BA(10.5), R3 NC BA(10.5), R4 NC BA(10.5), R5 NC BA(10.5), PRINTO=3
PRINT CSV=T, LIST=
7.0,R1 NC WK(10.5), R1 WC WK(10.5), R1 ST WK(10.5), R2 NC WK(10.5), R2 WC WK(10.5), R2 ST WK(10.5), R3 NC WK(10.5), R4 NC WK(10.5), R5 NC WK(10.5), PRINTO=3
PRINT CSV=T, LIST=
8.0,R1 NC BK(10.5), R1 WC BK(10.5), R1 ST BK(10.5), R2 NC BK(10.5), R2 WC BK(10.5), R2 ST BK(10.5), R3 NC BK(10.5), R4 NC BK(10.5), R5 NC BK(10.5), PRINTO=3

endif; MC Cal
ENDIF
ENDRUN

MCMAT00C.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX PRNFILE="(SCENARIO_DIR)\output\FINALTABLES.PRN" MSG='Creates Tables for Loading'
FILEI MATI[3] = "(SCENARIO_DIR)\output\EETAB.MAT"
FILEI MATI[4] = "(SCENARIO_DIR)\output\ETRIPS.MAT"
FILEO MATO[5] = "(SCENARIO_DIR)\output\VHSBYUP_NON.MAT",
MO=41,42,51,52,61,62,
NAME=UF_light, UF_heavy, Non_Light, Non_heavy, SelZones_Light, SelZones_heavy, DEC=6*S
FILEI ZDATI[1] = "(SCENARIO_DIR)\input\zoneData\YEAR.DBF",
Z=TAZ_20\year
FILEO MATO[4] = "(SCENARIO_DIR)\output\IAUTOPEPERSONS.MAT"
MO=31, NAME=1AutoPersons, DEC=5*S
FILEO MATO[3] = "{SCENARIO_DIR}\output\NONMOTOR.MAT",
MO=21-22, NAME=WALK,BIKE, DEC=2*S
FILEO MATO[2] = "{SCENARIO_DIR}\output\TRANSIT.MAT",
MO=11-16,
NAME=PKWALKLOCAL,PKWALKPREM,PKAUTOBA,OPWALKLOCAL,OPWALKPREM,OPAUTOBA, DEC=6*S
FILEO MATO[1] = "{SCENARIO_DIR}\output\VEHICLES.MAT",
MO=1-5,
NAME=DRIVEALONE,CARPPOOL,LIGHTTRUCK,HEAVYTRUCK,EETRIPS, DEC=5*S
FILEI M ATI[1] = "{SCENARIO_DIR}\output\MODEOUT.MAT"

PARAMETERS ZONEMSG=100

; DRIVE ALONE
MW[1]=((MI.1.HBWDAA+MI.1.HBWWA.T)/1.0+
(MI.1.HBODA+MI.1.HBODA.T)/1.0+
(MI.1.NHBDA+MI.1.NHBDA.T)/1.0+
(MI.1.HBUUDA+MI.1.HBUUDA.T)/1.0+
(MI.2.SOVIE+MI.2.SOVIE.T)/1.0)*0.50

; CARPOOL
MW[2]=((MI.1.HBWCPC+MI.1.HBWCPC.T)/2.0+
(MI.1.HBOCP+MI.1.HBOCP.T)/2.0+
(MI.1.NHBCP+MI.1.NHBCP.T)/2.0+
(MI.1.HBUCP+MI.1.HBUCP.T)/2.0+
(MI.1.HBOCX+MI.1.HBOCX.T)/{HBW3P}+
(MI.1.HBUCX+MI.1.HBUCX.T)/{HBW3P}+ ; ASSUME 3+ occ same as work
(MI.1.HBOCXX+MI.1.HBOCXX.T)/{HBW3P}+
(MI.1.NHBXX+MI.1.NHBXX.T)/{HBW3P}+
(MI.1.HBUXX+MI.1.HBUXX.T)/{HBW3P}+ ; ASSUME 3+ occ same as work
(MI.2.HOVIE+MI.2.HOVIE.T)/1.0)*0.50

; LIGHT DUTY TRUCKS

; HEAVY DUTY TRUCKS
MW[4]=((MI.2.TRUCKTRLR+MI.2.TRUCKTRLR.T)+(MI.2.TRUCKHDIE+MI.2.TRUCKHDIE.T))*0.50 ; 0.5 added by KDK

; EETRIPS
MW[5]=MI.3.1

; TRANSIT, PEAK PERIOD
MW[011]=MI.1.HBWB
MW[012]=MI.1.HBWX
MW[013]=MI.1.HBWA

; TRANSIT, OFF-PEAK PERIOD

; NON-MOTORIZED
MW[022]=MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HBBK+MI.1.HDORMUBK

; Internal Auto Persons
MW[31]= MI.1.HBWDA+ MI.1.HBODA+ MI.1.NHBDA+
MI.1.HBUUDA+ MI.1.HBUCP+ MI.1.HBOCP+
MI.1.NHBCP+ MI.1.HBWCX+ MI.1.HBWX+
MI.1.HBOCX+ MI.1.HBUCX+ MI.1.HBUX

; Select Zone Vehicle Trips
JLOOP
if(i=1 & j=1) MW[011]=MW[011]+0.01
if(i=1 & j=1) MW[012]=MW[012]+0.01
if(i=1 & j=1) MW[013]=MW[013]+0.01
if(i=1 & j=1) MW[014]=MW[014]+0.01
if(i=1 & j=1) MW[015]=MW[015]+0.01
if(i=1 & j=1) MW[016]=MW[016]+0.01
if((ZI.1.SELECTZONE[J]=1) || (ZI.1.SELECTZONE[I]=1)) ; Select Zones
endif

; UF vs Non-UF Vehicle Trips
if((ZI.1.UFZONES[J]=1) || (ZI.1.UFZONES[I]=1)) ; UF related
else
endif
ENDIFLOOP
ENDIFRUN

Assignment Step

ASHWY000A.S
; Do not change filenames or add or remove FILEI/FIELO statements using an editor. Use
Cube/Application Manager.
RUN PGM=HIGHWAY PRFILE=\"{SCENARIO_DIR}\output\HASIGN.PRN\" MSG='Peak Season Weekday (24 hr)
 Assignment'
FILEI NETI = \"(SCENARIO_DIR)\output\UNLOADED.NET\"
FILEO TURMVLO = \"(SCENARIO_DIR)\output\TURNVOL.BIN\",
FORMAT=DFP
FILEI MATI[1] = \"(SCENARIO_DIR)\output\VEHSBYUF_NON.MAT\"
FILEO NETO = \"(SCENARIO_DIR)\output\HASIGN.NET\"
FILEO PATHO[1] = \"(SCENARIO_DIR)\output\AUTOS.PTH\"
FILEI TURNPENI = \"(SCENARIO_DIR)\input\TCARDS.PEN\"
parameters zonemsg=100
PROCESS PHASE=LINKREAD
; USE THE USER SUPPLIED ALPHA AND BETA FOR THE BPR CURVE
IF (LI.BPRCOEFFICIENT=0)
  LW.BPRCOEFFICIENT=0.15
ELSE
  LW.BPRCOEFFICIENT=LI.BPRCOEFFICIENT
ENDIF
IF (LI.BPREXONENT=0)
  LW.BPREXONENT=4.0
ELSE
  LW.BPREXONENT=LI.BPREXONENT
ENDIF
IF (LI.CAPACITY=0)
  LW.DAILYCAP=999999
ELSE
  LW.DAILYCAP=(LI.CAPACITY/li.confac)*li.uroadfactor
ENDIF
IF (LI.TIME=0)
  LW.FFTIME=0.00001
ELSE
  LW.FFTIME=LI.TIME
ENDIF
C=LW.DAILYCAP
T0=LW.FFTIME
IF (LI.FTYPE=49) ADDTOGROUP=1
ENDPROCESS

PROCESS PHASE=LOOP
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update

Technical Report No. 4: 2010 Model Update and Validation

\[ MW[1]=MI.1.UF\_light \]
\[ MW[2]=MI.1.UF\_heavy\{PCE\_HT\} \]
\[ MW[3]=MI.1.Non\_light \]
\[ MW[4]=MI.1.Non\_heavy\{PCE\_HT\} \]
\[ MW[6]=MI.1.SelZones\_heavy\{PCE\_HT\} \]

\textbf{PATHLOAD PATH\_TIME,}
\[ \text{VOL}[1]=MW[1], \text{VOL}[2]=MW[2], \text{VOL}[3]=MW[3], \text{VOL}[4]=MW[4], \text{VOL}[5]=MW[5], \text{VOL}[6]=MW[6], \text{PENI}=1, \text{EXCLUDEGROUP}=1, \]
\[ \text{PATHO}=1, \text{ALLJ}=T, \text{INCLUDECOST}=F, \text{NAME}=\text{ALLTRIPS} \]
ENDPROCESS

PROCESS PHASE\_ADJUST
\[ V = \text{VOL}[1] + \text{VOL}[2] + \text{VOL}[3] + \text{VOL}[4] \]
\[ \text{FUNCTION } TC[1]=70*(1+\text{LW.BPRECOEFFICIENT}*(\text{V}/\text{C})^\text{LW.BPREXONENT}) ; \text{congested time equation, no toll model in place} \]
ENDPROCESS
ENDRUN

\textbf{ASHWY008.S}

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=HIGHWAY PRNFILE="{SCENARIO\_DIR}\output\ASHWY00A.PRN" MSG=’Non-Motorized Assignment’
FILEI NETI = "{SCENARIO\_DIR}\output\UNLOADED.NET"
FILEO PATHO[1] = "{SCENARIO\_DIR}\output\NONMOTOR.PTH"
FILEO NETO = "{SCENARIO\_DIR}\output\NONMOTOR.NET"
FILEI MATI[1] = "{SCENARIO\_DIR}\output\NONMOTOR.MAT"

PARAMETERS MAXITERS=1

PROCESS PHASE=LINKREAD
IF (LI.FTYPE=10-19,70-99) ADDGROUP=1
ENDPROCESS

PROCESS PHASE=ILOOP
\[ \text{PATHLOAD } \text{PATH}\_\text{DISTANCE}, \text{VOL}[1]=\text{MI.1.WALK}, \text{VOL}[2]=\text{MI.1.BIKE}, \text{EXCLUDEGROUP}=1, \]
\[ \text{PATHO}=1, \text{NAME}=\text{'NONMOTOR'}, \text{ALLJ}=T, \text{INCLUDECOSTS}=F \]
ENDPROCESS
ENDRUN
ASPTR00A.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRNFILE="(SCENARIO_DIR)\output\TRANSIT_AMWALK.PRN" MSG='Peak Period Walk to Transit Assignment'
FILEI ROUTE1[2] = "(SCENARIO_DIR)\output\WALKPREMAM.RTE"
FILEI ROUTE1[1] = "(SCENARIO_DIR)\output\WALKLBAM.RTE"
FILEI NETI = "(SCENARIO_DIR)\output\TNETWALKAM.NET"
FILEO LINKO = "(SCENARIO_DIR)\output\TLOAD1.DBF",
SKIP0=Y,NTLEGS=F
FILEI MATI[1] = "(SCENARIO_DIR)\output\TRANSIT.MAT"
FILEO REPORTO = "(SCENARIO_DIR)\output\ASPTR00B.PRN"
FILEO NETO = "(SCENARIO_DIR)\output\TLOADAM1.NET"
PARAMETERS USERCLASSES=1,2,
TRIPSJ[1]=(MI.1.PKWALKLOCAL),
TRIPSJ[2]=(MI.1.PKWALKREM),
NOROUTEERRS=9999999,
NOROUTEMSGS=0
;Selection of Loading Reports
REPORT LINES=T; LINEVOLS=T STOPONLY=T
PAGEHEIGHT=32767
ENDRUN

ASPTR00B.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRNFILE="(SCENARIO_DIR)\output\TRANSIT_AMAUTO.PRN" MSG='Peak Period Auto to Transit Assignment'
FILEI NETI = "(SCENARIO_DIR)\output\TNETAUTOAM.NET"
FILEO LINKO = "(SCENARIO_DIR)\output\TLOAD2.DBF",
SKIP0=Y,NTLEGS=F
FILEO REPORT0 = "(SCENARIO_DIR)\output\ASPTR00E.PRN"
FILEO NETO = "(SCENARIO_DIR)\output\TLOADAM1.NET"
FILEI MATI[1] = "(SCENARIO_DIR)\output\TRANSIT.MAT"
FILEI ROUTE1[1] = "(SCENARIO_DIR)\output\AUTOALLAM.RTE"
PARAMETERS USERCLASSES=1,
   TRIPSIJ=(MI.1.PKAUTOB),
   NOROUTERRS=999999,
   NOROUTEMSGS=0
REPORT LINES=T ; LINEVOLS=T
PAGEHEIGHT=32767
ENDRUN

ASPTR00E.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRNFILE="(SCENARIO_DIR)\output\TRANSIT_MDWALK.PRN" MSG='Off-Peak Period Walk to Transit Assignment'
FILEI ROUTE1[2] = "(SCENARIO_DIR)\output\WALKPREMMD.RTE"
FILEI NETI = "(SCENARIO_DIR)\output\TNETWALKMD.NET"
FILEO LNKI = "(SCENARIO_DIR)\output\TLOAD3.DBF",
   SKIP0=Y, NTLEGGS=F
FILEI ROUTE1[1] = "(SCENARIO_DIR)\output\WALKLBMD.RTE"
FILEI MATI[1] = "(SCENARIO_DIR)\output\TRANSIT.MAT"
FILEO REPORT0 = "(SCENARIO_DIR)\output\ASPTR00F.PRN"
FILEO NETO = "(SCENARIO_DIR)\output\TLOADMD1.NET"
PARAMETERS USERCLASSES=1,2,
   TRIPSIJ1=(MI.1.OPWALKLOCAL),
   TRIPSIJ2=(MI.1.OPWALKPREM)
   NOROUTERRS=999999,
   NOROUTEMSGS=0, HDWAYPERIOD=2
REPORT LINES=T ; LINEVOLS=T
PAGEHEIGHT=32767
ENDRUN
ASPTR00F.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=PUBLIC TRANSPORT PRNFILE="{SCENARIO_DIR}\output\TRANSIT_MDAUTO.PRN" MSG='Off-Peak Period Auto to Transit Assignment'
FILEI ROUTE1[1] = "{SCENARIO_DIR}\output\AUTOALLMD.RTE"
FILEO LINKO = "{SCENARIO_DIR}\output\TLD04.DBF",
SKIP0=Y, N
FILEI NETI = "{SCENARIO_DIR}\output\TNETAUTOMD.NET"
FILEI MATI[1] = "{SCENARIO_DIR}\output\TRANSIT.MAT"
FILEO REPORTO = "{SCENARIO_DIR}\output\ASPTR00H.PRN"
FILEO NETO = "{SCENARIO_DIR}\output\TLOADMD2.NET"
PARAMETERS USERCLASSES=1,
TRIPSJ=MI.1.OPAUTOB,
NOROUTEERRS=999999,
NOROUTEMSGS=0, HDWAYPERIOD=2
REPORT LINES=T ;LINEVOLS=T
PAGEHEIGHT=32767
ENDRUN

ASMAT00A.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX
FILEO RECO[1] = "{SCENARIO_DIR}\output\L1L.DBF",
FIELDS=A,B,MODE,NAME,DIST,TIME,SEQ,CNT,HEADWAY_1,VOL
FILEI RECI = "{SCENARIO_DIR}\output\TLOAD1.DBF"
vtot=vtot+ri.VOL
if (ri.SEQ==ri.CNT)
  A = ri.A
  B = ri.B
  MODE = ri.MODE
  NAME = 'COMBINED'
  DIST = ri.DIST
  TIME = ri.TIME
  SEQ = 1
  CNT = 1
ASMAT00B.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX
FILEI RECI = "\{SCENARIO\_DIR\}output\TLOD2.DBF"
FILEO RECO[1] = "\{SCENARIO\_DIR\}output\LL2.DBF",
  FIELDS=A,B,MODE,NAM,DIST,TIME,SEQ,CNT,HEADWAY_1,VOL
vtot=vtot+ri.VOL
if (ri.SEQ==ri.CNT)
  RO.A = ri.A
  RO.B = ri.B
  RO.MODE = ri.MODE
  RO.NAME = 'COMBINED'
  RO.DIST = ri.DIST
  RO.TIME = ri.TIME
  RO.SEQ = 1
  RO.CNT = 1
  RO.HEADWAY = ri.HEADWAY_1
  RO.VOL = vtot
  vtot = 0.0
  WRITE RECO=1
endif
ENDRUN

ASMAT00C.S

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=MATRIX
FILEI RECI = "\{SCENARIO\_DIR\}output\TLOD3.DBF"
FILEO RECO[1] = "\{SCENARIO\_DIR\}output\LL3.DBF",
  FIELDS=A,B,MODE,NAM,DIST,TIME,SEQ,CNT,HEADWAY_2,VOL
vtot=vtot+ri.VOL
if (ri.SEQ==ri.CNT)
  RO.A = ri.A
  RO.B = ri.B
  RO.MODE = ri.MODE
  RO.NAME = 'COMBINED'

Page | 130
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update  
*Technical Report No. 4: 2010 Model Update and Validation*

```plaintext
RO.DIST = ri.DIST
RO.TIME = ri.TIME
RO_SEQ = 1
RO_CNT = 1
RO.HEADWAY = "ri.HEADWAY_2"
RO.VOL = vtot
vtot = 0.0
WRITE RECO=1
endif
ENDRUN

**ASMAT00D.S**

; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use  
Cube/Application Manager.
RUN PGM-MATRIX
FILEI RECI = "\{SCENARIO_DIR\}output\TLOD4.DBF"
FILEO RECO[1] = "\{SCENARIO_DIR\}output\LL4.DBF",
  FIELDS=A,B,MODE,NAME,DIST,TIME,SEQ,CNT,HEADWAY_2,VOL
vtot=vtot+ri.VOL
if (ri.SEQ==ri.CNT)
  RO.A = ri.A
  RO.B = ri.B
  RO.MODE = ri.MODE
  RO.NAME = 'COMBINED'
  RO.DIST = ri.DIST
  RO.TIME = ri.TIME
  RO_SEQ = 1
  RO_CNT = 1
  RO.HEADWAY = "ri.HEADWAY_2"
  RO.VOL = vtot
  vtot = 0.0
  WRITE RECO=1
endif
ENDRUN
```

[Diagram of network analysis process]

Page | 131
ASNET00A.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=NETWORK PROFILE="{SCENARIO_DIR}\output\ASNET00A.PRN" MSG='Merge Motorized & Non-motorized'
FILEO NETO = "{SCENARIO_DIR}\output\COMB_TEMP.NET"
;FILEO PRINTO[1] = "{SCENARIO_DIR}\output\UFOUCH.PRN"
FILEI LINKI[6] = "{SCENARIO_DIR}\output\LL4.DBF"
FILEI LINKI[5] = "{SCENARIO_DIR}\output\LL3.DBF"
FILEI LINKI[4] = "{SCENARIO_DIR}\output\LL2.DBF"
FILEI LINKI[3] = "{SCENARIO_DIR}\output\LL1.DBF"
FILEI LINKI[2] = "{SCENARIO_DIR}\output\NONMOTOR.NET"
FILEI LINKI[1] = "{SCENARIO_DIR}\output\HASSIGN.NET"

PROCESS PHASE=LINKMERGE

NONMOTORVOL=LI.2.V_1
CGSPEED=LI.1.CSPD_1
CGTIME=LI.1.TIME_1
_USF_LIGHT=LI.2.V_1
_USF_HEAVY=LI.1.V2_1/{PCE_HT}
_NON_LIGHT=LI.1.V3_1
_NON_HEAVY=LI.1.V4_1/{PCE_HT}
_SELZONE_LIGHT=LI.1.V5_1
_SELZONE_HEAVY=LI.1.V6_1/{PCE_HT}

SELZONE_MOTOR=_SELZONE_LIGHT+_SELZONE_HEAVY

UF_MOTOR= UF_LIGHT+ UF_HEAVY
LIGHTVEHICLES= UF_LIGHT+ NON_LIGHT
HEAVYTRUCKS= UF_HEAVY+ NON_HEAVY
MOTORIZEDVOL=LIGHTVEHICLES+HEAVYTRUCKS
if (MOTORIZEDVOL>0.0)
  UFPCT=100*UF_MOTOR/MOTORIZEDVOL
endif
VMT=MOTORIZEDVOL*DISTANCE
VHT=MOTORIZEDVOL*CGTIME/60.
PEDESTRIANS=LI.2.V1_1
BICYCLISTS=LI.2.V2_1
VOL_CAP=MOTORIZEDVOL/DAILYCAP
if (CAPACITY=0)
  DAILYCAPE=999999
else
  DAILYCAPE= 10.0*CAPACITY
endif
VOL_CAPE=MOTORIZEDVOL/DAILYCAPE
TranVol=li.3.vol+li.4.vol+li.5.vol+li.6.vol
if(COUNT{year} > 0)
  VC=MOTORIZEDVOL/COUNT{year}
endif
ENDPROCESS
ENDRUN
ASNET00E.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM-NETWORK PNRFILE="C:\Users\mgao\Documents\Gainesville 2040 LRTP\Project Management\08-
Model\Gainesville_2010_2040_run7_Final\APPLICATIONS\ASNET00C.PRN" MSG="Network Cleaning"
FILEO NETO = "{SCENARIO_DIR}\output\COMBINEDLOADED.NET",
EXCLUDE=VOL,MODE,DIST,TIME,SEQ,CNT,HEADWAY_1,HEADWAY_2,SECNUM,TWOWAY,
FTYPE1,DIR,ATYPE1,BPCOEFFICIENT,V1,VC,CSFPD,VT,VI,V2,VI,VT,VI,VT_1,
V2T_1,V3_1,V4_1,V5_1,VI,V3T_1,V4T_1,V5T_1,VI,VT,ATYPE1,TIME,URoadFactor,
BPCOEFFICIENT,BPREXponent
FILEI LINKI[1] = "{SCENARIO_DIR}\output\COMB_TEMP.NET"
ENDRUN

ASNET00C.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use
Cube/Application Manager.
RUN PGM-NETWORK PNRFILE="{SCENARIO_DIR}\output\HEVAL_Daily.PRN" MSG="Highway Evaluation Scripts"
FILEO LINKO = "{SCENARIO_DIR}\output\Daily_Links.DBF"
FILEI LINKI[1] = "{SCENARIO_DIR}\output\COMBINEDLOADED.NET"
FILEO PRINTO[1] = "{SCENARIO_DIR}\output\RMSE.PRN"

; ------------------------------------------------------------------------
; LINKMERGE PHASE
; ------------------------------------------------------------------------
PHASE-LINKMERGE

; DUMMY VARIABLES FOR HEVALDBF
ZONE = 1
USECODE = 1
LOCATION = 1
LANDUSE = 1
CCODE = 1
TOTCNT = li.1.COUNT(year)
CAP = li.1.CAPACITY*li.1.LANES/li.1.CONFAC

;
ft=int(li.1.FTYPE/10)
at=int(li.1.ATYPE/10)
LNS-li.1.LANES
TOTAL_VOL=ROUND(li.1.MOTORIZEDVOL)
;
; initialize arrays and variables
ARRAY _err=13, _cns=13, _cnt=13, _RGP=13, _volbyft=100, _cntbyft=100
ARRAY _volbyat=100, _cntbyat=100, _lnkbyft=100, _lnkbyat=100
ARRAY _capbyft=100, _capbyat=100

_RGP[7]=50000,

if (ft<8)
    IF (A=1)
        LOOP _iter=1,13
            _err[_iter]=0, _cnt[_iter]=0, _cns[_iter]=0
        ENDLOOP
    ENDIF
endif

links=1
lanemiles=LNS*li.1.distance
; calculate and compartmentalize
if (TOTCNT>0)
    IF (TOTAL_VOL/TOTCNT, NETDIFF=TOTAL_VOL-TOTCNT, ABSDIFF=ABS(NETDIFF),
        ERRORSQ=NETDIFF^2, PCTDIFF=100*NETDIFF/TOTCNT

        _group=(0.0*FT)
        _RGP[7]=50000,
        IF (A=1)
            LOOP _iter=1,13
                _err[_iter]=0, _cnt[_iter]=0, _cns[_iter]=0
            ENDLOOP
        ENDIF

        _volbyft[ft]=TOTAL_VOL/TOTCNT
        _cntbyft[ft]=TOTCNT
        _lnkbyft[ft]=1
        _capbyft[ft]=CAP
        _volbyat[at]=TOTAL_VOL/TOTCNT
        _cntbyat[at]=TOTCNT
        _lnkbyat[at]=1
        _capbyat[at]=CAP
        _volbyft[100]=TOTAL_VOL
        _cntbyft[100]=TOTCNT
        _lnkbyft[100]=1
        _capbyft[100]=CAP
        _volbyat[100]=TOTAL_VOL
        _cntbyat[100]=TOTCNT
        _lnkbyat[100]=1
        _capbyat[100]=CAP

    endif
ENDIF

CROSSTAB VAR= LINKS LANEMILES, form=14.0c,
    col=FT, range=1-9-1,1-9,
    row=LANES, range=1-9-1,1-9

if (A=1)
    _volbyft[ft]=TOTAL_VOL
    _cntbyft[ft]=TOTCNT
    _lnkbyft[ft]=1
    _capbyft[ft]=CAP
    _volbyat[at]=TOTAL_VOL
    _cntbyat[at]=TOTCNT
    _lnkbyat[at]=1
    _capbyat[at]=CAP
ENDIF

CROSSTAB VAR= TOTAL_VOL, form=14.0c,
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

\_A\_VOL=TOTAL\_VOL
\_C\_VOL=TOTCNT
\_C\_CAP=CAP
CROSSTAB VAR= \_A\_VOL, \_C\_VOL, \_C\_VMT, \_A\_VMT, \_C\_VHT, \_A\_VHT, \_C\_CAP, form=14.0c,
col=FT, range=1-4-1,6-9-1,1-9,
row=AT, range=1-5-1,1-9,
comp=\_A\_VOL/_\_C\_VOL,form=8.3,
comp=\_A\_VMT/_\_C\_VMT,form=8.3,
comp=\_A\_VHT/_\_C\_VHT,form=8.3,
comp=\_A\_VOL/_\_C\_CAP,form=8.3
CROSSTAB VAR= \_A\_VOL, \_C\_VOL, form=14.0c,
col=FT, range=1-4-1,6-9-1,1-9,
row=LANES, range=1-9-1,1-6
;

CROSSTAB VAR= \_A\_VOL, \_C\_VOL, \_C\_VMT, \_A\_VMT, \_C\_VHT, \_A\_VHT, \_C\_CAP, form=14.0c,
; col=FT, range=1-4-1,6-9-1,1-9,
; row=LOCATION, range=1-7-1,1-7,
; comp=\_A\_VOL/_\_C\_VOL,form=8.3,
; comp=\_A\_VMT/_\_C\_VMT,form=8.3,
; comp=\_A\_VHT/_\_C\_VHT,form=8.3,
; comp=\_A\_VOL/_\_C\_CAP,form=8.3
ENDIF
IF (TOTCNT>0 & SCRNR<>0)
\_sVOL=TOTAL\_VOL
\_sCNT=TOTCNT
CROSSTAB VAR= \_sVOL, \_sCNT, form=9.0c,
col=FT, range=1-9,
row=SCRNR, range=1-20-1,1-20,
comp=\_sVOL/_\_sCNT,form=8.3
ENDIF
\_A\_VMT\_ALL=DISTANCE*TOTAL\_VOL
CROSSTAB VAR= \_A\_VMT\_ALL, form=14.0c,
col=FT, range=1-4-1,6-9-1,1-9,
row=AT, range=1-5-1,1-9
;

***************PERCENT ERROR BY VOLUME GROUPS CALCULATION***************

ARRAY \_PCT\_ERR\_VOLGRP=7
ARRAY \_LINKS\_VOLGRP=7

IF (_C\_VOL<>0)
IF (\_A\_VOL >= 1 & _A\_VOL< 5000)
\_LINKS\_VOLGRP[1]= \_LINKS\_VOLGRP[1] + 1
ENDIF

IF (\_A\_VOL >= 5000 & _A\_VOL< 10000)
ENDIF

IF (\_A\_VOL >= 10000 & _A\_VOL< 20000)
ENDIF

IF (\_A\_VOL >= 20000 & _A\_VOL< 30000)
ENDIF

IF (\_A\_VOL >= 30000 & _A\_VOL< 40000)
ENDIF

IF (\_A\_VOL >= 40000 & _A\_VOL< 50000)

ENDIF
ENDIF

IF (_A_VOL >= 1 & _A_VOL< 500000)
ENDIF

ENDIF

ENDPHASE

; =========================================================
; SUMMARY REPORTING
; =========================================================
PHASE=SUMMARY
PRINT LIST="Scenario = {SCENARIO_SHORTNAME}\n" PRINTO=1
;a little loop to write out the Percent Root Mean Square Error
print list="*************** ALL COUNT ROOT MEAN SQUARE ERROR SUMMARY (exclude HOV)
***************", printo=1
print list=" Group   Volume Range    % RMSE   Target %   Obs", PRINTO=1
LOOP _iter=1,12
  if (_iter=1) _limit='45 - 55'
  if (_iter=2) _limit='35 - 45'
  if (_iter=3) _limit='27 - 35'
  if (_iter=4) _limit='24 - 27'
  if (_iter=5) _limit='22 - 24'
  if (_iter=6) _limit='20 - 22'
  if (_iter=7) _limit='18 - 20'
  if (_iter=8) _limit='17 - 18'
  if (_iter=9) _limit='16 - 17'
  if (_iter=10) _limit='15 - 16'
  if (_iter=11) _limit='14 - 15'
  if (_iter=12) _limit='LT 14 '
  if (_cnt[_iter]>0)
    _RMSE=sqrt(_err[_iter]/(_cnt[_iter]-1))/(_cns[_iter]/_cnt[_iter])*100
    list=_iter(6.0c),"",_RGP[_iter](7.0c),"-",_RGP[_iter+1](7.0c),"",_RMSE(7.1),"%", " ",_limit," ",_cnt[_iter](5.0), PRINTO=1
  endif
ENDLOOP

_iter=13
_RMSE=sqrt(_err[_iter]/(_cnt[_iter]-1))/(_cns[_iter]/_cnt[_iter])*100
_limit='32 - 39'
list=_iter(6.0c),"",_RGP[1](7.0c),"-",_RGP[_iter](7.0c),"",_RMSE(7.1),"%", " ",_limit," ",_cnt[_iter](5.0), PRINTO=1

; one for Vol/Cnt by FT
_iter=0
LOOP _iter=1,100
  if (_iter=1) print list="\n","\n *************** VOLUME AND COUNT SUMMARY BY FACILITY
  _cntbyft[_iter]>0 print,
    list="Facility Type Summary for FT=",_iter(3.0c),
    " VOL=",_volbyft[_iter](11.0c),
    " CNT=",_cntbyft[_iter](11.0c),
    " VOL/CNT=",_volbyft[_iter]/_cntbyft[_iter](5.2c),
    " N=",_lnkbyft[_iter](5.0c), PRINTO=1
ENDLOOP

; one for Vol/Cnt by AT
_iter=0
LOOP _iter=1,100
  if (_iter=1) print list="\n","\n *************** VOLUME AND COUNT SUMMARY BY AREA
  _cntbyat[_iter]>0 print,
    " TYPE = " _iter(12.0c), PRINTO=1
list=" Area Type Summary for AT="_iter(3.0c)," VOL="_volbyat[_iter](1.0c)," CNT="_cntbyat[_iter](1.0c)," VOL/CNT="_volbyat[_iter]/_cntbyat[_iter](5.2c)," N="_linkbyat[_iter](5.0c), PRINTO=1 ENDOLOOP

_iter=0 LOOP _iter= 1,7
  if (_iter=0) print list="\n","n "*************** PERCENT ERROR SUMMARY BY VOLUME GROUP "***************", PRINTO=1 PERCENT_ERROR= _PCT_ERR_VOLGRP[_iter]/_LINKS_VOLGRP[_iter] PRINT,
  LIST=" VOLUME GROUP="_iter(3.0c),", " ," GROUP TOTAL PERCENT ERROR="_PCT_ERR_VOLGRP[_iter](13.0c)," PERCENT ERROR="_PCT_ERR_VOLGRP(_iter)(6.2c)," N="_LINKS_VOLGRP[_iter](11.0c), PRINTO=1 ENDOLOOP ENDPHASE ENDRUN

ASNET00D.S
; Do not change filenames or add or remove FILEI/FILEO statements using an editor. Use Cube/Application Manager.
RUN PGM=NETWORK PRNFILE="{SCENARIO_DIR}\output\HASSIGN2.PRN" MSG='Additional Highway Evaluation Scripts'
FILEO PRINTO[2] = "{SCENARIO_DIR}\output\OverallSummary.PRN"
FILEO PRINTO[1] = "{SCENARIO_DIR}\output\HEVAL_Daily2.PRN"
FILEI LINKI[1] = "{SCENARIO_DIR}\output\COMBINEDLOADED.NET"

;=========================================================================
; LINKMERGE BEGINS
;=========================================================================

PROCESS PHASE=LINKMERGE

;Calculate working link variables for highway analysis calculations
;==========================================================================
_FT2=ii.1.FTYPE ;2-digit Facility Type
_AT2=ii.1.ATYPE ;2-digit Area Type
_FT1=int(_FT2/10) ;1-digit Facility Type
_AT1=int(_AT2/10) ;1-digit Area Type
_LNS=ii.1.LANES ;Number of Lanes
_SL=ii.1.SCRN ;Screenline
_CNT=ii.1.COUNT\{year\} ;Traffic Count
_CAP=ii.1.CAPACITY*ii.1.LANES/ii.1.CONFAC ;Daily Directional Capacity
_VOL=BOUND(ii.1.MOTORIZEDVOL) ;Estimated Volume
LNKCNTR=1 ;Link Counter
if (_CNT>0) _LNK_w_CNT=1 ;Counter for links w/counts

_MLS=li.1.DISTANCE ;Directional System Miles
_LMLS=_LNS*_MLS ;Lane Miles
_CGT=li.1.CTIME ;Congested Time
_CGSP=li.1.CSSPEED ;Congested Speed
_SPD=li.1.SP ;Free Flow Speed
_WCGS=CGS*MLS ;Weighted Congested Speed
_WSPD=SPD*MLS ;Weighted Free Flow Speed
_VMT=VOL*MLS ;Vehicle Miles Traveled
_CVMT=COUNT*MLS ;Vehicle Miles Traveled using counts
_VHT=VOL*(CGT/60) ;Vehicle Hours Traveled
_CVHT=COUNT*(CGT/60) ;Vehicle Hours Traveled using counts
if (_CNT>0) _VCNT=VOL/_CNT ;Volume over Count
_VCAP=VOL_CAP ;Volume over Capacity

_LIND2=(AT2*10000)+(FT2*100)+(LNS) ;2-digit index

;initialize arrays and variables
ARRAY _err=13, _cns=13, _count=13, _RGP=13
ARRAY _volby=999999, _cntby=999999, _vcntby=999999, _volall=999999
ARRAY _lmiles=999999, _dmiles=999999, _wcgsp=999999, _wffsp=999999
ARRAY _volvmtval=999999, _cntvmtval=999999, _volvhtval=999999, _cntvhtval=999999
ARRAY _volall=999999, _vhtall=999999
ARRAY _slvol=99, _slcnt=99

_lnktot[_LIND2]=_lnktot[_LIND2]+_LNKCNTR
_lmiles[_LIND2]=_lmiles[_LIND2]+_LMLS
_dmiles[_LIND2]=_dmiles[_LIND2]+_MLS
_volall[_LIND2]=volall[_LIND2]+VOL
_wcgsp[_LIND2]=wcgsp[_LIND2]+WCGS
_wffsp[_LIND2]=wffsp[_LIND2]+WSPD
_vmtall[_LIND2]=vmtall[_LIND2]+VMT
_vhtall[_LIND2]=vhtall[_LIND2]+VHT

IF (_CNT>0) ;Variables only for link with counts
_volby[_LIND2]=volby[_LIND2]+VOL
_cntby[_LIND2]=cntby[_LIND2]+CNT
_vcntby[_LIND2]=vcntby[_LIND2]+VCNT
_lwctot[_LIND2]=lwctot[_LIND2]+_LNK_w_CNT
_volvmtval[_LIND2]=volvmtval[_LIND2]+VMT
_cntvmtval[_LIND2]=cntvmtval[_LIND2]+CVMT
_volvhtval[_LIND2]=volvhtval[_LIND2]+VHT
_cntvhtval[_LIND2]=cntvhtval[_LIND2]+CVHT
_slvol[_SL]=slvol[_SL]+VOL
_slcnt[_SL]=slcnt[_SL]+CNT
ENDIF

ENDPROCESS

;~~~~~~~~~~~~~~~~BEGIN REPORTING PORTION OF SCRIPT~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Facility Types:

- Generic Freeway:
- Urban Freeway Group 1:
- Other Freeway not in Group 1:
- Collector/Distributor Lanes:
- Controlled Access Expressway:
- Controlled Access Parkway:
- Generic Divided Arterial:
- Divided Arterial Unsignalized (55 mph):
- Divided Arterial Unsignalized (45 mph):
- Divided Arterial Class Ia:
- Divided Arterial Class Ib:
- Divided Arterial Class II/III:
- Generic Undivided Arterial:
- Undivided Arterial Unsignalized with Turn Bays:
- Undivided Arterial Class Ia with Turn Bays:
- Undivided Arterial Class Ib with Turn Bays:
- Undivided Arterial Class II/III with Turn Bays:
- Undivided Arterial Class Ia without Turn Bays:
- Undivided Arterial Class Ib without Turn Bays:
- Undivided Arterial Class II/III without Turn Bays:
- Generic Collector:
- Major Local Divided Roadway:
- Major Local Undivided Roadway with Turn Bays:
- Major Local Undivided Roadway without Turn Bays:
- Other Local Divided Roadway:
- Other Local Undivided Roadway with Turn Bays:
- Other Local Undivided Roadway without Turn Bays:
- Low Speed Local Collector:
- Very Low Speed Local Collector:
- Generic Centroid Connector:
- Basic Centroid Connector:
- External Station Centroid Connector:
- Generic One-Way:
- One-Way Facility Unsignalized:
- One-Way Facility Class Ia:
- One-Way Facility Class Ib:
- One-Way Facility Class II/III:
- Frontage Road Unsignalized:
- Frontage Road Class Ia:
- Frontage Road Class Ib:
- Frontage Road Class II/III:
- Generic Ramp:
- Freeway On-Ramp:
- Freeway Loop On-Ramp:
- Other On-Ramp:
- Other Loop On-Ramp:
- Freeway Off-Ramp:
- Freeway Loop Off-Ramp:
- Other Off-Ramp:
- Other Loop Off-Ramp:
- Freeway-Freeway High-Speed Ramp:

BEGIN NUMBER OF LINKS REPORT ------------ X = NUMBER OF LINKS

Print list=" ", PRINTO=1
Print list=" ", PRINTO=1
Print list="*******************************************************************************************
Print list="*                                Number of Directional Links (Centroid Connectors Excluded)                                    *

PRINTO=1

BEGIN NUMBER OF LINKS REPORT ------------ X = NUMBER OF LINKS

Print list=" ", PRINTO=1
Print list=" ", PRINTO=1
Print list="*******************************************************************************************
Print list="*                                Number of Directional Links (Centroid Connectors Excluded)                                    *

PRINTO=1
LOOP _a1iter=100000,599999,100000 ;^Begin Loop 1: Cycles through Area Types (ATYPE) by 10
   _aat1=int(_a1iter/100000) ; in order to get single digit ATYPE.
   print list= "Area Type ",_aat1(1.0),"x Range: ",
   "\n ", PRINTO=1
LOOP _a1iter=1,599999,1 ;^Begin Loop 2: Cycles through ATYPE by 1
   if (_a1iter>_a1iter+99999) BREAK ; in order to get two-digit ATYPE.
   _aat2=int(_a1iter/10000)
   _avcheck=0 ;^Initialize ATYPE X checking variable.
   LOOP _achkiter=_a1iter,599999,1 ;^Begin Loop 3: Cycles through Lanes and Facility Types (FTYPE)
      if (_achkiter>_a1iter+9999) BREAK ; for current ATYPE in Loop 2 and totals X checking variable.
      _avcheck=_avcheck+_lnktot[_achkiter]
   ENDLOOP
   if (_avcheck>0) ;^Begin Condition 1: If current ATYPE has X>0 continue to report X. Else skip ATYPE.
      _supertotal=0 ;^Initialize ATYPE total X.
      print list= "Area Type ",_aat2(2.0), PRINTO=1 ;^Header
      print list= "
                                                  Number of Lanes per Direction       
                                                  1           2           3           4           5           6
    7           8           9       Totals", PRINTO=1
      print list= "------------------------------------------------------------------------------
      Print list= "\n      LOOP _fiter=100,9900,100 ;^Begin Loop 4: Cycles through FTYPE by 1 in order to get two-digit FTYPE.
         _vcheck=0 ;^Initialize FTYPE X checking variable.
         LOOP _liter=1,9,1 ;^Begin Loop 5: Cycles through Lanes for current FTYPE in Loop 4 and totals X checking variable.
            _vcheck=_vcheck+_lnktot[_a1iter+_fiter+_liter]
         ENDLOOP
         if (_vcheck>0 & (_fiter<5000 | _fiter>5999)) ;^Begin Condition 2: If current FTYPE has X>0 continue to report X. Else skip FTYPE.
            _fft2=int(_fiter/100) ;^Initialize FTYPE total X.
            print list= _fft2(2.0)," ", PRINTO=1
            _totvols=0 ;^Initialize FTYPE total X.
            LOOP _liter2=1,9,1 ;^Begin Loop 6: Cycles through Lanes to generate ATYPE by FTYE by Lanes total X.
               print list="\", ",_lnktot[_a1iter+_fiter+_liter2](10.0C)," ", PRINTO=1
               _totvols=_totvols+_lnktot[_a1iter+_fiter+_liter2]
            _supertotal=_supertotal+_lnktot[_a1iter+_fiter+_liter2]
         ENDLOOP
      endif
   endif
ENDLOOP ;^End Loop 4.
print list= "Totals", PRINTO=1
  LOOP _liter3=1,9,1
    _lntotals=0
    LOOP _aiter2=_aiter,599999,100
      if (_aiter2>_aiter+9999) BREAK
      if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
        _lntotiter=_aiter2+_liter3
        _lntotals=_lntotals+_lnktot[_lntotiter]
      endif
    ENDLOOP
  print list="\"," ",_lntotals(10.0C)," ", PRINTO=1
ENDLOOP

print list="\",",_supertotal(10.0C), PRINTO=1
endif
ENDLOOP

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
print list= "Total Area Types ", PRINTO=1
print list= "                                    Number of Lanes per Direction
FType         1           2           3           4           5           6
            7           8           9       Totals", PRINTO=1
print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _fiter2=100,9900,100
  _fft2=int(_fiter2/100)
  _tafvcheck=0
  if (_fft2<50 | _fft2>59)
    LOOP _liter5=1,9,1
      LOOP _aiter4= 100000,599999,10000
        _tafvcheck=_tafvcheck+_lnktot[_aiter4+_fiter2+_liter5]
      ENDLOOP
    ENDLOOP
  endif
  if (_tafvcheck>0)
    LOOP _liter4= 1,9,1
      _totftat=0
      LOOP _aiter3= 100000,599999,10000
        _totftat=_totftat+_lnktot[_aiter3+_fiter2+_liter4]
      ENDLOOP
      print list= _fft2(2.0)," ", PRINTO=1
    ENDLOOP
  endif
ENDLOOP

LOOP _fiter2=100,9900,100
  _fft2=int(_fiter2/100)
  _tafvcheck=0
  if (_fft2<50 | _fft2>59)
    LOOP _liter5=1,9,1
      LOOP _aiter4= 100000,599999,10000
        _tafvcheck=_tafvcheck+_lnktot[_aiter4+_fiter2+_liter5]
      ENDLOOP
    ENDLOOP
  endif
ENDLOOP
totftat=totftat+lnktot[_aiter3+_filter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
ENDLOOP ;^End Loop 13.

print list="\\",",_totftat(10.0C),"", PRINTO=1
ENDLOOP ;^End Loop 12.

print list="\\",",_tafvcheck(10.0C), PRINTO=1
endif ;^End Condition 3.
endif
ENDLOOP ;^End Loop 9.

Print list= "------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
print list="Totals", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0 ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.
    _lntotals=0 ;^Initialize total X for Lanes.
    LOOP _aiter5=100000,599999,100 ;^Begin Loop 15: Cycles through ATYPE
and
    if {(_aiter5<105000 | _aiter5>105999) &
    (_aiter5<115000 | _aiter5>115999) &
    (_aiter5<125000 | _aiter5>125999) &
    (_aiter5<135000 | _aiter5>135999) &
    (_aiter5<145000 | _aiter5>145999) &
    (_aiter5<155000 | _aiter5>155999) &
    (_aiter5<165000 | _aiter5>165999) &
    (_aiter5<175000 | _aiter5>175999) &
    (_aiter5<185000 | _aiter5>185999) &
    (_aiter5<195000 | _aiter5>195999) &
    (_aiter5<205000 | _aiter5>205999) &
    (_aiter5<215000 | _aiter5>215999) &
    (_aiter5<225000 | _aiter5>225999) &
    (_aiter5<235000 | _aiter5>235999) &
    (_aiter5<245000 | _aiter5>245999) &
    (_aiter5<255000 | _aiter5>255999) &
    (_aiter5<265000 | _aiter5>265999) &
    (_aiter5<275000 | _aiter5>275999) &
    (_aiter5<285000 | _aiter5>285999) &
    (_aiter5<295000 | _aiter5>295999) &
    (_aiter5<305000 | _aiter5>305999) &
    (_aiter5<315000 | _aiter5>315999) &
    (_aiter5<325000 | _aiter5>325999) &
    (_aiter5<335000 | _aiter5>335999) &
    (_aiter5<345000 | _aiter5>345999) &
    (_aiter5<355000 | _aiter5>355999) &
    (_aiter5<365000 | _aiter5>365999) &
    (_aiter5<375000 | _aiter5>375999) &
    (_aiter5<385000 | _aiter5>385999) &
    (_aiter5<395000 | _aiter5>395999) &
    (_aiter5<405000 | _aiter5>405999) &
    (_aiter5<415000 | _aiter5>415999) &
    (_aiter5<425000 | _aiter5>425999) &
    (_aiter5<435000 | _aiter5>435999) &
    (_aiter5<445000 | _aiter5>445999) &
    (_aiter5<455000 | _aiter5>455999) &
    (_aiter5<465000 | _aiter5>465999) &
    (_aiter5<475000 | _aiter5>475999) &
    (_aiter5<485000 | _aiter5>485999) &
    (_aiter5<495000 | _aiter5>495999) &
    (_aiter5<505000 | _aiter5>505999) &
    (_aiter5<515000 | _aiter5>515999) &
    (_aiter5<525000 | _aiter5>525999) &
    (_aiter5<535000 | _aiter5>535999) &
    (_aiter5<545000 | _aiter5>545999) &
    (_aiter5<555000 | _aiter5>555999) &
    (_aiter5<565000 | _aiter5>565999) &

Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

{\_alter5<575000 | \_alter5>575999} &
{\_alter5<585000 | \_alter5>585999} &
{\_alter5<595000 | \_alter5>595999}

\_lntotiter=\_alter5+_\_liter6 ; FTYPE in order to generate total X for \_lntotals=\_lntotals+_lnktot(\_lntotiter) ; Lanes.
endfor

ENDLOOP

print list="\\", \_lntotals(10.0C), ",", PRINTO=1
\_supertotal=\_supertotal+_lntotals ; \^Generate total X for all ATYPE.
ENDLOOP

print list="\\", \_supertotal(10.0C), PRINTO=1

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------

Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;\^Header
Print list= "Single Digit Facility Types ", PRINTO=1
Print list= "AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
\------------------------------------------\nLOOP _a1iter2=100000,599999,100000 ;\^Begin Loop 16: Cycles through ATYPE by 10 to \_aat1=\_a1iter2/100000 ; get single digit ATYPE.
print list= _aat1(1.0),"x", ",", PRINTO=1
\_fttotal=0 ;\^Initialize total X for all ATYPE
LOOP _f1iter=1000,9900,100 ;\^Begin Loop 17: Cycles through FTYPE by 10 to \_totftlns=0 ;\^Initialize total X for all FTYPE by all Lanes.
if (_f1iter<5000 | _f1iter>5999) LOOP _fiter3=_f1iter,9900,100 ;\^Begin Loop 18: Cycles through two-digit FTYPE
if (_fiter3>_f1iter+999) BREAK ; for current single digit FTYPE in Loop 17.
LOOP _aiter6=_a1iter2,599999,10000 ;\^Begin Loop 19: Cycles through two-digit ATYPE
if (_aiter6>_a1iter2+9999) BREAK ; for current single digit ATYPE in Loop 16.
LOOP _liter7=1,9,1 ;\^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
\_totftlns=\_totftlns+_lnktot(_aiter6+_fiter3+_liter7) ; in order to generate total X for FTYPE by ATYPE.
ENDLOOP
ENDLOOP
ENDLOOP

ENDLOOP

ENDLOOP

print list="\\", \_totftlns(10.0C), ",", PRINTO=1
ENDLOOP

print list="\\", \_ftotal(10.0C), PRINTO=1
ENDLOOP

PENDLIST

Print list= "------------------------------------------------------------------------------------
\------------------------------------------\n
Page | 143
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update  
Technical Report No. 4: 2010 Model Update and Validation

print list="Totals", PRINTO=1

_supertotal=0                                           ;^Initialize overall total X.
LOOP _fiter2=1000,9900,1000                             ;^Begin Loop 21: Cycles through FTYPE
by 10
    _ftotals=0                                        ;^Initialize total X by FTYPE
    LOOP _fiter4=_fiter2,9900,100                      ;^Begin Loop 22: Cycles through FTYPE
    by 1 to
        if (_fiter4>_fiter2+999) BREAK
    FTYPE in
        if (_fiter2<5000 | _fiter2>5999)
    Loop 21.
        LOOP _liter8=1,9,1
            LOOP _aiter7=100000,599999,10000            ;^Begin Loop 24: Cycles through ATYPE in order
                _ftotiter=_aiter7+_fiter4+_liter8   ; to generate total X by single digit
                _ftotals=_ftotals+_lnktot[_ftotiter]  
            ENDLOOP
        ENDLOOP
        endif
    ENDLOOP
    _supertotal=_supertotal+_ftotals                  ;^Generate overall total for all single
digit ATYPE; by all single digit FTYPE.
print list=""," _ftotals(10.0C), PRINTO=1
ENDLOOP                                              ;^End Loop 21.
_numlinks=_supertotal
print list="", _supertotal(10.0C), PRINTO=1
print list=" ";************************************************************************
; END NUMBER OF LINKS REPORT
;************************************************************************
;======================================================================================
; BEGIN DIRECTIONAL MILES REPORT ----------- X = DIRECTIONAL MILES
;======================================================================================

print list=" "; BEGIN DIRECTIONAL MILES REPORT ------------ X = DIRECTIONAL MILES
print list="*******************************************************************************************
************************************* Directional Miles (Centroid Connectors Excluded) *
*******************************************************************************************
*************************************

LOOP _a1iter=100000,599999,100000                  ;^Begin Loop 1: Cycles through Area
by 10
    _aat1=int(_a1iter/100000)                       ; in order to get single digit ATYPE.
    print list="Area Type ",_aat1(1.0)," Range: ",
               "n ", PRINTO=1
    BEGIN DIRECTIONAL MILES TYPES BY 2-DIGIT AREA TYPES
LOOP _a1iter=100000,599999,10000                  ;^Begin Loop 2: Cycles through ATYPE by
1
    if (_a1iter>_a1iter+99999) BREAK
    _aat2=int(_a1iter/10000)
    _avcheck=0                                        ;^Initialize ATYPE X checking variable.
LOOP _achkiter=_alter,599999,1
and Facility Types (FTYPE)
    if (_achkiter>_alter+9999) BREAK
    _avcheck=_avcheck+_dmiles[_achkiter]
ENDLOOP

if (_avcheck>0)
in Loop 2
    _supertotal=0
    Print list= "Area Type ",_aat2(2.0), PRINTO=1
    Print list= "Number of Lanes per Direction FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
    Print list= "------------------------------------------------------------------------------
    ^Begin Loop 3: Cycles through Lanes and Facility Types (FTYPE)
    ^Begin Condition 1: If current ATYPE in Loop 2 has X>0 continue to report X. Else skip ATYPE.
    ^Initialize ATYPE total X.
    ^End Loop 3.

ENDLOOP

LOOP _fiter=100,9900,100
    _vcheck=0
    LOOP _liter=1,9,1
    _vcheck=_vcheck+_dmiles[_aiter+_fiter+_liter]
    ENDLOOP
    if (_vcheck>0 & (_fiter<5000 | _fiter>5999))
in Loop 4
    _fft2=int(_fiter/100)
    skip FTYPE.
    print list= _fft2(2.0),",", PRINTO=1
    _totvols=0
    LOOP _liter2=1,9,1
    _totvols=_totvols+_dmiles[_aiter+_fiter+_liter2]
    _supertotal=_supertotal+_dmiles[_aiter+_fiter+_liter2]
    ENDLOOP
    print list=_totvols(10.2C), PRINTO=1
    endif
    ^Begin Loop 5: Cycles through Lanes for current FTYPE in Loop 4 and totals X
    ^Initialize FTYPE total X.
    ^End Loop 5.

ENDLOOP

Print list= "Totals", PRINTO=1
for current ATYPE in Loop 2
    _lntotals=0
    LOOP _aiter2=_alter,599999,100
    _lntotals=0
    LOOP _alter2=_alter,599999,100
    _lntotals=0
    LOOP _alter2=_alter,599999,100
    if (_alter2>_alter+9999) BREAK
    if (_alter2<_alter+5000 | _alter2>_alter+5999)
    _lntotiter=_aiter2+_liter3
    _lntotals=_lntotals+_dmiles[_lntotiter]
    endif
    ENDLOOP
    _lntotals=_lntotals+_dmiles[_lntotiter]
    ENDLOOP
    print list="\"," ",_lntotals(10.2C), PRINTO=1
    endif
    ^End Loop 6.

ENDLOOP

^End Loop 4.

Print list= "Totals", PRINTO=1
for current ATYPE in Loop 2
    _lntotals=0
    LOOP _aiter2=_alter,599999,100
    _lntotals=0
    LOOP _alter2=_alter+9999)
    ENDLOOP
    print list="\"," ",_lntotals(10.2C), PRINTO=1
    endif
    ^End Loop 7.

ENDLOOP
<table>
<thead>
<tr>
<th>FType</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:**

- The table above represents the 2-digit facility types by total area types.
- The columns indicate the number of lanes per direction for each facility type.
- The rows signify the total area types.
- The totals column sums up the values for each lane type.

---

**Additional Code Snippet:**

```plaintext
print list="\", PRINTO=1
endif
ENDLOOP

print list="", PRINTO=1
ENDLOOP

print list="", PRINTO=1
ENDLOOP

print list="", PRINTO=1
ENDLOOP
```

---

**Further Explanation:**

- The code initializes variables for total area types and prints them out in a tabular format.
- It then cycles through different facility types and calculates the total area types by summing up the number of lanes per direction.
- Conditions are used to ensure that only non-zero values are printed in the totals column.

---

**Technical Details:**

- This technical report focuses on the model update and validation for the Gainesville Urbanized Area Year 2040 Long Range Transportation Plan.
- The code snippet provided is part of the computational methods used for the plan update.
- The table and code work together to provide a comprehensive view of the transportation network planning.
LOOP _aiter5=100000,599999,100 ;^Begin Loop 15: Cycles through ATYPE
and
if ((_aiter5<105000 | _aiter5>105999) &
(_aiter5<115000 | _aiter5>115999) &
(_aiter5<125000 | _aiter5>125999) &
(_aiter5<135000 | _aiter5>135999) &
(_aiter5<145000 | _aiter5>145999) &
(_aiter5<155000 | _aiter5>155999) &
(_aiter5<165000 | _aiter5>165999) &
(_aiter5<175000 | _aiter5>175999) &
(_aiter5<185000 | _aiter5>185999) &
(_aiter5<195000 | _aiter5>195999) &
(_aiter5<205000 | _aiter5>205999) &
(_aiter5<215000 | _aiter5>215999) &
(_aiter5<225000 | _aiter5>225999) &
(_aiter5<235000 | _aiter5>235999) &
(_aiter5<245000 | _aiter5>245999) &
(_aiter5<255000 | _aiter5>255999) &
(_aiter5<265000 | _aiter5>265999) &
(_aiter5<275000 | _aiter5>275999) &
(_aiter5<285000 | _aiter5>285999) &
(_aiter5<295000 | _aiter5>295999) &
(_aiter5<305000 | _aiter5>305999) &
(_aiter5<315000 | _aiter5>315999) &
(_aiter5<325000 | _aiter5>325999) &
(_aiter5<335000 | _aiter5>335999) &
(_aiter5<345000 | _aiter5>345999) &
(_aiter5<355000 | _aiter5>355999) &
(_aiter5<365000 | _aiter5>365999) &
(_aiter5<375000 | _aiter5>375999) &
(_aiter5<385000 | _aiter5>385999) &
(_aiter5<395000 | _aiter5>395999) &
(_aiter5<405000 | _aiter5>405999) &
(_aiter5<415000 | _aiter5>415999) &
(_aiter5<425000 | _aiter5>425999) &
(_aiter5<435000 | _aiter5>435999) &
(_aiter5<445000 | _aiter5>445999) &
(_aiter5<455000 | _aiter5>455999) &
(_aiter5<465000 | _aiter5>465999) &
(_aiter5<475000 | _aiter5>475999) &
(_aiter5<485000 | _aiter5>485999) &
(_aiter5<495000 | _aiter5>495999) &
(_aiter5<505000 | _aiter5>505999) &
(_aiter5<515000 | _aiter5>515999) &
(_aiter5<525000 | _aiter5>525999) &
(_aiter5<535000 | _aiter5>535999) &
(_aiter5<545000 | _aiter5>545999) &
(_aiter5<555000 | _aiter5>555999) &
(_aiter5<565000 | _aiter5>565999) &
(_aiter5<575000 | _aiter5>575999) &
(_aiter5<585000 | _aiter5>585999) &
(_aiter5<595000 | _aiter5>595999))

_for
_intotiter=_aiter5+_liter6 ; FTYPE in order to generate total X
_for
_intotals=_intotals+_dmiles[_intotiter] ; Lanes.
_endif
ENDLOOP ;^End Loop 15.

print list="\"","_intotals(10.2C),"", PRINTO=1
_supertotal=_supertotal+_intotals ;^Generate total X for all ATYPE.
ENDLOOP ;^End Loop 14.
print list="\"","_supertotal(10.2C), PRINTO=1
print list="\n"," PRINTO=1
;
------------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------

Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;"Header
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

Print list= "                                  Single Digit Facility Types
", PRINTO=1
Print list= "AType         1x          2x          3x          4x          5x          6x
7x          8x          9x      Totals", PRINTO=1
Print list= "------------------------------------------", PRINTO=1
LOOP _a1iter2=100000,599999,100000             ;^Begin Loop 16: Cycles through ATYPE
by 10 to
   _aat1=int(_a1iter2/100000)                  ; get single digit ATYPE.
   print list= _aat1(1.0),"x","    ", PRINTO=1
   _fttotal=0                                  ;^Initialize total X for all ATYPE
   LOOP _f1iter=1000,9900,1000                 ;^Begin Loop 17: Cycles through FTYPE
      by 10 to
         _totftlns=0                           ;^Initialize total X for all FTYPE by
all Lanes.
         if (_f1iter<5000 | _f1iter>5999)      ;^Begin Loop 18: Cycles through two-
digit FTYPE
            LOOP _fiter3=_f1iter,9900,100      ;^Begin Loop 19: Cycles through two-
digit ATYPE
            if (_fiter3>_f1iter+999) BREAK     ; for current single digit FTYPE in
            LOOP _aiter6=_a1iter2,599999,10000   ;^Begin Loop 20: Cycles
               by 10 to
                  _totftlns=_totftlns+_dmiles[_aiter6+_fiter3+_liter7]   ; in order to generate total
X for FTYPE by ATYPE.
            ENDLOOP                                 ;^End Loop 20.
            ENDLOOP                                 ;^End Loop 19.
         endif
         _fttotal=_fttotal+_totftlns              ;^Generate total X for ATYPE.
      ENDLOOP                                    ;^End Loop 17.
   ENDLOOP                                      ;^End Loop 16.
END LOOP                                        ;^End Loop 15.
Print list= "------------------------------------------", PRINTO=1
Print list= "Totals", PRINTO=1
_supertotal=0                                    ;^Initialize overall total X.
LOOP _f1iter2=1000,9900,1000                   ;^Begin Loop 21: Cycles through FTYPE
by 10
   _ftotals=0                                   ;^Initialize total X by FTYPE
   LOOP _fiter4=_f1iter2,9900,100               ;^Begin Loop 22: Cycles through FTYPE
      by 1 to
         if (_fiter4>_f1iter2+999) BREAK       ; to get single digit FTYPE.
         LOOP _liter8=1,9,1                      ;^Begin Loop 23: Cycles through Lanes.
            LOOP _a1iter8=100000,599999,10000    ;^Begin Loop 24: Cycles through ATYPE
               in order
                  _ftotiter=_a1iter8+_fiter4+_liter8; to generate total X by single digit
FTYPE.
_ftotals=_ftotals+_dmiles[_ftotiter]  
ENDLOOP  
|ENDLOOP|  
endif  
ENDLOOP  
_supertotal=_supertotal+_ftotals  
;^Generate overall total for all single  
digit ATYPE  
print list="\\", ",_ftotals(10.2C)," ", PRINTO=1  
ENDLOOP  
_dirmiles=_supertotal  
print list="\\", ",_supertotal(10.2C), PRINTO=1  
print list=" ", PRINTO=1  
;************************************************************************  
; END DIRECTIONAL MILES REPORT  
;************************************************************************  
;======================================================================================  
; BEGIN LANE MILES REPORT ------------- X = LANE MILES  
;======================================================================================  
Print list=" ", PRINTO=1  
Print list="*******************************************************************************************  
*************************************", PRINTO=1  
Print list="*
*
L
ane Miles (Centroid Connectors Excluded)
*
*
*******************************************************************************************", PRINTO=1  
Print list=" ", PRINTO=1  
;-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES--------------------  
LOOP _a1iter=100000,599999,10000  
;^Begin Loop 1: Cycles through Area  
Types (ATYPE) by 10  
_aat1=int(_a1iter/100000)  
print list= "Area Type ",_aat1(1.0),"x Range:

", PRINTO=1  
END LOOP  
if (_a1iter>_a1iter+99999) BREAK  
_LOOP =_a1iter=100000,599999,10000  
;^Begin Loop 2: Cycles through ATYPE by  
in order to get single digit ATYPE.  
if (_a1iter>_a1iter+9999) BREAK  
_aat2=int(_a1iter/10000)  
LOOP _achkiter=_a1iter,599999,1  
;^Begin Loop 3: Cycles through Lanes  
and Facility Types (FTYPE)  
if (_achkiter>_a1iter+9999) BREAK  
_totals X checking variable.  
_avcheck=_avcheck+_lmiles[_achkiter]  
END LOOP  
if (_avcheck>0)  
;^Begin Condition 1: If current ATYPE  
has X>0 continue to report X. Else  
_skip ATYPE.  
_LOOP =_supertotal=0  
;^Initialize ATYPE total X.  
print list= "Area Type ",_aat2(2.0), PRINTO=1  
print list= " ", PRINTO=1  
;--------------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES----------------------  
print list="Direction ", PRINTO=1  
print list="FType ", PRINTO=1  
print list="1" 
print list="2" 
print list="3" 
print list="4" 
print list="5" 
print list="6" 
print list="7" 
print list="8" 
print list="9" 
print list="Totals", PRINTO=1  
print list="-----------------------------------", PRINTO=1
LOOP _fiter=100,9900,100
  _vcheck=0
  LOOP _liter=1,9,1
    for current
      _vcheck=_vcheck+_lmiles[_aiter+_fiter+_liter]
    ENDLOOP
    if (_vcheck>0 & (_fiter<5000 | _fiter>5999))
      _fft2=int(_fiter/100)
    endif
    print list= _fft2(2.0),"    ", PRINTO=1
    _totvols=0
    LOOP _liter2=1,9,1
      print list="\",_lmiles[_aiter+_fiter+_liter2](10.2C)," ", PRINTO=1
      _totvols=_totvols+_lmiles[_aiter+_fiter+_liter2]
    ENDLOOP
    print list="\",_totvols(10.2C), PRINTO=1
  ENDLOOP
ENDLOOP

Print list= "------------------------------------------------------------------------------
------------------------------------------------ 
Totals"
LOOP _liter3=1,9,1
  _lntotals=0
  LOOP _aiter2=_aiter,599999,100
    for current ATYPE in Loop 2.
    if (_aiter2>_aiter+9999) BREAK
    if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
      _lntotiter=_aiter2+_liter3
    endif
    _lntotals=_lntotals+_lmiles[_lntotiter]
  ENDLOOP
  print list="\",_lntotals(10.2C)," ", PRINTO=1
ENDLOOP
print list="\",_supertotal(10.2C), PRINTO=1
print list=" ", PRINTO=1
endif
ENDLOOP

Print list= "2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES ------------------------------- 
Total Area Types 
Number of Lanes per Direction 
FType     1     2     3     4     5     6 
7     8     9     Totals"
LOOP _fiter2=100,9900,100
  "Begin Loop 9: Cycles through FTYPES to get

Page | 150
_fft2=int(_fiter2/100) ; two-digit FTYPE.
_tafvcheck=0 ;^Initialize FTYPE X checking variable.
if (_fft2<50 | _fft2>59)
  LOOP _liter5=1,9,1
  for current
    LOOP _aiter4= 100000,599999,10000
      for
        _tafvcheck=_tafvcheck+_lmiles[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
      ENDLOOP
    ENDLOOP
  ENDLOOP
endif ;^End Condition 3.
endif
ENDLOOP ;^End Loop 9.

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
_supertotal=0 ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1
  _lntotals=0 ;^Initialize total X for Lanes.
  LOOP _aiter5=100000,599999,100
    LOOP _aiter3= 100000,599999,10000
      for current Lanes in Loop 12
      _totftat=_totftat+_lmiles[_aiter3+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
    ENDLOOP
    print list="\",",_totftat(10.2C)," ", PRINTO=1
  ENDLOOP
  if ((_aiter5<105000 | _aiter5>105999) &
    (_aiter5<115000 | _aiter5>115999) &
    (_aiter5<125000 | _aiter5>125999) &
    (_aiter5<135000 | _aiter5>135999) &
    (_aiter5<145000 | _aiter5>145999) &
    (_aiter5<155000 | _aiter5>155999) &
    (_aiter5<165000 | _aiter5>165999) &
    (_aiter5<175000 | _aiter5>175999) &
    (_aiter5<185000 | _aiter5>185999) &
    (_aiter5<195000 | _aiter5>195999) &
    (_aiter5<205000 | _aiter5>205999) &
    (_aiter5<215000 | _aiter5>215999) &
    (_aiter5<225000 | _aiter5>225999) &
    (_aiter5<235000 | _aiter5>235999) &
    (_aiter5<245000 | _aiter5>245999) &
    (_aiter5<255000 | _aiter5>255999) &
    (_aiter5<265000 | _aiter5>265999) &
    (_aiter5<275000 | _aiter5>275999) &
    (_aiter5<285000 | _aiter5>285999) &
    (_aiter5>295000 | _aiter5<305000)) &
    (_aiter5>305000 | _aiter5<315000) &
    (_aiter5>315000 | _aiter5<325000) &
    (_aiter5>325000 | _aiter5<335000) &
    (_aiter5>335000 | _aiter5<345000) &
    (_aiter5>345000 | _aiter5<355000) &
    (_aiter5>355000 | _aiter5<365000) &
    (_aiter5>365000 & _aiter5<375000) &
    (_aiter5>375000 & _aiter5<385000) &
    (_aiter5>385000 & _aiter5<395000) &
    (_aiter5>395000 & _aiter5<405000) &
    (_aiter5>405000 & _aiter5<415000) &
    (_aiter5>415000 & _aiter5<425000) &
    (_aiter5>425000 & _aiter5<435000) &
    (_aiter5>435000 & _aiter5<445000) &
    (_aiter5>445000 & _aiter5<455000) &
    (_aiter5>455000 & _aiter5<465000) &
    (_aiter5>465000 & _aiter5<475000) &
    (_aiter5>475000 & _aiter5<485000) &
    (_aiter5>485000 & _aiter5<495000) &
    (_aiter5>495000 & _aiter5<505000) &
    (_aiter5>505000 & _aiter5<515000) &
    (_aiter5>515000 & _aiter5<525000) &
    (_aiter5>525000 & _aiter5<535000) &
    (_aiter5>535000 & _aiter5<545000) &
    (_aiter5>545000) &
  ENDLOOP
  print list="\",",_lntotals(10.2C)," ", PRINTO=1
endif ;^End Condition 3.
ENDLOOP ;^End Loop 9.
_lntotiter=_alter5+_liter6
for
_lntotals=_lntotals+_lmiles[_lntotiter]; FTYPE in order to generate total X

endif
ENDLOOP

print list="\",",_lntotals(10.2C),"", PRINTO=1
_supertotal=_supertotal+_lntotals;^Generate total X for all ATYPE.
ENDLOOP

print list="\",",_supertotal(10.2C), PRINTO=1
print list=" "," 

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= "
Single Digit Facility Types ", PRINTO=1
Print list= "AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO=1
Print list= "-----------------------------------------------------------------------------", PRINTO=1
LOOP _alter2=100000,599999,100000 ;^Begin Loop 16: Cycles through ATYPE by 10
_alat1=int(_alter2/100000); get single digit ATYPE.
print list= _alat1(1.0),"x", PRINTO=1
_fttotal=0
LOOP _filter=1000,9900,1000 ;^Begin Loop 17: Cycles through FTYPE by 10
_totftlns=0
all Lanes.
if (_filter<5000 | _filter>59999)
LOOP _filter3=_filter,9900,1000 ;^Begin Loop 18: Cycles through two-
if (_fiter3>_f1iter+999) BREAK ; for current single digit FTYPE in Loop 17.

LOOP _aiter6=_a1iter2,599999,10000 ;^Begin Loop 19: Cycles through two-digit ATYPE
if (_aiter6>_a1iter2+99999) BREAK ; for current single digit ATYPE in Loop 16.

LOOP _liter7=1,9,1 ;^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_lmiles[_aiter6+_fiter3+_liter7] ; in order to generate total X for FTYPE by ATYPE.
ENDLOOP ;^End Loop 20.
ENDLOOP
ENDLOOP
ENDLOOP
endif
_fttotal=_fttotal+_totftlns ;^Generate total X for ATYPE.

print list="\"," ",_totftlns(10.2C)," ", PRINTO=1
ENDLOOP ;^End Loop 17.

print list="\"," ",_fttotal(10.2c), PRINTO=1
ENDLOOP ;^End Loop 16.

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list= "Totals", PRINTO=1
_supertotal=0 ;^Initialize overall total X.
LOOP _f1iter2=1000,9900,1000 ;^Begin Loop 21: Cycles through FTYPE by 10
by 10
_ftotals=0
LOOP _fiter4=_f1iter2,9900,100 ;^Begin Loop 22: Cycles through FTYPE by 1 to 100
if (_fiter4>_f1iter2+999) BREAK ; get all two-digit FTYPE for current FTYPE in Loop 21.
if (_f1iter2<5000 | _f1iter2>5999) ;
LOOP _liter8=1,9,1 ;^Begin Loop 23: Cycles through Lanes.
LOOP _aiter7=100000,599999,10000 ;^Begin Loop 24: Cycles through ATYPE in order
_ftotiter=_aiter7+_fiter4+_liter8 ; to generate total X by single digit FTYPE.
_ftotals=_ftotals+_lmiles[_ftotiter]
ENDLOOP ;^End Loop 24.
ENDLOOP
endif
ENDLOOP ;^End Loop 22.
_supertotal=_supertotal+_ftotals ;^Generate overall total for all single digit ATYPE.

print list="\"," ",_ftotals(10.2C)," ", PRINTO=1
ENDLOOP ;^End Loop 21.

_printlist="_lanemiles=_supertotal
print list="\"," ",_supertotal(10.2C), PRINTO=1
print list=" ", PRINTO=1
************************************************************************
END LANE MILES REPORT
************************************************************************
======================================================================================
BEGIN VMT VAL REPORT ------------- X = VMT on Links w/ Counts
======================================================================================
Print list= " 
END LANE MILES REPORT

; BEGIN VMT VAL REPORT --------------- X = VMT on Links w/ Counts
;-------------------------------------------------------------------------------
Print list=" ", PRINTO=1
null
ENDLOOP ;^End Loop 6.

print list="\"",_totvols(10.0C), PRINTO=1
endif ;^End Condition 2.
ENDLOOP ;^End Loop 4.

Print list= "------------------------------------------------------------------------------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

LOOP _liter3=1,9,1 ;^Begin Loop 7: Cycles through Lanes
for
    _lntotals=0 ;^Initialize Lane total X.
    LOOP _aiter2=_aiter,599999,100 ;^Begin Loop 8: Cycles through FTYPE for current ATYPE in Loop 2.
        if (_aiter2>_aiter+9999) BREAK ; in Loop 2 to generate Lane total X.
        _lntotiter=_aiter2+_liter3
        _lntotals=_lntotals+_volvmtval[_lntotiter]
    ENDLOOP ;^End Loop 8
    print list="\"",_lntotals(10.0C), PRINTO=1
ENDLOOP ;^End Loop 7

print list="\",_supertotal(10.0C), PRINTO=1
print list="\", PRINTO=1
endif ;^End Condition 1.
ENDLOOP ;^End Loop 2.
print list="\", PRINTO=1
ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES---------------------
Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= "                                                  Number of Lanes per Direction
Number of Lanes per Direction

FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100 ;^Begin Loop 9: Cycles through FTYPES to get two-digit FTYPE.
    _fft2=int(_fiter2/100) ;^Initialize FTYPE X checking variable.
    _tafvcheck=0 ;^Initialize FTYPE X checking variable.
    LOOP _liter5=1,9,1 ;^Begin Loop 10: Cycles through Lanes for current FTYPE in Loop 9.
        LOOP _aiter4= 100000,599999,10000 ;^Begin Loop 11: Cycles through ATYPE for _tafvcheck= _tafvcheck+ _volvmtval[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
            _tafvcheck= _tafvcheck+ _volvmtval[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
        ENDLOOP ;^End Loop 11.
    ENDLOOP ;^End Loop 10.

if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE has X>0 continue to report X. Else skip FTYPE.
    LOOP _liter4= 1,9,1 ;^Begin Loop 12: Cycles through Lanes in Loop 9.
        for current FTYPE

_totftat=0 ;^Initialize FTYPE total X for all
ATYPE.

LOOP _aiter3= 100000,599999,10000 ;^Begin Loop 13: Cycles through ATYPE
   for current Lanes in Loop 12
   _totftat=_totftat+_volvmtval[_aiter3+_fiter2+_liter4] ; in order to generate total X for
   FTYPE by Lane for all ATYPE.
   ENDLOOP ;^End Loop 13.

   print list="\\"," ",_totftat(10.0C)," ", PRINTO=1
   ENDLOOP ;^End Loop 12.

endif ;^End Condition 3.

ENDLOOP ;^End Loop 9.

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
Print list= "Totals", PRINTO=1
_supertotal=0 ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.
   _lntotals=0 ;^Initialize total X for Lanes.
   LOOP _aiter5=100000,599999,100 ;^Begin Loop 15: Cycles through ATYPE
      and
      _lntotiter=_aiter5+_liter6 ; FTYPE in order to generate total X
      _lntotals=_lntotals+_volvmtval[_lntotiter] ; Lanes.
   ENDLOOP ;^End Loop 15.
   print list="\\"," ",_lntotals(10.0C)," ", PRINTO=1
   _supertotal=_supertotal+_lntotals ;^Generate total X for all ATYPE.
   ENDLOOP ;^End Loop 14.
   print list="\\"," ",_supertotal(10.0C), PRINTO=1
   print list=" 
   ;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= " Single Digit Facility Types ", PRINTO=1 ;^Header
Print list= "AType 
   1x 
   2x 
   3x 
   4x 
   5x 
   6x 
   7x 
   Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
Print list= "Totals", PRINTO=1

LOOP _a1iter2=100000,599999,100000 ;^Begin Loop 16: Cycles through ATYPE
   by 10 to
   _aat1=int(_a1iter2/100000) ; get single digit ATYPE.
   _fttotal=0 ;^Initialize total X for all ATYPE
   LOOP _f1iter=1000,9900,1000 ;^Begin Loop 17: Cycles through FTYPE
      by 10 to
      _totftlins=0 ;^Initialize total X for all FTYPE by all Lanes.
      LOOP _fiter3=_f1iter,9900,100 ;^Begin Loop 18: Cycles through two-
digit FTYPE
         if (_fiter3>_f1iter+999) BREAK ; for current single digit FTYPE in
Loop 17.
      _loop _aiter6=_aliter2,599999,100000 ;^Begin Loop 19: Cycles through two-
digit ATYPE
if (_aiter6>_a1iter2+99999) BREAK ; for current single digit ATYPE in
Loop 16.

LOOP _liter7=1,9,1 ;^Begin Loop 20: Cycles
through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_volvmval[_aiter6+_fiter3+_liter7] ; in order to generate
total X for FTYPE by ATYPE.
ENDLOOP ;^End Loop 20.

ENDLOOP

_LOOP _fiter4=_f1iter2,9900,100 ;^Begin Loop 21: Cycles through FTYPE
by 10 ; to get single digit FTYPE.
_FTOTALS=_FTOTALS+_VOLVMVAT[ _FTOTITER ]
ENDLOOP ;^End Loop 21.

ENDLOOP

_LOOP _liter8=1,9,1
_LOOP _aiter7=1000,59999,1000 ;^Begin Loop 24: Cycles through ATYPE
in order
_FTOTALS=_FTOTALS+_VOLVMVAT[ _FTOTITER ]
ENDLOOP

ENDLOOP

_SUPERTOTAL=_SUPERTOTAL+_FTOTALS ;^Generate overall total for all single
digit ATYPE

_END LOOP 16.

_END LOOP 17.

_END LOOP 18.

_END LOOP 19.

_END LOOP 20.

_END LOOP

_SUPERTOTAL=0

_LOOP _F1ITER2=1000,9900,1000
by 10
_FTOTALS=0

_LOOP _FITER4=_F1ITER2,9900,100
by 1
if (_FITER4>_F1ITER2+999) BREAK ; get all two-digit FTYPE for current
FTYPE in
_LOOP _LITER8=1,9,1
_LOOP _AITER7=100000,599999,10000 ;^Begin Loop 23: Cycles through Lanes.
in order
_FTOTITER=_AITER7+_FITER4+_LITER8
_FTOTALS=_FTOTALS+_VOLVMVT[ _FTOTITER ]
ENDLOOP

_END LOOP

_END LOOP 22.

_END LOOP 23.

_SUPERTOTAL=_SUPERTOTAL+_FTOTALS ;^Generate overall total for all single
digit ATYPE

_END LOOP

_END LOOP

_VMTVOLCounts=_SUPERTOTAL
print list="\",",_SUPERTOTAL(10.0C)," ", PRINTO=1
print list="", PRINTO=1

*************************************************************************
*************************************************************************
======================================================================================
BEGIN VMT Count REPORT ------------- X = Count VMT on Links w/ Counts
======================================================================================

Print list="", PRINTO=1
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="*", PRINTO=1
Print list="*                              Vehicle Miles Traveled (VMT) using Counts on Links
with Counts *", PRINTO=1
Print list="*\", PRINTO=1
Print
list="*******************************************************************************************
***********************************************************", PRINTO=1
Print list="                        2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES----------------------
LOOP _aliter=100000,599999,100000                        ;^Begin Loop 1: Cycles through Area
Types (ATYPE) by 10
   _aat1=int(_aliter/100000)                               ; in order to get single digit ATYPE.
   print list= "Area Type ",_aat1(1.0),"x Range:\n",
   \n \", PRINTO=1
   LOOP _aiter=_aliter,599999,10000                        ;^Begin Loop 2: Cycles through ATYPE by
   1
   if (_aiter>_aliter+99999) BREAK                       ; in order to get two-digit ATYPE.
   _avcheck=0                                            ;^Initialize ATYPE X checking variable.
   LOOP _achkiter=_aiter,599999,1                        ;^Begin Loop 3: Cycles through Lanes
   and Facility Types (FTYPE)
   if (_achkiter>_aiter+9999) BREAK                    ; for current ATYPE in Loop 2 and
totals X checking variable.
      _avcheck=_avcheck+_cntvmtval[_achkiter]
   ENDLOOP                                               ;^End Loop 3.
   if (_avcheck>0)                                       ;^Begin Condition 1: If current ATYPE
   ; has X>0 continue to report X. Else
   _supertotal=0                                       ;^Initialize ATYPE total X.
   Print list= "Area Type ",_aat2(2.0), PRINTO=1       ;^Header
   Print list= "                                                  Number of Lanes per
   Direction                                               
   FType         1           2           3           4           5           6 
   Totals", PRINTO=1
   Print list= "------------------------------------------------------------------------------
   ------------------------------------------------", PRINTO=1
   LOOP _fiter=100,9900,100                            ;^Begin Loop 4: Cycles through FTYPE
   ; by 1 in order to get two-digit FTYPE.
   _vcheck=0                                          ;^Initialize FTYPE X checking variable.
   LOOP _liter=1,9,1                                  ;^Begin Loop 5: Cycles through Lanes
   for current
   _vcheck=_vcheck+_cntvmtval[_aiter+_fiter+_liter]    ; FTYPE in Loop 4 and totals X
checking variable.
   ENDLOOP                                            ;^End Loop 5.
   if (_vcheck>0)                                    ;^Begin Condition 2: If current FTYPE
   ; has X>0 continue to report X. Else
   _fft2=int(_fiter/100)                          ;^Initialize FTYPE total X.
   _totvols=0                                     ;^Initialize FTYPE total X.
   LOOP _liter2=1,9,1                             ;^Begin Loop 6: Cycles through Lanes to
generate ATYPE by FTYPE by Lanes total X.
   print list=_fft2(2.0),"\", PRINTO=1
   _totvols=0
   LOOP _liter2=1,9,1
   _supertotal=_supertotal+_cntvmtval[_aiter+_fiter+_liter2] 
   ENDLOOP                                        ;^End Loop 6.
   endif                                             ;^End Condition 2.
ENDLOOP ; End Loop 4.

Print list="------------------------------------------------------------------------------
------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

LOOP _liter3=1,9,1

_endtotals=0

LOOP _aiter2=_aiter,599999,100
for current ATYPE in Loop 2.
if (_aiter2>_aiter+9999) BREAK
_endtotiter=_aiter2+_liter3
_endtotals=_endtotals+_cntvmtval[_endtotiter]
ENDLOOP
print list="\", "_endtotals(10.0C), "", PRINTO=1
ENDLOOP ; End Loop 7

print list="\", "_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
endif ; End Condition 1.
ENDLOOP ; End Loop 2.
ENDLOOP ; End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES-------------------
Print list= "Total Area Types ", PRINTO=1 ; Header
Print list= "Number of Lanes per Direction ", PRINTO=1
Print list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100
to get
_ff2=int(_fiter2/100)
_tafvcheck=0

LOOP _liter5=1,9,1
for current

LOOP _aiter4= 100000,599999,10000
for current ATYPE

_tafvcheck=_tafvcheck+_cntvmtval[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in
order to total X checking variable.
ENDLOOP
ENDLOOP

if (_tafvcheck>0)
in Loop 9
print list= _ff2(2.0),", "", PRINTO=1
skip FTYPE.

LOOP _liter4= 1,9,1
for current FTYPE
_totftat=0

LOOP _aiter3= 100000,599999,10000
for current Lanes in Loop 12
_totftat =_totftat+_cntvmtval[|_aiter3+_fiter2+_liter4|] ; in order to generate total X for FTYPE by Lane for all ATYPE.
ENDLOOP ;^End Loop 13.

print list="\"," _totftat(10.0C)," ", PRINTO=1
ENDLOOP ;^End Loop 12.

print list="\"," _tafvcheck(10.0C), PRINTO=1
endif ;^End Condition 3.

ENDLOOP ;^End Loop 9.

Print list= "---------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
print list="Totals", PRINTO=1

_supertotal=0 ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.

_intotals=0 ;^Initialize total X for Lanes.
LOOP _aiter5=100000,599999,100 and 
_intotiter=_aiter5+_liter6
for 
_intotals=_intotals+_cntvmtval[_intotiter] ; Lanes.
ENDLOOP ;^End Loop 15.

print list="\"," _intotals(10.0C)," ", PRINTO=1
_supertotal=_supertotal+_intotals ;^Generate total X for all ATYPE.
ENDLOOP ;^End Loop 14.
print list="\"," _supertotal(10.0C), PRINTO=1
print list="\n", PRINTO=1

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= "Single Digit Facility Types ", PRINTO=1
Print list= "AType         1x          2x          3x          4x          5x          6x 
7x          8x          9x      Totals", PRINTO=1
Print list= "---------------------------------------------------------------------------------------------------------------------------------------------------
------------------------------------------
LOOP _a1iter2=100000,599999,100000 ;^Begin Loop 16: Cycles through ATYPE by 10 to 
_aat1=int(_a1iter2/100000) ; get single digit ATYPE.

_fftotal=0 ;^Initialize total X for all ATYPE
LOOP _f1iter=1000,9900,1000 ;^Begin Loop 17: Cycles through FTYPE by 10 to 
_totftlns=0 ;^Initialize total X for all FTYPE by all Lanes.
LOOP _fiter3=_f1iter,9900,100 ;^Begin Loop 18: Cycles through two-
digit FTYPE
if (_fiter3>_f1iter+999) BREAK
Loop 17.
LOOP _aiter6=_a1iter2,599999,100000 ;^Begin Loop 19: Cycles through two-
digit ATYPE
if (_aiter6>_a1iter2+99999) BREAK
Loop 16.
LOOP _liter7=1,9,1 ;^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
through Lanes for current FTYPE and ATYPE
totflIns=totflIns+_cntvmtval[aiter6+_fiter3+_liter7] ; in order to generate
total X for FTYPE by ATYPE.
ENDLOOP

ENDLOOP

fftotals=fftotals+totflIns ;^Generate total X for FTYPE.
ENDLOOP

print list="\",","_totflIns(10.0C)," ",PRINTO=1
ENDLOOP

print list="\",","_fftotals(10.0c), PRINTO=1
ENDLOOP

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0 ;^Initialize overall total X.
LOOP _f1iter2=1000,9900,1000
by 10

_fftotals=0
LOOP _fiter4=_f1iter2,9900,100
by 1 to
if (_fiter4>_f1iter2+999) BREAK
FTYPE in

LOOP _liter8=1,9,1
ENDLOOP

LOOP _aiter7=100000,599999,1000
in order

_fftotiter=_aiter7+_fiter4+_liter8
ENDLOOP

_fftotals=_fftotals+_cntvmtval[_fftotiter]
ENDLOOP

ENDLOOP

_supertotal=_supertotal+_fftotals ;^Generate overall total for all single
digit ATYPE

print list="\",","_fftotals(10.0C)," ",PRINTO=1
ENDLOOP

_vmtcountsoncounts=_supertotal
print list="\",","_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1

;************************************************************************
; END COUNT VMT REPORT
;************************************************************************

;============================================================================================= 
; BEGIN VOLUME/COUNT VMT REPORT ------------- X = Volumes over Counts VMT on Links w/ Counts
;============================================================================================= 
Print list=" ", PRINTO=1
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="* 

VMT Volume over Count Ratios on Links with Counts

Print list="** 
**, PRINTO=1
Print list="** 
**, PRINTO=1
Print list="** 
**, PRINTO=1
Print list="** 
**, PRINTO=1

Page | 161
LOOP _alter=100000,599999,100000
  _aat1=int(_alter/100000)
  print list= "Area Type ",_aat1(1.0),"x Range:",
  
  LOOP _aiter=_alter,599999,10000
    if (_aiter>_alter+99999) BREAK
    _aat2=int(_aiter/10000)
    _avcheck=0
    LOOP _achkiter=_aiter,599999,1
      _avcheck=_avcheck+_vcntby[_achkiter]
    ENDLOOP
    if (_avcheck>0)
      _supertotal=0
      _supercnts=0
      Print list= "Area Type ",_aat2(2.0), PRINTO=1
      Print list= "                                                  Number of Lanes per 
      Direction 
      Print list= "FType         1           2           3           4           5           6 
      7           8           9       Totals", PRINTO=1
      Print list= "------------------------------------------------------------------------------ 
      ------------------------------------------------", PRINTO=1
      LOOP _fiter=100,9900,100
        _vcheck=0
        LOOP _liter=1,9,1
          _vcheck=_vcheck+_vcntby[_aiter+_fiter+_liter]
        ENDLOOP
        if (_vcheck>0)
          _fft2=int(_fiter/100)
          LOOP _liter2=1,9,1
            _links=_volvmtval[_aiter+_fiter+_liter2]/_cntvmtval[_aiter+_fiter+_liter2]
            endif
            _links=0
            print list=":\", PRINTO=1
            _totvols=_totvols+_volvmtval[_aiter+_fiter+_liter2]
            _totcnts=_totcnts+_cntvmtval[_aiter+_fiter+_liter2]
          ENDLOOP
          LOOP _liter2=1,9,1
            if (_cntby[_aiter+_fiter+_liter2]>0) 
              _links=_volvmtval[_aiter+_fiter+_liter2]/_cntvmtval[_aiter+_fiter+_liter2]
              else
                _links=0
              endif
              print list="\", PRINTO=1
              _totvols=_totvols+_volvmtval[_aiter+_fiter+_liter2]
            
    ENDLOOP
  ENDLOOP
LOOP _alter=599999,100000
  _aal1=int(_alter/10000)
  print list= "Area Type ",_aal1(1.0),"x Range:",
  "\n ", PRINTO=1
  LOOP _alter=_alter,599999,100000
    if (_alter>_alter+99999) BREAK
    _aal2=int(_alter/10000)
    _avcheck=0
    LOOP _achkiter=_alter,599999,1
      _avcheck=_avcheck+_vcntby[_achkiter]
    ENDLOOP
    if (_avcheck>0)
      _supertotal=0
      _supercnts=0
      Print list= "Area Type ",_aal2(2.0), PRINTO=1
      Print list= "                                                  Number of Lanes per 
      Direction 
      Print list= "FType         1           2           3           4           5           6 
      7           8           9       Totals", PRINTO=1
      Print list= "------------------------------------------------------------------------------ 
      ------------------------------------------------", PRINTO=1
      LOOP _fiter=100,9900,100
        _vcheck=0
        LOOP _liter=1,9,1
          _vcheck=_vcheck+_vcntby[_alter+_fiter+_liter]
        ENDLOOP
        if (_vcheck>0)
          _fft2=int(_fiter/100)
          LOOP _liter2=1,9,1
            _links=_volvmtval[_alter+_fiter+_liter2]/_cntvmtval[_alter+_fiter+_liter2]
            endif
            _links=0
            print list=":\", PRINTO=1
            _totvols=_totvols+_volvmtval[_alter+_fiter+_liter2]
            _totcnts=_totcnts+_cntvmtval[_alter+_fiter+_liter2]
ENDLOOP ;^End Loop 6.

if (_totcnts>0)
   _totvc=_totvols/_totcnts
else
   _totvc=0
endif
print list="\\", _totvc(10.2C), PRINTO=1
endif ;^End Condition 2.
ENDLOOP ;^End Loop 4.

Print list= "-----------------------------------------------------------------------------------------------", PRINTO=1
print list= "Totals", PRINTO=1

LOOP _liter3=1,9,1 ;^Begin Loop 7: Cycles through Lanes
for
   _lntotals=0
   _lncnts=0
   LOOP _aiter2=_aiter,599999,100 ;^Begin Loop 8: Cycles through FTYPE
   for current ATYPE in Loop 2.
      if (_aiter2>_aiter+9999) BREAK ; in Loop 2 to generate Lane total X.
      _lntotiter=_aiter2+_liter3
      _lntotals=_lntotals+_volvmtval[_lntotiter]
      _lncnts=_lncnts+_cntvmtval[_lntotiter]
   ENDLOOP ;^End Loop 8
   if (_lncnts>0)
      _lnvc=_lntotals/_lncnts
   else
      _lnvc=0
   endif
   print list="\\", _lnvc(10.2C), PRINTO=1
   ENDLOOP ;^End Loop 7
   if (_supercnts>0)
      _supervc=_supertotal/_supercnts
   else
      _supervc=0
   endif
   print list="\\", _supervc(10.2C), PRINTO=1
   print list=" ", PRINTO=1
   endif ;^End Condition 1.
ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES---------------------

Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= "
FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100 ;^Begin Loop 9: Cycles through FTYPES
to get
   _fft2=int(_fiter2/100)
   _tafvcheck=0
   _tafcnts=0
   LOOP _liter5=1,9,1 ;^Begin Loop 10: Cycles through Lanes
for current
LOOP _aiter4= 100000,599999,10000
for _tafvcheck=_tafvcheck+_volvmtval[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
 Endloop 

Endloop

if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE in Loop 9
print list= _fft2(2.0),"    ", PRINTO=1
else
skip FTYPE.
endif

LOOP _liter4= 1,9,1 ;^Begin Loop 12: Cycles through Lanes for current FTYPE
_totftat=0
_totcnts=0

LOOP _aiter3= 100000,599999,10000 ;^Begin Loop 13: Cycles through ATYPE for current Lanes in Loop 12
_totftat=_totftat+_volvmtval[_aiter3+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
_totcnts=_totcnts+_cntvmtval[_aiter3+_fiter2+_liter4]
Endloop

if (_totcnts>0)
_totvc=_totftat/_totcnts
else
_totvc=0
endif
print list="\"," ",_totvc(10.2C)," ", PRINTO=1
Endloop

if (_tafcnts>0)
_tafvc=_tafvcheck/_tafcnts
else
_tafvc=0
endif

if (_tafcnts>0) ;^Begin Condition 3: If current FTYPE in Loop 9
print list= _fft2(2.0),"    ", PRINTO=1
else
skip FTYPE.
endif

Endloop

Print list= "---------------------------------------------------------------------------
----------------------------------------
Totals"
_supertotal=0
_supercnts=0

LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.
_intotals=0
_lncnts=0

LOOP _aiter5=100000,599999,100 and _lntotiter=_aiter5+_liter6
for _lntotals=_lntotals+_volvmtval[_lntotiter] ; Lanes.
_lncnts=_lncnts+_cntvmtval[_lntotiter]
Endloop

if (_lncnts>0)
_lnvc=_lntotals/_lncnts
else
_lnvc=0
endif
print list="\"," ",_lnvc(10.2C)," ", PRINTO=1
_supertotal=_supertotal+_lntotals ;^Generate total X for all ATYPE.
_supercnts=_supercnts+_lncnts
ENDLOOP  ;^End Loop 14.
if (_supercnts>0)
  _supervc=_supertotal/_supercnts
else
  _supervc=0
endif
print list="\\",_supervc(10.2C), PRINTO=1
print list=" \n ", PRINTO=1

ENDLOOP  ;^End Loop 14.

-----------------------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY--------------------------

Print list= "Total Summary Area Types by Facility Types \\n\n", PRINTO=1 ;^Header
Print list= "Single Digit Facility Types \\n\n", PRINTO=1

Print list= "AType         1x          2x          3x          4x          5x          6x          7x          8x          9x      Totals \\n\n------------------------------------------------------------------------------------
\n\n------------------------------------------ \\n
LOOP _a1iter2=100000,599999,100000  ;^Begin Loop 16: Cycles through ATYPE by 10 to
by 10 to
  _aat1=int(_a1iter2/100000)        ; get single digit ATYPE.
  print list= _aat1(1.0),"x", PRINTO=1

  _fttotal=0
  _ftcnts=0

LOOP _f1iter=1000,9900,1000       ;^Begin Loop 17: Cycles through FTYPE by 10 to
by 10 to
  _totftlns=0
  _totftcnts=0

LOOP _fiter3=_f1iter,9900,100     ;^Begin Loop 18: Cycles through two-
digit FTYPE
  if (_fiter3>_f1iter+999) BREAK    ; for current single digit FTYPE in
Loop 17.

LOOP _a1iter6=_a1iter2,599999,100000  ;^Begin Loop 19: Cycles through two-
digit ATYPE
  if (_a1iter6>_a1iter2+99999) BREAK  ; for current single digit ATYPE in
Loop 16.

  LOOP _liter7=1,9,1           ;^Begin Loop 20: Cycles
    through Lanes for current FTYPE and ATYPE
    _totftlns=_totftlns+_volvmtval[_a1iter6+_fiter3+_liter7]  ; in order to generate
    _totftcnts=_totftcnts+_cntvmtval[_a1iter6+_fiter3+_liter7]
  ENDLOOP  ;^End Loop 20.
ENDLOOP  ;^End Loop 19.
ENDLOOP
ENDLOOP

  _fttotal=_fttotal+_totftlns
  _ftcnts=_ftcnts+_totftcnts

if (_totftcnts>0)
  _totftvc=_totftlns/_totftcnts
else
  _totftvc=0
endif
print list="\\",_totftvc(10.2C),",", PRINTO=1
ENDLOOP  ;^End Loop 17.
if (_ftcnts>0)
  _ftvc=_fttotal/_ftcnts
else
  _ftvc=0
endif
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

```
print list="\",",_ftvc(10.2c), PRINTO-1
ENDLOOP ;^End Loop 16.

Print list=""-----------------------------------------------------------------------------------------------", PRINTO-1
print list="Totals", PRINTO-1

_supertotal=0
スーパーカウント=0

LOOP _filter2=1000,9900,1000 by 10
 Superintendent=0
スーパーファイリング=0

LOOP _filter4=_filter2,9900,100 by 1 to
 Superintendent=0
スーパーファイリング=0

LOOP _filter4=_filter2,9900,100 by 1 to
 Superintendent=0
スーパーファイリング=0

 Superintendent=superntotal+ftotals
 Supermanカウント= superfntotal+ftcnts

if (_ftcnts>0)
 Superintendent=_footals/_ftcnts
 Supermanカウント= superfntotal/ superfntcnts
else
 Superintendent=0
 Supermanカウント=0
endif
print list="\",",_ftvc(10.2C),", PRINTO-1
ENDLOOP ;^End Loop 21.

if (_supercnts>0)
 Superintendent=superntotal/supercnts
 Supermanカウント= superfntotal/ superfntcnts
else
 Superintendent=0
 Supermanカウント=0
endif
print list="\",",_ftvc(10.2C),", PRINTO-1
ENDLOOP ;^End Loop 22.

;************************************************************************
; END VMT VOLUME OVER COUNT REPORT
;************************************************************************

;======================================================================================
; BEGIN VHT VAL REPORT ------------- X = VHT on Links w/ Counts
;======================================================================================

print list="\",",_ftvc(10.2C),", PRINTO-1

%;*******************************************************************************************
;*******************************************************************************************

; BEGIN VMT VOLUME OVER COUNT REPORT
;***************************************************************************

; BEGIN VHT VAL REPORT ------------- X = VHT on Links w/ Counts
;***************************************************************************

Vehicle Hours Traveled (VHT) using Volumes on Links with Counts
```

Page | 166
LOOP _alter=100000,599999,100000
_types (ATYPE) by 10
_print list= "Area Type ",_aat1(1.0),"x Range:",
="", PRINTO=1
LOOP _alter=_alter,599999,10000
1
if (_alter>_alter+99999) BREAK
_endloop

LOOP _achkiter=_alter,599999,1
if (_achkiter>_alter+9999) BREAK
_avcheck=0
_ENDLOOP

LOOP _fiter=100,9900,100
_vcheck=0
_ENDLOOP

_LOOP _liter=1,9,1
_vcheck=_vcheck+_volvhtval[_alter+_fiter+_liter]
_ENDLOOP

_LOOP _liter2=1,9,1
Totals" PRNINT=1
_ENDLOOP

_LOOP _filter=100,9900,100
_vcheck=0
_LOOP _liter=1,9,1
_vcheck=_vcheck+_volvhtval[_alter+_filter+_liter]
_ENDLOOP

_LOOP _liter2=1,9,1
_generate ATYPE by FTYPE by Lanes total X.
print list="\",",totvols(10.0C), PRINTO=1
_ENDLOOP

ENDLOOP

print list="\",",totvols(10.0C), PRINTO=1
ENDLOOP

ENDLOOP

PRINT list="-----------------------------------------------------", PRINTO=1
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

print list="Totals", PRINTO=1

LOOP _liter3=1,9,1 ;^Begin Loop 7: Cycles through Lanes

_for current ATYPE in Loop 2.

_\lntotals=0 ;^Initialize Lane total X.

LOOP _aiter2=_aiter,599999,100 ;^Begin Loop 8: Cycles through FTYPE

for current ATYPE in Loop 2 to generate Lane total X.

if (_aiter2>_aiter+9999) BREAK

_\lntotiter=_aiter2+_liter3

_\lntotals_\lntotals+_volvhtval[\lntotiter]

ENDLOOP ;^End Loop 8

print list="\\", ",_\lntotals(10.0C),", PRINTO=1

ENDLOOP ;^End Loop 7

print list="\\", ",_supertotal(10.0C), PRINTO=1

print list=" ", PRINTO=1

endif ;^End Condition 1.

ENDLOOP ;^End Loop 2.

print list=" ", PRINTO=1

ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------

Print list= "Total Area Types ", PRINTO=1 ;^Header

Print list= " Number of Lanes per Direction ", PRINTO=1

Print list= "FType  1     2     3     4     5     6     7     8     9     Totals", PRINTO=1

Print list= "------------------------------------------------------------------------------------

------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100 ;^Begin Loop 9: Cycles through FTYPES

to get

_\fft2=int(_fiter2/100)

_\tafvcheck=0 ;^Initialize FTYPE X checking variable.

LOOP _liter5=1,9,1 ;^Begin Loop 10: Cycles through Lanes

for current FTYPE

LOOP _aiter4= 100000,599999,10000 ;^Begin Loop 11: Cycles through ATYPE

for _tafvcheck=_tafvcheck+_volvhtval[\aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in

order to total X checking variable.

ENDLOOP ;^End Loop 11.

ENDLOOP ;^End Loop 10.

if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE

in Loop 9 has X>0 continue to report X. Else

skip FTYPE.

LOOP _liter4= 1,9,1 ;^Begin Loop 12: Cycles through Lanes

for current FTYPE

_\totftat=0 ;^Initialize FTYPE total X for all

ATYPE.

LOOP _aiter3= 100000,599999,10000 ;^Begin Loop 13: Cycles through ATYPE

for current Lanes in Loop 12

_\totftat=_\totftat+_volvhtval[\aiter3+_fiter2+_liter4] ; in order to generate total X for

FTYPE by Lane for all ATYPE.

ENDLOOP ;^End Loop 13.

print list="\\", ",_\totftat(10.0C),", PRINTO=1

ENDLOOP
ENDLOOP

print list="\", _tafvcheck(10.0C), PRINTO=1
endif ;^End Condition 3.
ENDLOOP ;^End Loop 9.

Print list= "-------------------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
_supertotal=0 ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.
    _lntotals=0 ;^Initialize total X for Lanes.
    LOOP _aiter5=100000,599999,100 and
        _lntotiter=_aiter5+_liter6
        for
        _lntotals=_lntotals+_volvhtval[_lntotiter] ; Lanes.
        ENDLOOP ;^End Loop 15.
    print list="\", _lntotals(10.0C),"", PRINTO=1
    _supertotal=_supertotal+_lntotals ;^Generate total X for all ATYPE.
 ENDLOOP ;^End Loop 14.
print list="\", _supertotal(10.0C), PRINTO=1
print list=" 

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= " Single Digit Facility Types ", PRINTO=1
Print list= " AType            1x           2x           3x           4x           5x           6x 
7x           8x           9x           Totals", PRINTO=1
Print list= "-------------------------------------------------------------", PRINTO=1
ENDLOOP ;^End Loop 12.

print list="\", _tafvcheck(10.0C), PRINTO=1 ;^End Loop 12.
ENDLOOP

PRINTO=1
_loop2=100000,599999,100000
_loop21=int(_loop2/100000) ; get single digit ATYPE.
_loop2f0=0 ;^Initialize total X for all ATYPE
_LOOP_fiter3=_fiter3,9900,100
_LOOPS2=0 ; get single digit FTYPE.
_LOOP_fiter3=_fiter3,9900,100
if (_fiter3>_fiter3+999) BREAK ; for current single digit FTYPE in Loop 17.
_LOOP_aiter6=_aiter62,599999,100000
if (_aiter6>._aiter6+299999) BREAK ; for current single digit ATYPE in Loop 16.
_LOOP_liter7=1,9,1
through Lanes for current FTYPE and ATYPE
_loop2f2=0 ; in order to generate total X for FTYPE by ATYPE.
ENDLOOP ;^End Loop 20.
ENDLOOP ;^End Loop 19.
ENDLOOP ;^End Loop 18.
_fttotal=_fttotal+_totftlns ;^Generate total X for ATYPE.
ENDLOOP

print list="\\", ",_totftlns(10.0C)," ", PRINTO=1
ENDLOOP ;^End Loop 17.

print list="\\", ",_fttotal(10.0c), PRINTO=1
ENDLOOP ;^End Loop 16.

Print list= "-------------------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0 ;^Initialize overall total X.
LOOP _f1iter2=1000,9900,1000 by 10

_ftotals=0 ;^Initialize total X by FTYPE
LOOP _fiter4=_f1iter2,9900,100
if (_fiter4>_f1iter2+999) BREAK ; get all two-digit FTYPE for current
  FTYPE in
    LOOP _liter8=1,9,1
        LOOP _aiter7=100000,599999,1000 ;^Begin Loop 24: Cycles through ATYPE
            _ftotiter=_aiter7+_fiter4+_liter8 ; to generate total X by single digit
            _ftotals=_ftotals+_volvhtval[_ftotiter]
        ENDLOOP
    ENDLOOP
ENDLOOP

_ENDLOOP
_LOOP _f2iter2=1000,9900,1000 by 10

_SUPERTOTAL=0 ;^Generate overall total for all single
digit ATYPE ; by all single digit FTYPE.
ENDLOOP ;^End Loop 21.

_vhtvoloncounts=_supertotal
print list="\\", ",_supertotal(10.0C)," ", PRINTO=1
print list=" ", PRINTO=1
; END VHT VAL REPORT
;*******************************************************************************************
*************************************
Vehicle Hours Traveled (VHT) using Counts on Links with Counts
*************************************
; END VHT VAL REPORT
;*******************************************************************************************
                      2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000                         ;^Begin Loop 1: Cycles through Area
Types (ATYPE) by 10
_print list= "Area Type ",_aat1(1.0),"x Range:",
"\n ", PRINTO=1

LOOP _a1iter=599999,100000
_loop=1
  if (_a1iter+99999) BREAK
  _aat2=int(_a1iter/10000)

  _avcheck=0                                            ;^Initialize ATYPE X checking variable.
  LOOP _achkiter=_a1iter,599999,10000                        ;^Begin Loop 3: Cycles through Lanes
    if (_achkiter>_a1iter+9999) BREAK
    _avcheck=_avcheck+_cntvhtval[_achkiter]
  ENDLOOP

  if (_avcheck>0)                                       ;^Begin Condition 1: If current ATYPE
    _supertotal=0                                       ;^Initialize ATYPE total X.
    _supertotal=0                                       ;^Initialize ATYPE total X.
    Print list= _aat2(2.0), PRINTO=1
    Print list= "                                                  Number of Lanes per
    Direction ", PRINTO=1
    Print list= "FType         1           2           3           4           5           6
    7           8           9       Totals", PRINTO=1
  ENDLOOP

  LOOP _fiter=100,9900,100                            ;^Begin Loop 4: Cycles through FTYPE
    _vcheck=0                                          ;^Initialize FTYPE X checking variable.
    LOOP _liter=1,9,1                                  ;^Begin Loop 5: Cycles through Lanes
      _vcheck=_vcheck+_cntvhtval[_a1iter+_fiter+_liter]    ; FTYPE in Loop 4 and totals X
      _vcheck=_vcheck+_cntvhtval[_a1iter+_fiter+_liter]
    ENDLOOP

    if (_vcheck>0)                                    ;^Begin Condition 2: If current FTYPE
      _fft2=int(_fiter/100)                          ; has X>0 continue to report X. Else
      _totvols=0                                     ;^Initialize FTYPE total X.
      LOOP _liter2=1,9,1                             ;^Begin Loop 6: Cycles through Lanes to
        print list= _fft2(2.0),",    ", PRINTO=1
        _totvols=0
      ENDLOOP

      _totvols=_totvols+_cntvhtval[_a1iter+_fiter+_liter2]
      _supertotal=_supertotal+_cntvhtval[_a1iter+_fiter+_liter2]
      _totvols=_totvols+_cntvhtval[_a1iter+_fiter+_liter2]
    ENDLOOP

    print list="\",", ",_totvols(10.0C), PRINTO=1
  endif                                             ;^End Condition 2.
ENDLOOP                                             ;^End Loop 4.

=----------------------------------------------------------------------------------------------------=

LOOP _filter=100,99000,10000                         ;^Begin Loop 5: Cycles through Lanes
  _vcheck=0                                          ;^Initialize FTYPE X checking variable.
  LOOP _liter=1,9,1
    _vcheck=_vcheck+_cntvhtval[_filter+_liter]        ; FTYPE in Loop 4 and totals X

  if (_vcheck>0)                                    ;^Begin Condition 2: If current FTYPE
    _fft2=int(_filter/100)                          ; has X>0 continue to report X. Else
    _totvols=0                                     ;^Initialize FTYPE total X.
  ENDLOOP
  LOOP _liter2=1,9,1
    print list="\",",",_cntvhtval[_filter+_liter2](10.0C),", ", PRINTO=1
    _totvols=0
  ENDLOOP

  _lntotals=0                                       ;^Initialize Lane total X.
  LOOP _liter3=1,9,1                                  ;^Begin Loop 7: Cycles through Lanes
    print list="\",", ",_lntotals(10.0C), PRINTO=1
  ENDLOOP

=----------------------------------------------------------------------------------------------------=

```plaintext
LOOP _aiter2=_aiter,599999,100
  ;^Begin Loop 8: Cycles through FTYPE
  for current ATYPE
    if (_aiter2>_aiter+9999) BREAK
    _lntotiter=_aiter2+_liter3
    _lntotals=_lntotals+_cntvhtval[_lntotiter]
  ENDLOOP
  ;^End Loop 8

  print list="\\", ",_lntotals(10.0C)," ", PRINTO=1
ENDLOOP
  ;^End Loop 7

print list="\\", ",_supertotal(10.0C), PRINTO=1
endif
  ;^End Condition 1.
ENDLOOP
  ;^End Loop 2.
ENDLOOP
  ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1
Print list= " Number of Lanes per Direction"
Print list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "-----------------------------------------------------------------------------------"

LOOP _fiter2=100,9900,100
  ;^Begin Loop 9: Cycles through FTYPES
to get
  _fft2=int(_fiter2/100)
  _tafvcheck=0
  LOOP _liter5=1,9,1
    ;^Begin Loop 10: Cycles through Lanes for current FTYPE
    LOOP _aiter4= 100000,599999,10000
      ;^Begin Loop 11: Cycles through ATYPE for current Lanes and FTYPE
      _tafvcheck=_tafvcheck+_cntvhtval[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
    ENDLOOP
    ;^End Loop 11.
  ENDLOOP
  ;^End Loop 10.
  if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE in Loop 9 has X>0 continue to report X. Else skip FTYPE.
    LOOP _liter4= 1,9,1
      ;^Begin Loop 12: Cycles through Lanes in Loop 9
      _totftat=0
      LOOP _aiter3= 100000,599999,10000
        ;^Begin Loop 13: Cycles through ATYPE for current Lane in Loop 12
        _totftat=_totftat+_cntvhtval[_aiter3+_fiter2+_liter5] ; current Lane X for all ATYPE in order to generate total X for FTYPE by Lane for all ATYPE.
      ENDLOOP
      ;^End Loop 13.

      print list="\\", ",_totftat(10.0C)," ", PRINTO=1
    ENDLOOP
    ;^End Loop 12.
  endif
  ;^End Condition 3.
```
ENDLOOP                                                   ;^End Loop 9.

Print list= "-----------------------------------------------------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0                                             ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1                                          ;^Begin Loop 14: Cycles through Lanes.
    _lntotals=0                                            ;^Initialize total X for Lanes.
    LOOP _aiter5=100000,599999,100                     ;^Begin Loop 15: Cycles through ATYPE and
        _lntotiter=_aiter5+_liter6                        ; FTYPE in order to generate total X
        _lntotals=_lntotals+_cntvhtval[_lntotiter]              ; Lanes.
        ;^End Loop 15.
    ENDLOOP                                                 ;^End Loop 14.
    print list="\",_lntotals(10.0C),",", PRINTO=1
    _supertotal=_supertotal+_lntotals                    ;^Generate total X for all ATYPE.
ENDLOOP                                                   ;^End Loop 13.

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= "Single Digit Facility Types ", PRINTO=1
Print list= "AType  1x  2x  3x  4x  5x  6x  7x  8x  9x  Totals", PRINTO=1
Print list= "-----------------------------------------------------------------------------------------------", PRINTO=1
LOOP _a1iter2=100000,599999,100000                         ;^Begin Loop 16: Cycles through ATYPE
    _aat1=int(_a1iter2/100000)                         ; get single digit ATYPE.
    print list= _aat1(1.0),"x","    ", PRINTO=1
    _fttotal=0                                          ;^Initialize total X for all ATYPE
    LOOP _f1iter=1000,9900,100                          ;^Begin Loop 17: Cycles through FTYPE by
        _totftlns=0                                      ; all Lanes.
        LOOP _fiter3=_f1iter,9900,100                   ;^Begin Loop 18: Cycles through two-
            _totftlns=_totftlns+_cntvhtval[_aiter6+_fiter3+_liter7]   ; digit FTYPE
            ; for current single digit FTYPE in
            ; Loop 17.
        ENDLOOP                                           ;^End Loop 19.
        _fttotal=_fttotal+_totftlns                      ;^Generate total X for ATYPE.
ENDLOOP                                                   ;^End Loop 18.

ENDLOOP                                             ;^End Loop 17.
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

print list="\\" ",_totftlns(10.0C)," ", PRINTO=1
ENDLOOP ;^End Loop 17.

print list="\\" ",_fttotal(10.0C), PRINTO=1
ENDLOOP ;^End Loop 16.

Print list= "-----------------------------------------------------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
_supertotal=0 ;^Initialize overall total X.

LOOP _f1iter2=1000,9900,1000 ;^Begin Loop 21: Cycles through FTYPE by 10
_by 10'
_ftotals=0 ;^Initialize total X by FTYPE
LOOP _fiter4=_f1iter2,9900,100 ;^Begin Loop 22: Cycles through FTYPE by 1 to
get all two-digit FTYPE for current
FTYPE in

_LOOP _liter8=1,9,1
_LOOP _aiter7=100000,599999,10000 ;^Begin Loop 24: Cycles through ATYPE in order
_ftotiter=_aiter7+_fiter4+_liter8 ; to generate total X by single digit
_FTYPE._ftotals=_ftotals+_cntvhtval[_ftotiter] ;^End Loop 24.
ENDLOOP ;^End Loop 23.
ENDLOOP ;^End Loop 22.
_supertotal=_supertotal+_ftotals ;^Generate overall total for all single
digit ATYPE; by all single digit FTYPE.

print list="\\" ",_ftotals(10.0C), PRINTO=1
ENDLOOP ;^End Loop 21.

_vhtcountsoncounts=_supertotal
print list="\\" ",_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
;************************************************************************
; END COUNT VHT REPORT
;************************************************************************
;=============================================================================================  
; BEGIN VOLUME/COUNT VHT REPORT ------------- X = Volumes over Counts VHT on Links w/ Counts
;============================================================================================= 

Print list=" ", PRINTO=1
Print
list="********************************************************************************************
********************************************************************************************
*
*
*                                 VHT Volume over Count Ratios on Links with Counts
*
*
********************************************************************************************
********************************************************************************************

;-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES--------------------
_LOOP _a1iter=100000,599999,100000 ;^Begin Loop 1: Cycles through Area
_TYPES (ATYPE) by 10
_aat1=int(_a1iter/100000) ; in order to get single digit ATYPE.
print list= "Area Type ",_aat1(1.0),"x Range:",
\"n ", PRINTO=1
LOOP _a1iter=_a1iter,599999,10000
1        if (_a1iter>_a1iter+99999) BREAK
         _aat2=int(_a1iter/10000)
         _avcheck=0
         LOOP _achkiter=_a1iter,599999,1
         and Facility Types (FTYPE)
         if (_achkiter>_a1iter+9999) BREAK
         totals X checking variable.
         _avcheck=_avcheck+_vcntby[_achkiter]
         ENDLOOP
         if (_avcheck>0) in Loop 2
         skip ATYPE.
         _supertotal=0
         _supercnts=0
         Print list= "Area Type ",_aat2(2.0), PRINTO=1
         Print list= "                                                  Number of Lanes per
         Direction                                               ", PRINTO=1
9       Print list= " ATYPE         1           2           3           4           5           6
7       Print list= "FType         8           9           Totals", PRINTO=1
8       Print list= "------------------------------------------------------------------------------
7       Print list= "------------------------------------------------", PRINTO=1
9       LOOP _fiter=100,9900,100
       _vcheck=0
       LOOP _liter=1,9,1
       for current
       _vcheck=_vcheck+_vcntby[_a1iter+_fiter+_liter] checking variable.
       ENDLOOP
       if (_vcheck>0) in Loop 4
       skip FTYPE.
       _fft2=int(_fiter/100)
       Print list= _fft2(2.0),"     ", PRINTO=1
       _totvols=0
       _totcnts=0
       LOOP _liter2=1,9,1
       to generate
       if (_cntby[_a1iter+_fiter+_liter2]>0)
       ATYPE by FTYPE by lanes total X.
       _links=_volvhtval[_a1iter+_fiter+_liter2]/_cntvhtval[_a1iter+_fiter+_liter2]
       else
       _links=0
       endif
       print list="\"," ",_links(10.2C)," ", PRINTO=1
       _totvols=_totvols+_volvhtval[_a1iter+_fiter+_liter2]
       _totcnts=_totcnts+_cntvhtval[_a1iter+_fiter+_liter2]
       _supertotal=_supertotal+_volvhtval[_a1iter+_fiter+_liter2]
       _supercnts=_supercnts+_cntvhtval[_a1iter+_fiter+_liter2]
       ENDLOOP
       if (_totcnts>0)
       _totvc=_totvols/_totcnts
       else
       _totvc=0
       endif
       print list="\"," ",_totvc(10.2C), PRINTO=1
       endif
       ENDLOOP
```
print list= "-----------------------------", PRINTO=1
print list="Totals", PRINTO=1

LOOP _liter3=1,9,1
for
    _lntotals=0
    _lnncnts=0

    LOOP _aiter2=_aiter,599999,100
        ;^Begin Loop 8: Cycles through FTYPE
        if (_aiter2>_aiter+9999) BREAK ; in Loop 2 to generate Lane total X.
        _lntotiter=_aiter2+_liter3
        _lntotals=_lntotals+_volvhtval[_lntotiter]
        _lnncnts=_lnncnts+_cntvhtval[_lntotiter]
    ENDDO
    if (_lnncnts>0)
        _lnvc=_lntotals/_lnncnts
    else
        _lnvc=0
    endif

ENDLOOP ;^End Loop 7

if (_supercnts>0)
    _supervc=_supertotal/_supercnts
else
    _supervc=0
endif

ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES-------------------

Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= " Number of Lanes per Direction ", PRINTO=1
Print list= " FType     1         2         3         4         5         6 
8         9       Totals", PRINTO=1
Print list= "-----------------------------", PRINTO=1

LOOP _fiter2=100,9900,100
for
    _fft2=int(_fiter2/100)
    _tafvcheck=0
    _tafcnts=0

    LOOP _liter5=1,9,1
        ;^Begin Loop 10: Cycles through Lanes
        for current
            LOOP _aiter4= 100000,599999,10000
                ;^Begin Loop 11: Cycles through ATYPE
                _tafcnts=_tafcnts+_cntvhtval[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
                _tafvcheck=_tafvcheck+_volvhtval[_aiter4+_fiter2+_liter5] ; two-digit FTYPE.
            ENDDO
        ENDDO
    ENDDO
ENDLOOP ;^End Loop 9.
```

if (_tafvcheck>0) ;"Begin Condition 3: If current FTYPE in Loop 9
print list= _fft2(2.0)," ", PRINTO=1 ; has X>0 continue to report X. Else
  skip FTYPE.
LOOP _liter4= 1,9,1 ;"Begin Loop 12: Cycles through Lanes
  for current FTYPE
    _totftat=0 ;"Initialize FTYPE total X for all ATYPE.
    _totcnts=0

    LOOP _aiter3= 100000,599999,10000 ;"Begin Loop 13: Cycles through ATYPE
      for current Lanes in Loop 12
        _totftat=_totftat+_volvhtval[_aiter3+_fiter2+_liter4] ; in order to generate total X for
        _totcnts=_totcnts+_cntvhtval[_aiter3+_fiter2+_liter4]
      ENDLOOP ;"End Loop 13.
      if (_totcnts>0)
        _totvc=_totftat/_totcnts
      else
        _totvc=0
      endif

      print list="\",_totvc(10.2C)," ", PRINTO=1 ;"End Loop 12.
    ENDLOOP ;"End Condition 3.
  ENDLOOP ;"End Loop 9.

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0 ;"Initialize all ATYPE total X.
_supercnts=0

LOOP _liter6=1,9,1 ;"Begin Loop 14: Cycles through Lanes.
  _lntotals=0 ;"Initialize total X for Lanes.
  _lncnts=0

  LOOP _aiter5=100000,599999,100 ;"Begin Loop 15: Cycles through ATYPE
    _lntotiter=_aiter5+_liter6 ; FTYPE in order to generate total X
    _lntotals=_lntotals+_volvhtval[_lntotiter] ; Lanes.
  ENDLOOP ;"End Loop 15.
  if (_lncnts>0)
    _lnvc=_lntotals/_lncnts
  else
    _lnvc=0
  endif

  print list="\",_lnvc(10.2C)," ", PRINTO=1 ;"Generate total X for all ATYPE.
  _supertotal=_supertotal+_lntotals
  _supercnts=_supercnts+_lncnts
ENDLOOP ;"End Loop 14.
  if (_supercnts>0)
    _supervc=_supertotal/_supercnts
  else
    _supervc=0
  endif

print list="\",_supervc(10.2C), PRINTO=1
print list=" ", PRINTO=1
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

----------------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------------

Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;"Header
Print list= " Single Digit Facility Types ", PRINTO=1
Print list= "AType  1x  2x  3x  4x  5x  6x  7x  8x  9x  Totals", PRINTO=1
Print list= "-----------------------------------------------------------------------------------------------", PRINTO=1

LOOP _a1iter2=1,000000,599999,100000                        ;^Begin Loop 16: Cycles through ATYPE by 10 to
_aat1=int(_a1iter2/100000)                              ; get single digit ATYPE.
print list= _aat1(1.0),"x","    ", PRINTO=1
_fttotal=0                                              ;^Initialize total X for all ATYPE
_ftcnts=0
LOOP _f1iter=1,000,9900,100                           ;^Begin Loop 17: Cycles through FTYPE by 10 to
_totftlns=0                                         ;^Initialize total X for all FTYPE by all Lanes.
_totftcnts=0
LOOP _fiter3=_f1iter,9900,100                       ;^Begin Loop 18: Cycles through two-digit FTYPE
if (_fiter3>_f1iter+999) BREAK                    ; for current single digit FTYPE in Loop 17.
LOOP _aiter6=_a1iter2,599999,10000                ;^Begin Loop 19: Cycles through two-digit ATYPE
if (_aiter6>_a1iter2+99999) BREAK               ; for current single digit ATYPE in Loop 16.
_LOOP _liter7=1,9,1                                      ;^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_volvhtval[_aiter6+_fiter3+_liter7]   ; in order to generate total X for FTYPE by ATYPE.
_totftcnts=_totftcnts+_cntvhtval[_aiter6+_fiter3+_liter7]
ENDLOOP                                                 ;^End Loop 20.
ENDLOOP                                           ;^End Loop 19.
ENDLOOP                                             ;^End Loop 18.
_fttotal=_fttotal+_totftlns                         ;^Generate total X for ATYPE.
_ftcnts=_ftcnts+_totftcnts
if (_totftcnts>0)
_totftvc=_totftlns/_totftcnts                  ;^End Loop 17.
else
_totftvc=0
endif
print list="\","," ,_totftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_ftcnts>0)
_ftvc=_fttotal/_ftcnts
else
_ftvc=0
endif
print list="\"," ",_ftvc(10.2c), PRINTO=1
ENDLOOP                                                 ;^End Loop 16.
Print list= "-----------------------------------------------------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
_supertotal=0                                            ;^Initialize overall total X.
_superusernts=0
LOOP _filter2=1000,9900,1000 by 10
   _ftotals=0
   _ftcnts=0
LOOP _filter4=_filter2,9900,100
   if (_filter4>_filter2+999) BREAK
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_filter4+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000
   _aat1=int(_a1iter/100000)
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_a1iter+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000
   _aat1=int(_a1iter/100000)
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_a1iter+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000
   _aat1=int(_a1iter/100000)
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_a1iter+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000
   _aat1=int(_a1iter/100000)
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_a1iter+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000
   _aat1=int(_a1iter/100000)
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_a1iter+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _a1iter=100000,599999,100000
   _aat1=int(_a1iter/100000)
   LOOP _liter8=1,9
      LOOP _aiter7=100000,599999,10000
         _ftotiter=_aiter7+_a1iter+_liter8
         _ftotals=_ftotals+_volvhtval[_ftotiter]
         _ftcnts=_ftcnts+_cntvhtval[_ftotiter]
      ENDLOOP
   ENDLOOP
   _supertotal=_supertotal+_ftotals
   _supercnts=_supercnts+_ftcnts
   if (_ftcnts>0)
      _ftvc=_ftotals/_ftcnts
   else
      _ftvc=0
   endif
   print list="\\",",",_ftvc(10.2C)," ", PRINTO=1
ENDLOOP
if (_supercnts>0)
   _supervc=_supertotal/_supercnts
else
   _supervc=0
endif
_vhtvolovercounts=_supervc
print list="\\",",_supervc(10.2C),", PRINTO=1
print list="\\",",_supervc(10.2C),", PRINTO=1
/**********************************************************************************
********** END VHT VOLUME OVER COUNT REPORT
**********************************************************************************

BEGIN VOLUME REPORT ---------------- X = Volumes on Links w/ Counts
*******************************************************************************************
*******************************************************************************************
-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
print list= "Area Type ",_aat1(1.0),"x Range: ",
"n ", PRINTO=1

LOOP _aiter=_a1iter,599999,10000
1
  if (_aiter>_a1iter+99999) BREAK
  _avcheck=0
  LOOP _achkiter=_aiter,599999,10000                        ;^Begin Loop 2: Cycles through ATYPE by
  and Facility Types (FTYPE)
    if (_achkiter>_aiter+9999) BREAK                      ; in order to get two-digit ATYPE.
    _avcheck= _avcheck+ _volby[_achkiter]
  ENDLOOP                                               ;^Initialize ATYPE X checking variable.

  if (_avcheck>0)                                       ;^Begin Condition 1: If current ATYPE
    _supertotal=0                                       ; has X>0 continue to report X. Else
    Print list= "Area Type ",_aat2(2.0), PRINTO=1       ;^Initialize ATYPE total X.
    Print list= "                                                  Number of Lanes per
    Print list= "FType         1           2           3           4           5           6
    Print list= "7           8           9       Totals", PRINTO=1
    Print list= "------------------------------------------------------------------------------
    Print list= "------------------------------------------------", PRINTO=1
    LOOP _fiter=100,9900,100                            ;^Begin Loop 4: Cycles through FTYPE
      _vcheck=0                                          ; by 1 in order to get two-digit FTYPE.
      LOOP _liter=1,9,1                                  ;^Initialize FTYPE X checking variable.
        _vcheck= _vcheck+ _volby[_aiter+_fiter+_liter]    ;^Begin Loop 5: Cycles through Lanes
        ENDLOOP                                            ; for current ATYPE in Loop 2 and
      if (_vcheck>0)                                    ;^End Loop 5.
        _fft2=int(_fiter/100)                          ;^Begin Condition 2: If current FTYPE
        _totvols=0                                     ; has X>0 continue to report X. Else
        LOOP _liter2=1,9,1                             ;^Initialize FTYPE total X.
          print list= _fft2(2.0),"   ", PRINTO=1        ;^Begin Loop 6: Cycles through Lanes to
          _totvols= _totvols+ _volby[_aiter+_fiter+_liter2]    ; generate ATYPE by FTYPE by Lanes total X.
          ENDLOOP                                        ;^End Loop 6.
          print list="\n","   ",_totvols(10.0C), PRINTO=1
        endif                                             ;^End Condition 2.
      ENDLOOP                                           ;^End Loop 4.

print list= "Totals", PRINTO=1
LOOP _liter3=1,9,1                                     ;^Begin Loop 7: Cycles through Lanes
    _lntotals=0                                       ; current ATYPE in Loop 2.
    LOOP _a1iter2=_a1iter,599999,10000                 ;^Initialize Lane total X.
        for current ATYPE
          print list="\n","   ",_totvols(10.0C), PRINTO=1
          ENDLOOP                                        ;^End Loop 8: Cycles through FTYPE
if (_aiter2>_aiter+9999) BREAK ; in Loop 2 to generate Lane total X.
_lntotiter=_aiter2+_liter3
_lntotals=_lntotals+_volby[_lntotiter]
ENDLOOP ;^End Loop 8

print list="\\"," _lntotals(10.0C)," ", PRINTO=1
ENDLOOP ;^End Loop 7

print list="\\"," _supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
endif ;^End Condition 1.
ENDLOOP ;^End Loop 2.

print list=" ", PRINTO=1
ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= "Number of Lanes per Direction ", PRINTO=1
Print list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100 to get
 _fft2=int(_fiter2/100)
 _tafvcheck=0
LOOP _liter5=1,9,1 for current
 LOOP _aiter4= 100000,599999,10000 for current Lanes
 _tafvcheck=_tafvcheck+_volby[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to
total X checking variable.
ENDLOOP ;^End Loop 11.

ENDLOOP ;^End Loop 10.

if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE
 in Loop 9
 print list= _fft2(2.0)," ", PRINTO=1 ; has X>0 continue to report X. Else
 skip FTYPE.
 LOOP _liter4= 1,9,1 for current FTYPE
 _totftat=0
 LOOP _aiter3= 100000,599999,10000 for current Lanes in Loop 12
 _totftat=_totftat+_volby[_aiter3+_fiter2+_liter4] ; current Lanes and FTYPE in order to
 total X checking variable.
 ENDLOOP ;^End Loop 13.

ENDLOOP ;^End Loop 9.

print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1

Print list= "Total Area Types ", PRINTO=1 ;^Begin Loop 9: Cycles through FTYPES
Print list= "two-digit FTYPE.
 LOOP _liter5=1,9,1 for current
 LOOP _aiter4= 100000,599999,10000 for current Lanes in Loop 10
 LOOP _aiter3= 100000,599999,10000 for current Lanes in Loop 12
 LOOP _aiter2=100,9900,100 to get
_LOOP _aiter=100000,599999,10000
_ENDLOOP

 if (_aiter2>_aiter+9999) BREAK ; in Loop 2 to generate Lane total X.
 _lntotiter=_aiter2+_liter3
 _lntotals=_lntotals+_volby[_lntotiter]
ENDLOOP ;^End Loop 8

print list="\\"," _lntotals(10.0C)," ", PRINTO=1
ENDLOOP ;^End Loop 7

print list="\\"," _supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
endif ;^End Condition 1.
ENDLOOP ;^End Loop 2.

print list=" ", PRINTO=1
ENDLOOP ;^End Loop 1.
print list="Totals", PRINTO=1

_supertotal=0
LOOP _liter6=1,9,1
   _lntotals=0
   LOOP _aiter5=100000,599999,100
      and
      _lntotiter=_aiter5+_liter6
      for
      _lntotals=_lntotals+_volby[_lntotiter]
      ENLOOP
   print list="\", "_lntotals(10.0C)," ", PRINTO=1
   _supertotal=_supertotal+_lntotals
   ENLOOP
print list="\", ",_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= "                                                  Single Digit Facility Types ", PRINTO=1
Print list= "AType         1x          2x          3x          4x          5x          6x
7x          8x          9x      Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _a1iter2=100000,599999,100000 ;^Begin Loop 16: Cycles through ATYPE
   _aat1=int(_a1iter2/100000) ; get single digit ATYPE.
   print list= _aat1(1.0),", " , PRINTO=1
   _fttotal=0
   LOOP _f1iter=1000,9900,1000 ;^Begin Loop 17: Cycles through FTYPE
      by 10 to
      _totftlns=0
      LOOP _fiter3=_f1iter,9900,100 ;^Begin Loop 18: Cycles through two-digit FTYPE
         if (_fiter3>_f1iter+999) BREAK ; for current single digit FTYPE in Loop 17.
         LOOP _aiter6=_a1iter2,599999,100000 ;^Begin Loop 19: Cycles through two-digit ATYPE
            if (_aiter6>_a1iter2+99999) BREAK ; for current single digit ATYPE in Loop 16.
            LOOP _liter7=1,9,1 ;^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
               _totftlns=_totftlns+_volby[_aiter6+_fiter3+_liter7] ; in order to generate total X for FTYPE by ATYPE.
               ENLOOP
         ENDLOOP
      ENDLOOP
      _fttotal=_fttotal+_totftlns ;^Generate total X for ATYPE.
   ENDLOOP
print list="\", ",_fttotal(10.0C), PRINTO=1
print list=" ", PRINTO=1

LOOP _aliter2=100000,599999,100000 ;^Begin Loop 16: Cycles through ATYPE
by 10 to
   _aat1=int(_aliter2/100000)
   print list= _aat1(1.0),", " , PRINTO=1
   _fttotal=0
   LOOP _fliter=1000,9900,1000 ;^Begin Loop 17: Cycles through FTYPE
      by 10 to
      _totftlns=0
      LOOP _fiter3=_fliter,9900,100 ;^Begin Loop 18: Cycles through two-digit FTYPE
         if (_fiter3>_fliter+999) BREAK ; for current single digit FTYPE in Loop 17.
         LOOP _a1iter2=_aliter2,599999,100000 ;^Begin Loop 19: Cycles through two-digit ATYPE
            if (_a1iter2>_aliter2+99999) BREAK ; for current single digit ATYPE in Loop 16.
            LOOP _liter7=1,9,1 ;^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
               _totftlns=_totftlns+_volby[_a1iter2+_fiter3+_liter7] ; in order to generate total X for FTYPE by ATYPE.
               ENLOOP
         ENDLOOP
      ENDLOOP
      _fttotal=_fttotal+_totftlns ;^Generate total X for ATYPE.
   ENDLOOP
print list="\", ",_fttotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
ENDLOOP ;^End Loop 16.
Print list="--------------------------------------------------------------------------------------------------------------------------
^Initialize overall total X.
--------------------------------------------------------------------------------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0
LOOP _filter2=1000,9900,1000 by 10
  _ftotals=0
  LOOP _filter4=_filter2,9900,100
     by 1 to if (_filter4>_filter2+999) BREAK
     LOOP _liter8=1,9,1
        LOOP _aiter7=100000,599999,10000
           _ftotiter=_aiter7+_filter4+_liter8
           _ftotals=_ftotals+_volby[_ftotiter]
        ENDLOOP
     ENDLOOP
  ENDLOOP
ENDLOOP
_ENDLOOP
^End Loop 23.
ENDLOOP
^End Loop 22.
_ENDLOOP
^End Loop 21.
_ENDLOOP
^End Loop 20.
_ENDLOOP
^End Loop 19.
_ENDLOOP
^End Loop 18.
_ENDLOOP
^End Loop 17.
_ENDLOOP
^End Loop 16.
_ENDLOOP
^End Loop 15.

print list="\", ",_ftotals(10.0C)," ", PRINTO=1

ENDLOOP
print list="\", ",_supertotal(10.0C), PRINTO=1
print list="","_supertotal+_ftotals", PRINTO=1
print list="","_ftotals(10.0C), PRINTO=1

; END VOLUME REPORT
;************************************************************************

; BEGIN Count REPORT ------------- X = Count on Links w/ Counts
;************************************************************************
Print list=" ", PRINTO=1
Print list="***************************************************************", PRINTO=1

Print list="*

Print list="* Observed Counts on Links with Counts

Print list="* 

Print list="* 

Print list="* 

Print list="* 

Print list="* 

Print list="* 

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1

PRINTO=1
```plaintext
_gat2=int(_aiter/10000)
_avcheck=0

LOOP _achkiter=_aiter,599999,1
    and Facility Types (FTYPE)
    if (_achkiter>_aiter+9999) BREAK
    totals X checking variable.
    _avcheck=_avcheck+_cntby[_achkiter]
ENDLOOP

if (_avcheck>0)
in Loop 2
    skip ATYPE.
_supertotal=0

Print list= "Area Type ",_aat2(2.0), PRINTO=1
Print list= "                                                  Number of Lanes per 
Direction                                               
FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------
------------------------------------------------",
LOOP _fiter=100,9900,100
    _vcheck=0
    LOOP _liter=1,9,1
        _vcheck=_vcheck+_cntby[_aiter+_fiter+_liter]
    ENDLOOP
    if (_vcheck>0)
in Loop 4
        _fft2=int(_fiter/100)
    skip FTYPE.
        print list= _fft2(2.0),"    ", PRINTO=1
        _totvols=0
        LOOP _liter2=1,9,1
            print list="\",_cntby[_aiter+_fiter+_liter2](10.0C)," ", PRINTO=1
            _totvols=_totvols+_cntby[_aiter+_fiter+_liter2]
    ENDLOOP
        print list="\",_totvols(10.0C), PRINTO=1
    endif
    ENDLOOP

Print list= "------------------------------------------------------------------------------
------------------------------------------------",
LOOP _liter3=1,9,1
    _lntotals=0
    LOOP _aiter2=_aiter,599999,100
        _lntotiter=_aiter2+_liter3
        _lntotals=_lntotals+_cntby[_lntotiter]
    ENDLOOP
    print list="\",_lntotals(10.0C)," ", PRINTO=1
```
ENDLOOP

print list="\\"","_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
endif
ENDLOOP

print list=" ", PRINTO=1
ENDLOOP

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------

Print list= "Total Area Types ", PRINTO=1
Print list= "Number of Lanes per Direction ", PRINTO=1
Print list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "---------------------------------------------------------------------
------------------------------------------", PRINTO=1

-------------------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES-----------------

PRINT list= "Total Area Types ", PRINTO=1
PRINT list= "Number of Lanes per Direction ", PRINTO=1
PRINT list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
PRINT list= "---------------------------------------------------------------------
------------------------------------------", PRINTO=1
PRINT list= "Totals", PRINTO=1
_supertotal=0
LOOP _liter6=1,9,1
ENDLOOP

if (_tafvcheck>0)
in Loop 9
  print list= _fft2(2.0)," ", PRINTO=1
skip FTYPE.
ENDLOOP

PRINT list= "Number of Lanes per Direction ", PRINTO=1
PRINT list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
PRINT list= "---------------------------------------------------------------------
------------------------------------------", PRINTO=1
PRINT list= "Totals", PRINTO=1
_supertotal=0
LOOP _liter6=1,9,1
ENDLOOP
LOOP _aiter5=100000,599999,100
and _lnototiter=_aiter5+_lntiter
for _lnototals=_lnototals+_cntby[_lnototiter]
ENDLOOP

_looptotal=_looptotal+_lnototals
ENDLOOP

_LOOP _aiter5=100000,599999,100
and _lnototiter=_aiter5+_lntiter
for _lnototals=_lnototals+_cntby[_lnototiter]
ENDLOOP

_looptotal=_looptotal+_lnototals
ENDLOOP

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= "                                                  Single Digit Facility Types
                                                  1x          2x          3x          4x          5x          6x
                                                  7x          8x          9x      Totals”, PRINTO=1
Print list= "--------------------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1

LOOP _a1iter2=100000,599999,100000                        ;^Begin Loop 16: Cycles through ATYPE
by 10 to _aat1=int(_a1iter2/100000)                              ; get single digit ATYPE.
print list= _aat1(1.0),"x","    ", PRINTO=1

_looptotal=0                                              ;^Initialize total X for all ATYPE
LOOP _f1iter=1000,9900,100                           ;^Begin Loop 17: Cycles through FTYPE
by 10 to _totftlns=0                                         ;^Initialize total X for all FTYPE by all Lanes.
LOOP _fiter3=_f1iter,9900,100                       ;^Begin Loop 18: Cycles through two-digit FTYPE
if (_fiter3>_f1iter+999) BREAK                    ; for current single digit FTYPE in Loop 17.
LOOP _aiter6=_a1iter2,599999,10000                ;^Begin Loop 19: Cycles through two-digit ATYPE
if (_aiter6>_a1iter2+99999) BREAK               ; for current single digit ATYPE in Loop 16.
LOOP _liter7=1,9,1                                      ;^Begin Loop 20: Cycles through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_cntby[_aiter6+_fiter3+_liter7]   ; in order to generate total X for FTYPE by ATYPE.
ENDLOOP                                                 ;^End Loop 20.
ENDLOOP                                           ;^End Loop 19.
ENDLOOP                                             ;^End Loop 18.
_looptotal=_looptotal+_totftlns                         ;^Generate total X for ATYPE.
ENDLOOP

print list="\\"," ",_totftlns(10.0C)," ", PRINTO=1
ENDLOOP

print list="\\"," ",_f1iter(10.0c), PRINTO=1
ENDLOOP

Print list= "--------------------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
supertotal=0
LOOP _f1iter2=1000,9900,1000
  by 10
  _ftotals=0
  LOOP _fiter4=_f1iter2,9900,100
    by 1 to
    if (_fiter4>_f1iter2+999) BREAK
    FTYPE in
      LOOP _liter8=1,9,1
        LOOP _aiter7=100000,599999,1000
          _ftotiter=_aiter7+_fiter4+_liter8
          _ftotals=_ftotals+_cntby[_ftotiter]
        ENDLOOP
      ENDLOOP
    ENDLOOP
  ENDLOOP
_ENDLOOP
_ENDLOOP
_ENDLOOP
supertotal=_supertotal+_ftotals
ENDLOOP
print list="\\", ",_ftotals(10.0C),"," , PRINTO=1
ENDLOOP
print list="\\", ",_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
;;;;;; END COUNT REPORT
;;;;;;=========================================================================
;;;;;; BEGIN VOLUME/COUNT REPORT ------------- X = Volumes over Counts on Links w/ Counts
;;;;;;=========================================================================
Print list=" ", PRINTO=1
Print list="*******************************************************************************************
************************************* Volume over Count Ratios on Links with Counts
*************************************
*******************************************************************************************
Print list=" ", PRINTO=1
Print list="*******************************************************************************************
************************************* 2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------------
*************************************

Loop _alter1=100000,599999,10000
  by 10
  _aat1=int(_alter1/100000)
  print list= "Area Type ",_aat1(1.0),"x Range: ",
    
Loop _alter=_alter1,599999,10000
  if (_alter>_alter1+99999) BREAK
  _aat2=int(_alter/10000)
  _avcheck=0
  LOOP _achkiter=_alter,599999,10000
    and Facility Types (FTYPE)
if (_achkiter>_aiter+9999) BREAK ; for current ATYPE in Loop 2 and totals X checking variable.
_avcheck=_avcheck+_vcntby[_achkiter]
ENDLOOP
if (_avcheck>0) ; for current ATYPE in Loop 2
in Loop 2
skip ATYPE.
_supertotal=0
_supercnts=0
Print list = "Area Type ",_aat2(2.0), PRINTO=1 ;^Header
Print list = "                                                  Number of Lanes per 
Direction ", PRINTO=1
Print list = "FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list = "------------------------------------------------------------------------------
------------------------------------------------", PRINTO=1
LOOP _fiter=100,9900,100 ;^Begin Loop 4: Cycles through FTYPE
_vcheck=0
LOOP _liter=1,9,1 ;^Begin Loop 5: Cycles through Lanes for current 
_vcheck=_vcheck+_vcntby[_aiter+_fiter+_liter] ; FTYPE in Loop 4 and totals X checking variable.
ENDLOOP
if (_vcheck>0) ;^Begin Condition 2: If current FTYPE in Loop 4
_fft2=int(_fiter/100) ; has X>0 continue to report X. Else
print list= _fft2(2.0),"    ", PRINTO=1
_totvols=0
_totcnts=0
LOOP _liter2=1,9,1 ;^Begin Loop 6: Cycles through Lanes to generate ATYPE by FTYPE by Lanes total X.
if (_cntby[_aiter+_fiter+_liter2]>0)
  _links=_volby[_aiter+_fiter+_liter2]/_cntby[_aiter+_fiter+_liter2]
else
  _links=0
endif
print list="\\", "/_links(10.2C)," ", PRINTO=1
_totvols=_totvols+_volby[_aiter+_fiter+_liter2]
_totcnts=_totcnts+_cntby[_aiter+_fiter+_liter2]
_supertotal=_supertotal+_volby[_aiter+_fiter+_liter2]
_supercnts=_supercnts+_cntby[_aiter+_fiter+_liter2]
ENDLOOP
if (_totcnts>0) ;^End Loop 6.
_totvc=_totvols/_totcnts
else
_totvc=0
endif
print list="\\", ",_totvc(10.2C), PRINTO=1
endif ;^End Condition 2.
ENDLOOP
Print list= "------------------------------------------------------------------------------
------------------------------------------------", PRINTO=1
print list= "Totals", PRINTO=1
LOOP _liter3=1,9,1 ;^Begin Loop 7: Cycles through Lanes for current ATYPE in Loop 2.
_lntotals=0
_lncnts=0
ENDLOOP
LOOP _aiter2=_aiter,599999,100                    ;^Begin Loop 8: Cycles through FTYPE
for current ATYPE
  if (_aiter2>_aiter+9999) BREAK
  _lntotiter=_aiter2+_liter3
  _lntotals=_lntotals+_volby[_lntotiter]
  _lncnts=_lncnts+_cntby[_lntotiter]
ENDLOOP ;^End Loop 8

if (_lncnts>0)
  _lnvc=_lntotals/_lncnts
else
  _lnvc=0
endif
print list="\\"," ",_lnvc(10.2C)," ", PRINTO=1
ENDLOOP ;^End Loop 7

if (_supercnts>0)
  _supervc=_supertotal/_supercnts
else
  _supervc=0
endif
print list="\\"," ",_supervc(10.2C), PRINTO=1
print list=" ", PRINTO=1
endif                                                 ;^End Condition 1.
ENDLOOP                                                 ;^End Loop 2.
print list=" ", PRINTO=1
ENDLOOP                                                   ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1                 ;^Header
Print list= "
Number of Lanes per Direction ", PRINTO=1
Print list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "-------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _fiter2=100,9900,100                                 ;^Begin Loop 9: Cycles through FTYPES
to get
  _fft2=int(_fiter2/100)                                  ; two-digit FTYPE.
  _tafvcheck=0                                            ;^Initialize FTYPE X checking variable.
  _tafcnts=0
LOOP _liter5=1,9,1                                      ;^Begin Loop 10: Cycles through Lanes
for current ; FTYPE in Loop 9.
  LOOP _aiter4= 100000,599999,10000                       ;^Begin Loop 11: Cycles through ATYPE
    for _tafvcheck=_tafvcheck+_volby[_alter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to
total X checking variable.
      _tafcnts=_tafcnts+_cntby[_alter4+_fiter2+_liter5]
    ENDLOOP ;^End Loop 11.
  ENDLOOP                                                 ;^End Loop 10.
  if (_tafvcheck>0)                                       ;^Begin Condition 3: If current FTYPE
    in Loop 9
      print list= _fft2(2.0)," ", PRINTO=1               ; has X>0 continue to report X. Else
    skip FTYPE.
    LOOP _liter4= 1,9,1                                  ;^Begin Loop 12: Cycles through Lanes
      for current FTYPE ; in Loop 9.
        _totftat=0
        _totcnts=0
      ENDLOOP
LOOP _aiter3= 100000,599999,10000  
for current_Lanes in Loop 12
  _totftat=_totftat+_volby[_aiter3+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
  _totcnts=_totcnts+_cntby[_aiter3+_fiter2+_liter4]
ENDLOOP ;^End Loop 13.
if (_totcnts>0)
  _totvc=_totftat/_totcnts
else
  _totvc=0
endif
print list="\",_totvc(10.2C)," ", PRINTO=1
ENDLOOP ;^End Loop 12.
if (_tafcnts>0)
  _tafvc=_tafvcheck/_tafcnts
else
  _tafvc=0
endif ;^End Condition 3.
ENDLOOP ;^End Loop 9.

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
_supertotal=0  
_supercnts=0  ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1  ;^Begin Loop 14: Cycles through Lanes.
  _lntotals=0  ;^Initialize total X for Lanes.
  _lncnts=0
  LOOP _aiter5=100000,599999,100  ;^Begin Loop 15: Cycles through ATYPE and _lntotiter=_aiter5+_liter6  ; FTYPE in order to generate total X for _lntotals=_lntotals+_volby[_lntotiter]  ; Lanes.
  _lncnts=_lncnts+_cntby[_lntotiter]
ENDLOOP ;^End Loop 15.
if (_lncnts>0)
  _lnvc=_lntotals/_lncnts
else
  _lnvc=0
endif
_print list="\",_lnvc(10.2C)," ", PRINTO=1
_supertotal= _supertotal+ _lntotals  ;^Generate total X for all ATYPE.
_supercnts= _supercnts+ _lncnts
ENDLOOP ;^End Loop 14.
if (_supercnts>0)
  _supervc=_supertotal/_supercnts
else
  _supervc=0
endif
print list="\",_supervc(10.2C)," ", PRINTO=1
print list=" 

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= " Single Digit Facility Types ", PRINTO=1
Print list= " AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO=1
Print list= "---------------------------------------------", PRINTO=1
LOOP _a1iter2=100000,599999,100000
by 10 to
_aat1=int(_a1iter2/100000)
print list="_aat1(1.0),"","", PRINTO=1

_fftotal=0
_ftcnts=0

LOOP _f1iter=1000,9900,1000
by 10 to
_totftlns=0
all Lanes.
_totftcnts=0

LOOP _fiter3=_f1iter,9900,100
digit FTYPE
if (_fiter3>_f1iter+999) BREAK
Loop 17.

LOOP _a1iter2=_a1iter2,599999,10000
by 10 to
_after6=int(_a1iter2/100000)
if (_after6>_a1iter2+99999) BREAK
Loop 16.

LOOP _liter7=1,9,1
through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_volby[_after6+_fiter3+_liter7] ;^in order to generate total
_X for FTYPE by ATYPE.
_totftcnts=_totftcnts+_cntby[_after6+_fiter3+_liter7]
ENDLOOP
ENDLOOP
ENDLOOP
ENDLOOP
ENDLOOP

if (_totftcnts>0)
_totftvc=_totftlns/_totftcnts
else
_totftvc=0
endif
print list="\"," ",_totftvc(10.2C)," ", PRINTO=1
ENDLOOP

if (_ftcnts>0)
_ftvc=_fttotal/_ftcnts
else
_ftvc=0
endif
print list="\"," ",_ftvc(10.2C), PRINTO=1
ENDLOOP

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1

_supertotal=0
_supercnts=0

LOOP _f1iter2=1000,9900,1000
by 10
_ftotals=0
_ftcnts=0

LOOP _fiter4=_f1iter2,9900,100
by 1 to
if (_fiter4>_fiter2+999) BREAK
FTYPE in
\begin{verbatim}
; Loop 21.
;^Begin Loop 23: Cycles through Lanes.
;^Begin Loop 24: Cycles through ATYPE
; to generate total X by single digit
;^End Loop 24.
;^End Loop 23.
;^End Loop 22.
;^Generate overall total for all single
digit ATYPE
; by all single digit FTYPE.
if (_ftcnts>0)
 _ftvc=_ftotals/_ftcnts
else
 _ftvc=0
endif
print list=", \"",_ftvc(10.2C),\", PRINTO=1
ENDLOOP ;^End Loop 21.
if (_supercnts>0)
 _supervc=_supertotal/_supercnts
else
 _supervc=0
endif
print list=", \"",_supervc(10.2C), PRINTO=1
print list=",\n", PRINTO=1
;************************************************************************
; END VOLUME OVER COUNT REPORT
;************************************************************************
;======================================================================================
; BEGIN VOLUME ON ALL LINKS ------------- X = VOLUME
;======================================================================================
Print list="\n", PRINTO=1
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="*
*
*
Total Volume on All Links (Centroid Connectors
Excluded)
*
*
*******************************************************************************************", PRINTO=1
Print list="", PRINTO=1
Print list="*", PRINTO=1
Print list="", PRINTO=1
Print list="*", PRINTO=1
Print list="", PRINTO=1
;-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES-------------------
LOOP _alter=100000,599999,100000 ;^Begin Loop 1: Cycles through Area
Types (ATYPE) by 10
_aat1=int(_alter/100000)
print list="Area Type \"",_aat1(1.0),\" Range:\",\n", PRINTO=1
ENDLOOP ;^Begin Loop 2: Cycles through ATYPE by
1
if (_alter>_alter+99999) BREAK
_aat2=int(_alter/100000)
_avcheck=0
\end{verbatim}
LOOP _achkiter=_aiter,599999,1
and Facility Types (FTYPE)
if (_achkiter>_aiter+9999) BREAK
  _avcheck=_avcheck+_volall[_achkiter]
ENDLOOP

if (_avcheck>0) in Loop 2
  _supertotal=0
  Print list= "Area Type ",_aat2(2.0), PRINTO=1
  Print list= "                                                  Number of Lanes per 
  Direction                                               
                                                  FType         1           2           3           4           5           6 
                                                  7           8           9       Totals", PRINTO=1
  Print list= "------------------------------------------------------------------------------
  ------------------------------------------------", PRINTO=1
  LOOP _fiter=100,9900,100
    _vcheck=0
    LOOP _liter=1,9,1
    _vcheck=_vcheck+_volall[_aiter+_fiter+_liter]
    ENDLOOP
    if (_vcheck>0 & (_fiter<5000 | _fiter>5999)) in Loop 4
      _fft2=int(_fiter/100)
      print list= _fft2(2.0),"    ", PRINTO=1
      _totvols=0
      LOOP _liter2=1,9,1
        print list="\",_volall[_aiter+_fiter+_liter2](10.0C)," ", PRINTO=1
        _totvols=_totvols+_volall[_aiter+_fiter+_liter2]
        _supertotal=_supertotal+_volall[_aiter+_fiter+_liter2]
      ENDLOOP
      print list="\",_totvols(10.0C), PRINTO=1
    endif
  ENDLOOP
  Print list= "Totals", PRINTO=1
  LOOP _liter3=1,9,1
    _lntotals=0
    LOOP _aiter2=_aiter,599999,100
      _lntotals=0
      LOOP _liter3=1,9,1
        _lntotals=_lntotals+_volall[_lntotiter](10.0C)
      ENDLOOP
      _lntotiter=_aiter2+_liter3
      print list="\",_lntotals(10.0C)," ", PRINTO=1
    ENDLOOP
ENDLOO

LOOP _fiter=100,9900,100
  _vcheck=0
  LOOP _liter=1,9,1
    _vcheck=_vcheck+_volall[_aiter+_fiter+_liter]
  ENDLOOP
  if (_vcheck>0 & (_fiter<5000 | _fiter>5999))
    _fft2=int(_fiter/100)
    print list= _fft2(2.0),"    ", PRINTO=1
    _totvols=0
    LOOP _liter2=1,9,1
      print list="\",_volall[_aiter+_fiter+_liter2](10.0C)," ", PRINTO=1
      _totvols=_totvols+_volall[_aiter+_fiter+_liter2]
      _supertotal=_supertotal+_volall[_aiter+_fiter+_liter2]
    ENDLOOP
    print list="\",_totvols(10.0C), PRINTO=1
  endif
ENDLOOP

Print list= "Totals", PRINTO=1
FROM
  LOOP _liter3=1,9,1
    _lntotals=0
    LOOP _aiter2=_aiter,599999,100
      _lntotals=0
      LOOP _liter3=1,9,1
        _lntotals=_lntotals+_volall[_lntotiter]
      ENDLOOP
      _lntotiter=_aiter2+_liter3
      print list="\",_lntotals(10.0C)," ", PRINTO=1
    ENDLOOP
  ENDLOOP
ENDLOOP
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

print list="\""," \",_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
endif

ENDLOOP

print list=" ", PRINTO=1
ENDLOOP

print list=" ", PRINTO=1
ENDLOOP

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1
Print list= " Number of Lanes per Direction ", PRINTO=1
Print list= "FType  1     2     3     4     5     6     7     8     9     Totals", PRINTO=1

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100
    _fft2=int(_fiter2/100)
    _tafvcheck=0
    if (_fft2<50 | _fft2>59)
        LOOP _liter5=1,9,1
            LOOP _aiter4= 100000,599999,10000
                _tafvcheck=_tafvcheck+_volall[_aiter4+_fiter2+_liter5]
            ENDLOOP
        ENDLOOP
    else
        LOOP _liter5=1,9,1
            LOOP _aiter4= 100000,599999,10000
                _tafvcheck=_tafvcheck+_volall[_aiter4+_fiter2+_liter5]
            ENDLOOP
    endif
ENDLOOP

if (_tafvcheck>0)
    LOOP _liter4= 1,9,1
        LOOP _aiter3= 100000,599999,10000
            _totftat=_totftat+_volall[_aiter3+_fiter2+_liter4]
        ENDLOOP
    ENDLOOP
    print list="\\",_totftat(10.0C)," ", PRINTO=1
ENDLOOP

print list="\\",_tafvcheck(10.0C), PRINTO=1
endif

ENDLOOP

PRINT list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1

PRINT list="Totals", PRINTO=1
_supertotal=0
LOOP _liter6=1,9,1
    _ntotals=0
ENDLOOP

ENDLOOP

ENDLOOP

ENDLOOP

ENDLOOP

ENDLOOP

ENDLOOP

Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

LOOP _aiter5=100000,599999,100 ;^Begin Loop 15: Cycles through ATYPE
and
if ((_aiter5<105000 | _aiter5>105999) &
(_aiter5<115000 | _aiter5>115999) &
(_aiter5<125000 | _aiter5>125999) &
(_aiter5<135000 | _aiter5>135999) &
(_aiter5<145000 | _aiter5>145999) &
(_aiter5<155000 | _aiter5>155999) &
(_aiter5<165000 | _aiter5>165999) &
(_aiter5<175000 | _aiter5>175999) &
(_aiter5<185000 | _aiter5>185999) &
(_aiter5<195000 | _aiter5>195999) &
(_aiter5<205000 | _aiter5>205999) &
(_aiter5<215000 | _aiter5>215999) &
(_aiter5<225000 | _aiter5>225999) &
(_aiter5<235000 | _aiter5>235999) &
(_aiter5<245000 | _aiter5>245999) &
(_aiter5<255000 | _aiter5>255999) &
(_aiter5<265000 | _aiter5>265999) &
(_aiter5<275000 | _aiter5>275999) &
(_aiter5<285000 | _aiter5>285999) &
(_aiter5<295000 | _aiter5>295999) &
(_aiter5<305000 | _aiter5>305999) &
(_aiter5<315000 | _aiter5>315999) &
(_aiter5<325000 | _aiter5>325999) &
(_aiter5<335000 | _aiter5>335999) &
(_aiter5<345000 | _aiter5>345999) &
(_aiter5<355000 | _aiter5>355999) &
(_aiter5<365000 | _aiter5>365999) &
(_aiter5<375000 | _aiter5>375999) &
(_aiter5<385000 | _aiter5>385999) &
(_aiter5<395000 | _aiter5>395999) &
(_aiter5<405000 | _aiter5>405999) &
(_aiter5<415000 | _aiter5>415999) &
(_aiter5<425000 | _aiter5>425999) &
(_aiter5<435000 | _aiter5>435999) &
(_aiter5<445000 | _aiter5>445999) &
(_aiter5<455000 | _aiter5>455999) &
(_aiter5<465000 | _aiter5>465999) &
(_aiter5<475000 | _aiter5>475999) &
(_aiter5<485000 | _aiter5>485999) &
(_aiter5<495000 | _aiter5>495999) &
(_aiter5<505000 | _aiter5>505999) &
(_aiter5<515000 | _aiter5>515999) &
(_aiter5<525000 | _aiter5>525999) &
(_aiter5<535000 | _aiter5>535999) &
(_aiter5<545000 | _aiter5>545999) &
(_aiter5<555000 | _aiter5>555999) &
(_aiter5<565000 | _aiter5>565999) &
(_aiter5<575000 | _aiter5>575999) &
(_aiter5<585000 | _aiter5>585999) &
(_aiter5<595000 | _aiter5>595999))

_for (_lntotiter=_aiter5+_liter6)

_endif
ENDLOOP ;^End Loop 15.

print list="\", "_lntotals(10.0C)," , PRINTO=1

_supertotal=_supertotal+_lntotals ;^Generate total X for all ATYPE.
ENDLOOP ;^End Loop 14.

print list="\", "_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1

;------------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY------------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO-1 ;^Header
Print list = "S single digit facility types", PRINTO=1
Print list = "atype   1x   2x   3x   4x   5x   6x   7x   8x   9x totals", PRINTO=1
Print list = "------------------------------------------------", PRINTO=1

LOOP alter2=100000,599999,100000 by 10 to
   _aatl=int(_alter2/100000)
   print list="_aatl(1.0),"",",", PRINTO=1
   _fttotal=0
   LOOP _filter=1000,9900,1000 by 10 to
      _totftlns=0
      LOOP _fiter3=_filter,9900,100 if (_fiter3>_filter+999) BREAK
      LOOP _aiter6=_alter2,599999,100000 if (_aiter6>_alter2+99999) BREAK
      LOOP _liter7=1,9,1
         LOOP _aiter7=100000,599999,100000 in order
            _ftotiter=_aiter7+_fiter3+_liter7
            _totftlns=_totftlns+_volall[_aiter6+_fiter3+_liter7] ; in order to generate total X for FTYPE by single digit ATYPE.
         ENDLOOP
      ENDLOOP
      _fttotal=_fttotal+_totftlns ;^Generate total X for single digit ATYPE.
   ENDLOOP
ENDLOOP
ENDLOOP
ENDLOOP
ENDLOOP

Print list="\",",",_fttotal(10.0C),",", PRINTO=1
ENDLOOP
ENDLOOP
ENDLOOP

print list="\",",",_fttotal(10.0c), PRINTO=1
ENDLOOP

Print list="------------------------------------------------", PRINTO=1
print list="totals", PRINTO=1

_supertotal=0
LOOP _filter2=1000,9900,1000 by 10
   _ftotals=0
   LOOP _filter4=_filter2,9900,100 by 1 if (_filter4>_filter2+999) BREAK
      LOOP _filter2<5000 | _filter2>5999 Loop 21.
      LOOP _liter8=1,9,1
         LOOP _alter7=100000,599999,100000 in order
            _ftotiter=_alter7+_filter4+_liter8
            _supertotal=_supertotal+_ftotiter
         ENDLOOP
      ENDLOOP
      _ftotals=_ftotals+_ftotiter ;^Initialize overall total X.
   ENDLOOP
ENDLOOP

LOOP _filter2=1000,9900,1000 by 1
   _ftotals=0
   LOOP _filter4=_filter2,9900,100 by 1 if (_filter4>_filter2+999) BREAK
      LOOP _alter7=100000,599999,100000 in order
         LOOP _alter7=100000,599999,100000
            _supertotal=_supertotal+_alter7
         ENDLOOP
      ENDLOOP
   ENDLOOP
ENDLOOP
ENDLOOP

;^Begin Loop 16: Cycles through ATYPE get single digit ATYPE.
;^Initialize total X for all ATYPE.
;^Begin Loop 17: Cycles through FTYPE get single digit FTYPE.
;^Initialize total X for all FTYPE by all Lanes.
;^Begin Loop 18: Cycles through two-digit FTYPE for current single digit FTYPE in Loop 17.
;^Initialize total X for all FTYPE by single digit ATYPE.
;^Begin Loop 19: Cycles through two-digit ATYPE for current single digit ATYPE in Loop 16.
;^Begin Loop 20: Cycles through two-digit ATYPE, FTYPE and Lanes in order to generate total X for FTYPE by single digit ATYPE.
;^End Loop 20.
;^End Loop 19.
;^End Loop 18.
;^Generate total X for single digit ATYPE.
;^End Loop 17.
;^End Loop 16.
;^Begin Loop 21: Cycles through FTYPE to get single digit FTYPE.
;^Initialize total X by FTYPE.
;^Begin Loop 22: Cycles through FTYPE to get all two-digit FTYPE for current FTYPE.
;^Begin Loop 23: Cycles through Lanes.
;^Begin Loop 24: Cycles through ATYPE to generate total X by single digit.
_ftotals=_ftotals+_volall[_ftotiter]
ENDLOOP

ENDLOOP
endif
ENDLOOP
_supertotal=_supertotal+_ftotals
;^Generate overall total for all single
digit ATYPE
print list="\", ",_ftotals(10.0C)," ", PRINTO=1
ENDLOOP
_totalvolumes=_supertotal
print list="\", ",_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
; END VOLUME ON ALL LINKS REPORT
************************************************************************
************************************************************************
; BEGIN VOLUME PERCENTAGES ON ALL LINKS ------------- X = VOLUME
************************************************************************
************************************************************************
Print list=" ", PRINTO=1
Print list=" ", PRINTO=1
print list=" ", PRINTO=1
;************************************************************************
; END VOLUME ON ALL LINKS REPORT
;************************************************************************

LOOP _superl=1,9,1
_supersuper=_supersuper+_volall[_supera+_superf+_superl]/100 ;^Divide by 100 to get
percentages and not ratios in later computations.
ENDLOOP
endif
ENDLOOP
LOOP _a1iter=100000,599999,10000
_print list= "Area Type ",_aat1(1.0),"x Range:",

PRINTO=1
LOOP _aiter=_a1iter,599999,1000

if (_aiter>_a1iter+99999) BREAK

_avecheck=0
LOOP _achkiter=_aiter,599999,1

if (_achkiter>_aiter+9999) BREAK

_avcheck=avcheck+_volall[_achkiter]
ENDLOOP
ENDLOOP
ENDLOOP
LOOP _aliter=100000,599999,10000

if (_aliter>_aliter+99999) BREAK

_avcheck=avcheck+_volall[_aliter]
ENDLOOP
LOOP _achkiter=_aliter,599999,1

if (_achkiter>_aliter+9999) BREAK

_avcheck=avcheck+_volall[_achkiter]
ENDLOOP

END_LOOP_24. ;^End Loop 24.
END_LOOP_23. ;^End Loop 23.
END_LOOP_22. ;^End Loop 22.
END_LOOP_21. ;^End Loop 21.
if (_avcheck>0)
in Loop 2
skip ATYPE.
_supertotal=0
Print list= "Area Type ",_aat2(2.0), PRINTO=1
Print list= "                                                  Number of Lanes per
7     8     9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------
------------------------------------------------
 LOOP _fiter=100,9900,100
         _vcheck=0
         LOOP _liter=1,9,1
         for current
         _vcheck=_vcheck+_volall[_aiter+_fiter+_liter]
         print list= _fft2(2.0)," ", PRINTO=1
         _totvols=0
         LOOP _liter2=1,9,1
         for current
         _lntotals=0
         LOOP _aiter2=_aiter,599999,100
         for current
         if (_aiter2>_aiter+9999) BREAK
         if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
         _lntotiter=_aiter2+_liter3
         _lntotals=_lntotals+_volall[_lntotiter]
         endif
         ENDLOOP
         print list= "\", ",(_lntotals/_supersuper)(10.2C), " ", PRINTO=1
         endif
         ENDLOOP
         ENDLOOP
         print list="\", ",(_totvols/_supersuper)(10.2C), PRINTO=1
         ENDLOOP
         print list="\", ",(_supertotal/_supersuper)(10.2C), PRINTO=1
         endif
         ENDLOOP
LOOP _liter3=1,9,1
         _lntotals=0
LOOP _aiter2=_aiter,599999,100
for current
if (_aiter2>_aiter+9999) BREAK
if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
    _lntotiter=_aiter2+_liter3
    _lntotals=_lntotals+_volall[_lntotiter]
endif
ENDLOOP
print list="\", " ",(_lntotals/_supersuper)(10.2C), " ", PRINTO=1
ENDLOOP
print list="\", " ",(_supertotal/_supersuper)(10.2C), PRINTO=1
endif
ENDLOOP

--- END LOOP 2 ---
<table>
<thead>
<tr>
<th>Number of Lanes per Direction</th>
<th>FType</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td>------------------------------</td>
<td>-------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>--------</td>
</tr>
</tbody>
</table>

---

```
print list="", PRINTO=1
ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------

Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= "Number of Lanes per Direction ", PRINTO=1
Print list= "FType      1   2   3   4   5   6   7   8   9   Totals", PRINTO=1

Print list= "------------------------------------------------------------------------------------

LOOP _fiter2=100,9900,100 to get
   _fft2=int(_fiter2/100)
   _tafvcheck=0
   if (_fft2<50 | _fft2>59)
      LOOP _liter5=1,9,1
         LOOP _aiter4= 100000,599999,10000
            _tafvcheck=_tafvcheck+_volall[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order
to total X checking variable.
      ENDLOOP
   endif
   LOOP _liter4= 1,9,1
      LOOP _aiter3= 100000,599999,10000
         _totftat=0
         LOOP _aiter5=100000,599999,10000
            _totftat=+_valall[_aiter5+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
         ENDLOOP
      endif
   ENDLOOP
   if (_tafvcheck>0)
      LOOP _liter4= 1,9,1
         LOOP _aiter3= 100000,599999,10000
            _totftat=+_valall[_aiter3+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
         ENDLOOP
      print list="\","",(_totftat/supersupersupersupersuper)(10.2C),"", PRINTO=1
   endif
   LOOP _liter6=1,9,1
      LOOP _aiter5=100000,599999,10000
         _intotals=0
         LOOP _aiter5=100000,599999,10000
            _intotals=+_valall[_aiter5<105000 | _aiter5>105999] &
            LOOP _aiter5=105000,105999
               LOOP _aiter5=115000,115999
                  _intotals=+_valall[_aiter5<115000 | _aiter5>115999] &
               endif
            endif
         endif
      endif
   endif
   ENDLOOP
print list="Totals", PRINTO=1
print list="\", PRINTO=1
_supertotal=0
LOOP _liter5=1,9,1
   LOOP _aiter5=100000,599999,10000
      _intotals=0
      LOOP _aiter5=100000,599999,10000
         _intotals=+_valall[_aiter5<105000 | _aiter5>105999] &
         LOOP _aiter5=105000,105999
            LOOP _aiter5=115000,115999
               _intotals=+_valall[_aiter5<115000 | _aiter5>115999] &
            endif
         endif
      endif
   endif
   ENDLOOP
```
### 1-Digit Facility Types by 1-Digit Area Types Summary

#### Total Summary Area Types by Facility Types

<table>
<thead>
<tr>
<th>AType</th>
<th>1x</th>
<th>2x</th>
<th>3x</th>
<th>4x</th>
<th>5x</th>
<th>6x</th>
<th>7x</th>
<th>8x</th>
<th>9x</th>
<th>Totals</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

```

(_aiter5<125000 | _aiter5>125999) &
(_aiter5<135000 | _aiter5>135999) &
(_aiter5<145000 | _aiter5>145999) &
(_aiter5<155000 | _aiter5>155999) &
(_aiter5<165000 | _aiter5>165999) &
(_aiter5<175000 | _aiter5>175999) &
(_aiter5<185000 | _aiter5>185999) &
(_aiter5<195000 | _aiter5>195999) &
(_aiter5<205000 | _aiter5>205999) &
(_aiter5<215000 | _aiter5>215999) &
(_aiter5<225000 | _aiter5>225999) &
(_aiter5<235000 | _aiter5>235999) &
(_aiter5<245000 | _aiter5>245999) &
(_aiter5<255000 | _aiter5>255999) &
(_aiter5<265000 | _aiter5>265999) &
(_aiter5<275000 | _aiter5>275999) &
(_aiter5<285000 | _aiter5>285999) &
(_aiter5<295000 | _aiter5>295999) &
(_aiter5<305000 | _aiter5>305999) &
(_aiter5<315000 | _aiter5>315999) &
(_aiter5<325000 | _aiter5>325999) &
(_aiter5<335000 | _aiter5>335999) &
(_aiter5<345000 | _aiter5>345999) &
(_aiter5<355000 | _aiter5>355999) &
(_aiter5<365000 | _aiter5>365999) &
(_aiter5<375000 | _aiter5>375999) &
(_aiter5<385000 | _aiter5>385999) &
(_aiter5<395000 | _aiter5>395999) &
(_aiter5<405000 | _aiter5>405999) &
(_aiter5<415000 | _aiter5>415999) &
(_aiter5<425000 | _aiter5>425999) &
(_aiter5<435000 | _aiter5>435999) &
(_aiter5<445000 | _aiter5>445999) &
(_aiter5<455000 | _aiter5>455999) &
(_aiter5<465000 | _aiter5>465999) &
(_aiter5<475000 | _aiter5>475999) &
(_aiter5<485000 | _aiter5>485999) &
(_aiter5<495000 | _aiter5>495999) &
(_aiter5<505000 | _aiter5>505999) &
(_aiter5<515000 | _aiter5>515999) &
(_aiter5<525000 | _aiter5>525999) &
(_aiter5<535000 | _aiter5>535999) &
(_aiter5<545000 | _aiter5>545999) &
(_aiter5<555000 | _aiter5>555999) &
(_aiter5<565000 | _aiter5>565999) &
(_aiter5<575000 | _aiter5>575999) &
(_aiter5<585000 | _aiter5>585999) &
(_aiter5<595000 | _aiter5>595999))

_for

_lntotiter=_aiter5+_liter6                          ; FTYPE in order to generate total X
_for
_lntotals=_lntotals+_volall[_lntotiter]              ; Lanes.
_for
endfor
ENDLOOP                                                 ;^End Loop 15.

print list="\",",(_lntotals/_supersuper)(10.2C)," ", PRINTO=1
_supertotal=_supertotal+_lntotals                       ;^Generate total X for all ATYPE.
_for
ENDLOOP                                                 ;^End Loop 14.

print list="\",",(_supertotal/_supersuper)(10.2C), PRINTO=1
print list=" ","n " , PRINTO=1

/-----------------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY-----------------------

Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;"Header
Print list= " Single Digit Facility Types
Print list= " AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO=1
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _aliter2=100000,599999,100000 by 10 to
    _aat1=int(_aliter2/100000)
print list= _aat1(1.0),"x",",", PRINTO=1
    _fttotal=0
    LOOP _filter=1000,9900,1000 by 10 to
        _totftlns=0
        all Lanes.
        if (_filter<5000 | _filter>5999)
            LOOP _fiter3=_filter,9900,100
                if (_fiter3>_filter+999) BREAK
            ENDLOOP
        endif
        LOOP _aiter6=_aliter2,599999,10000
            if (_aiter6>_aliter2+99999) BREAK
        ENDLOOP
    END LOOP
print list="\",",",(_totftlns/_supersuper)(10.2c), PRINTO=1
END LOOP
print list="\",",",(_fttotal/_supersuper)(10.2c), PRINTO=1
END LOOP
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
_LOOP _filter2=1000,9900,1000 by 10
    _fttotals=0
    LOOP _filter4=_filter2,9900,100
        if (_filter4>_filter2+999) BREAK
    END LOOP
    LOOP _liter7=1,9,1
        LOOP _aiter7=100000,599999,10000
            if (_aiter7>_filter2+99999) BREAK
        END LOOP
        LOOP _liter8=1,9,1
            LOOP _liter8=1,9,1
                LOOP _liter7=1,9,1
                    _ftotiter=_aiter7+_filter4+_liter8
                    _ftotals=_ftotals+_volall[_ftotiter]
                END LOOP
            END LOOP
        END LOOP
    END LOOP
print list="\",",",(_ftotals/_supersuper)(10.2c), PRINTO=1
END LOOP
;^End Loop 22.
_supertotal=_supertotal+_ftotals                        ;^Generate overall total for all single
digit ATYPE

print list="\\", "",(_ftotals/_superater) (10.2C)," ", PRINTO=1
ENDLOOP                                                 ;^End Loop 21.

print list="\\", "",(_supertotal/_superater) (10.2C), PRINTO=1
print list=" ", PRINTO=1
************************************************************************
************************************************************************
END VOLUME PERCENTAGES ON ALL LINKS REPORT
************************************************************************
************************************************************************

; BEGIN VMT ALL LINKS REPORT --------------- X = VMT ON ALL LINKS
;--------------------------------------------------------------------------------------
Print list=" ", PRINTO=1
Print
list="*******************************************************************************************
*************************************

*******************************************************************************************
*************************************", PRINTO=1
Print list=" ", PRINTO=1

;-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES--------------------
LOOP _a1iter=100000,599999,10000                         ;^Begin Loop 1: Cycles through Area
Types (ATYPE) by 10
_aat1=int(_a1iter/100000)                               ; in order to get single digit ATYPE.
print list= "Area Type ",_aat1(1.0),"x Range:",
     "\n ", PRINTO=1

LOOP _aiter=_a1iter,599999,10000                        ;^Begin Loop 2: Cycles through ATYPE by
if (_aiter>_a1iter+99999) BREAK                       ; in order to get two-digit ATYPE.

_avcheck=0                                            ;^Initialize ATYPE X checking variable.
LOOP _achkiter=_aiter,599999,1                        ;^Begin Loop 3: Cycles through Lanes
if (_achkiter>_aiter+9999) BREAK                    ; for current ATYPE in Loop 2 and
totals X checking variable.
_avcheck=_avcheck+_vmtall[_achkiter]
ENDLOOP                                               ;^End Loop 3.
if (_avcheck>0)                                       ;^Begin Condition 1: If current ATYPE
; has X>0 continue to report X. Else
 Superintendent=0                                       ;^Initialize ATYPE total X.
Print list= "Area Type ",_aat2(2.0), PRINTO=1       ;^Header
Print list= "                                                  Number of Lanes per
 Direction                                               
FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
======================================================================================
======================================================================================
LOOP _fiter=100,9900,100                            ;^Begin Loop 4: Cycles through FTYPE
; by 1 in order to get two-digit FTYPE.
_vcheck=0                                          ;^Initialize FTYPE X checking variable.

LOOP _liter=1,9,1
for current
_vcheck=_vcheck+_vmtall[_aiter+_fiter+_liter]
checking variable.
ENDLOOP

if (_vcheck>0 & (_fiter<5000 | _fiter>5999))
in Loop 4
_fft2=int(_fiter/100)
skip FTYPE.

print list= _fft2(2.0),"    ", PRINTO=1
_totvols=0

LOOP _liter2=1,9,1
generate ATYPE by FTYPE by Lanes total X.
_print list= _fft2(2.0),"    ", PRINTO=1
_totvols=_totvols+_vmtall[_aiter+_fiter+_liter2]
_supertotal=_supertotal+_vmtall[_aiter+_fiter+_liter2]
ENDLOOP

print list="\",",_totvols(10.0C), PRINTO=1
endif

END LOOP

Print list= "------------------------------------------------------------------------------
------------------------------------------------
Totals
---------------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES---------------------
Print list= "Total Area Types ", PRINTO=1
Print list= "Number of Lanes per Direction
FType    1    2    3    4    5    6
7    8    9    Totals", PRINTO=1
Print list= "------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100
to get
_fft2=int(_fiter2/100)
tafvcheck=0
if (_fft2<50 | _fft2>59)

PRINT list= _fft2(2.0),"    ", PRINTO=1
_totvols=0

LOOP _liter3=1,9,1
for current
_lntotals=0
_LOOP _aiter2=_aiter,599999,100
for current ATYPE
if (_aiter2>_aiter+9999) BREAK
if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
_lntotiter=_aiter2+_liter3
_lntotals=_lntotals+_vmtall[_lntotiter]
endif
ENDLOOP

_print list="\",",_lntotals(10.0C),",", PRINTO=1
END LOOP

print list="\",",_supertotal(10.0C), PRINTO=1
print list="", PRINTO=1
endif

END LOOP
LOOP _liter5=1,9,1
   LOOP _aiter4= 100000,599999,10000
      _tafvcheck = _tafvcheck + vmtall[_aiter4+_fiter2+_liter5]; current Lanes and FTYPE in order to total X checking variable.
   ENDLOOP
   ENDLOOP
   if (_tafvcheck>0)
      print list= "_fft2(2.0)","", PRINTO=1
   endif
   LOOP _liter4= 1,9,1
      _totftat=0
      LOOP _aiter3=100000,599999,10000
         _totftat = _totftat + vmtall[_aiter3+_fiter2+_liter4]; in order to generate total X for FTYPE by Lane for all ATYPE.
      ENDLOOP
      print list="\\","",_totftat(10.0C),"", PRINTO=1
   ENDLOOP
   print list="\\","",_tafvcheck(10.0C), PRINTO=1
endif
ENDLOOP
print list= "------------------------------------------------------------------------------------
------------------------------------------",
_ENDLOOP
_ENDLOOP
print list="Totals",
_ENDLOOP
_supertotal=0
LOOP _liter6=1,9,1
   _lntotals=0
   LOOP _aiter5=100000,599999,100
      if ((_aiter5<105000 | _aiter5>105999) &
         (_aiter5<115000 | _aiter5>115999) &
         (_aiter5<125000 | _aiter5>125999) &
         (_aiter5<135000 | _aiter5>135999) &
         (_aiter5<145000 | _aiter5>145999) &
         (_aiter5<155000 | _aiter5>155999) &
         (_aiter5<165000 | _aiter5>165999) &
         (_aiter5<175000 | _aiter5>175999) &
         (_aiter5<185000 | _aiter5>185999) &
         (_aiter5<195000 | _aiter5>195999) &
         (_aiter5<205000 | _aiter5>205999) &
         (_aiter5<215000 | _aiter5>215999) &
         (_aiter5<225000 | _aiter5>225999) &
         (_aiter5<235000 | _aiter5>235999) &
         (_aiter5<245000 | _aiter5>245999) &
         (_aiter5<255000 | _aiter5>255999) &
         (_aiter5<265000 | _aiter5>265999) &
         (_aiter5<275000 | _aiter5>275999) &
         (_aiter5<285000 | _aiter5>285999) &
         (_aiter5<295000 | _aiter5>295999) &
         (_aiter5<305000 | _aiter5>305999) &
         (_aiter5<315000 | _aiter5>315999) &
         (_aiter5<325000 | _aiter5>325999) &
         (_aiter5<335000 | _aiter5>335999) &
         (_aiter5<345000 | _aiter5>345999) &
         (_aiter5<355000 | _aiter5>355999) &
         (_aiter5<365000 | _aiter5>365999) &
         (_aiter5<375000 | _aiter5>375999) &
         (_aiter5<385000 | _aiter5>385999) &
         (_aiter5<395000 | _aiter5>395999) &
         (_aiter5<405000 | _aiter5>405999) &
         (_aiter5<415000 | _aiter5>415999) &
         (_aiter5<425000 | _aiter5>425999) &
         (_aiter5<435000 | _aiter5>435999) &
         (_aiter5<445000 | _aiter5>445999) &
         (_aiter5<455000 | _aiter5>455999) &
         (_aiter5<465000 | _aiter5>465999) &
         (_aiter5<475000 | _aiter5>475999) &
         (_aiter5<485000 | _aiter5>485999) &
         (_aiter5<495000 | _aiter5>495999) &
         (_aiter5<505000 | _aiter5>505999)) &
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

\(_{\text{aiter5}<325000 \mid _{\text{aiter5}}>325999} \& \)
\(_{\text{aiter5}<335000 \mid _{\text{aiter5}}>335999} \& \)
\(_{\text{aiter5}<345000 \mid _{\text{aiter5}}>345999} \& \)
\(_{\text{aiter5}<355000 \mid _{\text{aiter5}}>355999} \& \)
\(_{\text{aiter5}<365000 \mid _{\text{aiter5}}>365999} \& \)
\(_{\text{aiter5}<375000 \mid _{\text{aiter5}}>375999} \& \)
\(_{\text{aiter5}<385000 \mid _{\text{aiter5}}>385999} \& \)
\(_{\text{aiter5}<395000 \mid _{\text{aiter5}}>395999} \& \)
\(_{\text{aiter5}<405000 \mid _{\text{aiter5}}>405999} \& \)
\(_{\text{aiter5}<415000 \mid _{\text{aiter5}}>415999} \& \)
\(_{\text{aiter5}<425000 \mid _{\text{aiter5}}>425999} \& \)
\(_{\text{aiter5}<435000 \mid _{\text{aiter5}}>435999} \& \)
\(_{\text{aiter5}<445000 \mid _{\text{aiter5}}>445999} \& \)
\(_{\text{aiter5}<455000 \mid _{\text{aiter5}}>455999} \& \)
\(_{\text{aiter5}<465000 \mid _{\text{aiter5}}>465999} \& \)
\(_{\text{aiter5}<475000 \mid _{\text{aiter5}}>475999} \& \)
\(_{\text{aiter5}<485000 \mid _{\text{aiter5}}>485999} \& \)
\(_{\text{aiter5}<495000 \mid _{\text{aiter5}}>495999} \& \)
\(_{\text{aiter5}<505000 \mid _{\text{aiter5}}>505999} \& \)
\(_{\text{aiter5}<515000 \mid _{\text{aiter5}}>515999} \& \)
\(_{\text{aiter5}<525000 \mid _{\text{aiter5}}>525999} \& \)
\(_{\text{aiter5}<535000 \mid _{\text{aiter5}}>535999} \& \)
\(_{\text{aiter5}<545000 \mid _{\text{aiter5}}>545999} \& \)
\(_{\text{aiter5}<555000 \mid _{\text{aiter5}}>555999} \& \)
\(_{\text{aiter5}<565000 \mid _{\text{aiter5}}>565999} \& \)
\(_{\text{aiter5}<575000 \mid _{\text{aiter5}}>575999} \& \)
\(_{\text{aiter5}<585000 \mid _{\text{aiter5}}>585999} \& \)
\(_{\text{aiter5}<595000 \mid _{\text{aiter5}}>595999})

\(_{\text{intotiter}}=_{\text{aiter5}}+_{\text{liter6}} \)
\end{verbatim}

\_intotiter=_\_aiter5+_\_liter6

\for

\_intotals=_\_intotals+_\_vmtall[\_\_intotiter]

\endfor

\ endif

\ ENDLOOP

\ print list="\\"", _\_intotals(10.0C)," ", PRINTO=1

\ _\_supertotal=_\_supertotal+_\_intotals

\ ENDLOOP

\ print list="\\"", _\_supertotal(10.0C), PRINTO=1

\ print list=" \\

\ ;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------

\ print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header

\ print list= " Single Digit Facility Types ", PRINTO=1

\ print list= "AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO=1

\ print list= "-------------------------------------------------------------", PRINTO=1

\ LOOP _\_aiter2=100000,599999,100000

\ by 10 to

\ _\_aat1=int(_\_aiter2/100000)

\ print list= _\_aat1(1.0),"x", " ", PRINTO=1

\ _\_fttotal=0

\ LOOP _\_fiter=1000,9900,1000

\ by 10 to

\ _\_totflns=0

\ all Lanes.

\ if (_\_filter<5000 | _\_filter>5999)

\ LOOP _\_filter3=_\_filter,9900,100

\ digit FTYPE

\ if (_\_filter3>_\_filter+999) BREAK

\ Loop 17.

\ ;^Begin Loop 16: Cycles through ATYPE

\ ; get single digit ATYPE.

\ ;^Begin Loop 17: Cycles through FTYPE

\ ; get single digit FTYPE.

\ ;^Begin Loop 18: Cycles through two-digit FTYPE

\ for current single digit FTYPE in
LOOP _a1iter=aliter2,59999,10000
  if (_a1iter>_a1iter2+99999) BREAK
ENDLOOP

LOOP _liter=1,9,1
  toftlns=totftlns+_vmtall[_a1iter+_fiter+_liter]
ENDLOOP

_LOOP_ _a1iter6=_a1iter2,59999,10000
  if (_a1iter6>_a1iter2+99999) BREAK
ENDLOOP

LOOP _liter7=1,9,1
  toftlns=totftlns+_vmtall[_a1iter6+_fiter3+_liter] in order to generate total X for FTYPE by ATYPE.
ENDLOOP

LOOP _f1iter2=1000,9900,1000
  if (_f1iter2<5000 | _f1iter2>5999) LOOP 21.
    LOOP _liter8=1,9,1
      LOOP _aiter7=100000,599999,10000 in order
        _ftotiter=_aiter7+_fiter4+_liter8
        _ftotals=_ftotals+_vmtall[_ftotiter]
      ENDLOOP
      _ftotals=_ftotals+_vmtall(_ftotiter)
    ENDLOOP
    _supertotal=_supertotal+_ftotals
  ENDLOOP
LOOP _f1iter2=1000,9900,1000 by 10
  _ftotals=0
    LOOP _fiter4=_f1iter2,9900,100
      if (_fiter4>_f1iter2+9999) BREAK
      if (_f1iter2<5000 | _f1iter2>5999) LOOP 21.
        LOOP _aiter7=100000,599999,10000 in order
          _ftotiter=_aiter7+_fiter4+_liter8
          _ftotals=_ftotals+_vmtall[_ftotiter]
        ENDLOOP
        _ftotals=_ftotals+_vmtall(_ftotiter)
      ENDLOOP
      _supertotal=_supertotal+_ftotals
    ENDLOOP
_LOOP_ _a1iter=aliter2,59999,10000
  if (_a1iter>_a1iter2+99999) BREAK
ENDLOOP
_LOOP_ _liter=1,9,1
  toftlns=totftlns+_vmtall[_a1iter6+_fiter3+_liter] in order to generate total X for ATYPE.
ENDLOOP

_PRINTO=1
print list="\"," ",_ftotals(10.0C)," ", PRINTO=1
ENDLOOP
_PRINTO=1
print list="\"," ",_supertotal(10.0c), PRINTO=1
print list= "------------------------------------------------------------------------------------
------------------------------------------"
_PRINTO=1
print list="Totals", PRINTO=1
_PRINTO=1
_supertotal=0
_LOOP_ _f1iter=1000,9900,1000 by 10
  _ftotals=0
    LOOP _fiter4=_f1iter,9900,100
      if (_fiter4>_f1iter+9999) BREAK
      if (_fiter<5000 | _fiter>5999)
        LOOP _liter8=1,9,1
          LOOP _aiter7=100000,599999,10000 in order
            _ftotiter=_aiter7+_fiter4+_liter8
            _ftotals=_ftotals+_vmtall[_ftotiter]
          ENDLOOP
          _ftotals=_ftotals+_vmtall(_ftotiter)
        ENDLOOP
        _supertotal=_supertotal+_ftotals
      ENDLOOP
    ENDLOOP
_ENDPRINTO=1
print list="\"," ",_supertotal(10.0C), PRINTO=1
print list=" 
;************************************************************************
; END VMT ALL LINKS REPORT
;************************************************************************
;======================================================================================
; BEGIN VHT ALL LINKS REPORT ------------- X = VHT ON ALL LINKS
;======================================================================================
_PRINTO=1
print list=" 
;********************************************************************************
; END VMT ALL LINKS REPORT
;********************************************************************************
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="", PRINTO=1
Print list="", PRINTO=1
Print list="VHT on All Links (Centroid Connectors Excluded) ", PRINTO=1
Print list="", PRINTO=1
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="", PRINTO=1

LOOP _a1iter=100000,599999,100000                         ;^Begin Loop 1: Cycles through Area Types (ATYPE) by 10
 _aat1=int(_a1iter/100000)                               ; in order to get single digit ATYPE.
 print list= "Area Type ",_aat1(1.0),"x Range:", 
 \n ", PRINTO=1

LOOP _aiter=_a1iter,599999,10000                        ;^Begin Loop 2: Cycles through ATYPE by 1
if (_aiter>_a1iter+99999) BREAK                       ; in order to get two-digit ATYPE.
 _aat2=int(_aiter/10000)
 _avcheck=0                                            ;^Initialize ATYPE X checking variable.
 LOOP _achkiter=_aiter,599999,1                        ;^Begin Loop 3: Cycles through Lanes and Facility Types (FTYPE)
 if (_achkiter>_aiter+9999) BREAK                    ; for current ATYPE in Loop 2 and totals X checking variable.
 _avcheck=_avcheck+_vhtall[_achkiter]
ENDLOOP                                               ;^End Loop 3.
if (_avcheck>0)                                       ;^Begin Condition 1: If current ATYPE in Loop 2
 ; has X>0 continue to report X. Else skip ATYPE.
 _supertotal=0                                       ;^Initialize ATYPE total X.
Print list= "Area Type ",_aat2(2.0), PRINTO=1       ;^Header Number of Lanes per Direction
Print list= "FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------
------------------------------------------------", PRINTO=1
LOOP _fiter=100,9900,100                            ;^Begin Loop 4: Cycles through FTYPE by 1 in order to get two-digit FTYPE.
 _vcheck=0                                          ;^Initialize FTYPE X checking variable.
 LOOP _liter=1,9,1                                  ;^Begin Loop 5: Cycles through Lanes for current FTYPE in Loop 4 and totals X checking variable.
 _vcheck=_vcheck+_vhtall[_aiter+_fiter+_liter]    ;^End Loop 5.
if (_vcheck>0 & (_fiter<5000 | _fiter>5999))      ;^Begin Condition 2: If current FTYPE has X>0 continue to report X. Else skip FTYPE.
 _fft2=int(_fiter/100)                          ;^Initialize FTYPE total X.
 LOOP _liter2=1,9,1                             ;^Begin Loop 6: Cycles through Lanes to generate ATYPE by FTYPE by Lanes total X.
 _totvols=0                                     ;^Begin Loop 5: Cycles through Lanes
 _supertotal=_supertotal+_vhtall[_aiter+_fiter+_liter2]
ENDLOOP                                            ;^End Loop 6.
ENDLOOP ;^End Loop 6.

   print list="\\", ",_totvols(10.0C), PRINTO=1
endif ;^End Condition 2.
ENDLOOP ;^End Loop 4.

Print list= "---------------------------------------------------------------------------", PRINTO=1

print list="Totals", PRINTO=1

LOOP _liter3=1,9,1 ;^Begin Loop 7: Cycles through Lanes
   _lntotals=0 ;^Initialize Lane total X.

LOOP _aiter2=_aiter,599999,100 ;^Begin Loop 8: Cycles through FTYPE for current ATYPE in Loop 2.
   if (_aiter2>_aiter+9999) BREAK ; in Loop 2 to generate Lane total X.
   if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
      _lntotiter=_aiter2+_liter3
      _lntotals=_lntotals+_vhtall[_lntotiter]
   endif
   ENDLOOP ;^End Loop 8
   print list="\\", ",_lntotals(10.0C), ",", PRINTO=1
 ENDLOOP ;^End Loop 7

print list="\\", ",_supertotal(10.0C), PRINTO=1
print list=" ", PRINTO=1
endif ;^End Condition 1.
ENDLOOP ;^End Loop 2.
ENDLOOP ;^End Loop 1.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= " 
Number of Lanes per Direction 
FType          1         2         3         4         5         6
7             8        9 Totals", PRINTO=1
Print list= "---------------------------------------------------------------------------", PRINTO=1

LOOP _fiter2=100,9900,100 ;^Begin Loop 9: Cycles through FTYPES to get
   _fft2=int(_fiter2/100) ; two-digit FTYPE.
   _tafvcheck=0 ;^Initialize FTYPE X checking variable.
   if (_fft2<50 | _fft2>59)
      LOOP _liter5=1,9,1 ;^Begin Loop 10: Cycles through Lanes for current
         LOOP _aiter4= 100000,599999,10000 ;^Begin Loop 11: Cycles through ATYPE for _tafvcheck=_tafvcheck+_vhtall[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order to total X checking variable.
         ENDLOOP ;^End Loop 11.
      ENDLOOP ;^End Loop 10.
   endif
ENDLOOP ;^End Loop 9.

if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE in Loop 9 has X>0 continue to report X. Else
   skip FTYPE.
   print list= _fft2(2.0)," ", PRINTO=1
ENDLOOP ;^End Loop 6.
LOOP _liter4= 1,9,1 ;^Begin Loop 12: Cycles through Lanes for current FTYPE
   _totftat=0 ;^Initialize FTYPE total X for all ATYPE.
   LOOP _aiter3= 100000,599999,10000 ;^Begin Loop 13: Cycles through ATYPE for current Lanes in Loop 12
      _totftat+_vhtall[_aiter3+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
   ENDLOOP ;^End Loop 13.
   print list="\\", ",_totftat(10.0C)," ", PRINTO=1
   ENDLOOP ;^End Loop 12.
   print list="\\", ",_tafvcheck(10.0C), PRINTO=1
endif ;^End Condition 3.
endif
ENDLOOP ;^End Loop 9.
Print list= "------------------------------------------------------------------------------------
------------------------------------------
print list="Totals", PRINTO=1
_print list="UFan10000", PRINTO=1
_supertotal=0 ;^Initialize all ATYPE total X.
LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.
   _lntotals=0 ;^Initialize total X for Lanes.
   LOOP _aiter5=100000,599999,100 ;^Begin Loop 15: Cycles through ATYPE and
   if {{(_aiter5<105000 | _aiter5>105999) & (\_aiter5<115000 | \_aiter5>115999) & (\_aiter5<125000 | \_aiter5>125999) & (\_aiter5<135000 | \_aiter5>135999) & (\_aiter5<145000 | \_aiter5>145999) & (\_aiter5<155000 | \_aiter5>155999) & (\_aiter5<165000 | \_aiter5>165999) & (\_aiter5<175000 | \_aiter5>175999) & (\_aiter5<185000 | \_aiter5>185999) & (\_aiter5<195000 | \_aiter5>195999) & (\_aiter5<205000 | \_aiter5>205999) & (\_aiter5<215000 | \_aiter5>215999) & (\_aiter5<225000 | \_aiter5>225999) & (\_aiter5<235000 | \_aiter5>235999) & (\_aiter5<245000 | \_aiter5>245999) & (\_aiter5<255000 | \_aiter5>255999) & (\_aiter5<265000 | \_aiter5>265999) & (\_aiter5<275000 | \_aiter5>275999) & (\_aiter5<285000 | \_aiter5>285999) & (\_aiter5<295000 | \_aiter5>295999) & (\_aiter5<305000 | \_aiter5>305999) & (\_aiter5<315000 | \_aiter5>315999) & (\_aiter5<325000 | \_aiter5>325999) & (\_aiter5<335000 | \_aiter5>335999) & (\_aiter5<345000 | \_aiter5>345999) & (\_aiter5<355000 | \_aiter5>355999) & (\_aiter5<365000 | \_aiter5>365999) & (\_aiter5<375000 | \_aiter5>375999) & (\_aiter5<385000 | \_aiter5>385999) & (\_aiter5<395000 | \_aiter5>395999) & (\_aiter5<405000 | \_aiter5>405999) & (\_aiter5<415000 | \_aiter5>415999) & (\_aiter5<425000 | \_aiter5>425999) & (\_aiter5<435000 | \_aiter5>435999) & (\_aiter5<445000 | \_aiter5>445999) & (\_aiter5<455000 | \_aiter5>455999) & (\_aiter5<465000 | \_aiter5>465999) & (\_aiter5<475000 | \_aiter5>475999) & (\_aiter5<485000 | \_aiter5>485999)} &
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

(_aiter5<495000 | _aiter5>495999) &
(_aiter5<505000 | _aiter5>505999) &
(_aiter5<515000 | _aiter5>515999) &
(_aiter5<525000 | _aiter5>525999) &
(_aiter5<535000 | _aiter5>535999) &
(_aiter5<545000 | _aiter5>545999) &
(_aiter5<555000 | _aiter5>555999) &
(_aiter5<565000 | _aiter5>565999) &
(_aiter5<575000 | _aiter5>575999) &
(_aiter5<585000 | _aiter5>585999) &
(_aiter5<595000 | _aiter5>595999) &

_lntotiter=_aiter5+_liter6                          ; FTYPE in order to generate total X
for
_lntotals=_lntotals+_vhtall[_lntotiter]              ; Lanes.

endif
ENDLOOP

print list="\",",_lntotals(10.0C),"", PRINTO=1
_supertotal=_supertotal+_lntotals                       ;^Generate total X for all ATYPE.
ENDLOOP
print list="\",",_supertotal(10.0C), PRINTO=1
print list="","n ", PRINTO=1

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= "Single Digit Facility Types ", PRINTO=1
Print list= "AType         1x          2x          3x          4x          5x          6x
7x          8x          9x      Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------
LOOP _a1iter2=100000,599999,100000                        ;^Begin Loop 16: Cycles through ATYPE
by 10 to
_aat1=int(_a1iter2/100000)                              ; get single digit ATYPE.
print list= _aat1(1.0),"x","    ", PRINTO=1
_ffttotal=0                                              ;^Initialize total X for all ATYPE
LOOP _f1iter=1000,9900,1000                           ;^Begin Loop 17: Cycles through FTYPE
by 10 to
_totftlns=0                                         ;^Initialize total X for all FTYPE by
all Lanes.
if (_f1iter<5000 | _f1iter>5999) LOOP _fiter3=_f1iter,9900,100
if (_fiter3>_f1iter+999) BREAK                    ; for current single digit FTYPE in Loop 17.
LOOP _aiter6=_a1iter2,599999,10000                ;^Begin Loop 18: Cycles through two-
digit ATYPE
if (_aiter6>_a1iter2+99999) BREAK               ; for current single digit ATYPE in Loop 16.
LOOP _liter7=1,9,1

through Lanes for current FTYPE and ATYPE
_X for FTYPE by ATYPE.
ENDLOOP

_ENDLOOP

ENDLOOP

endif

_ffttotal=_fttotal+_totftlns

;^Generate total X for ATYPE.
ENDLOOP ;^End Loop 16.

Print list= "------------------------
------------------------------------------
Totals"
ENDLOOP ;^End Loop 21.

_supertotal=0 ;^Initialize overall total X.
LOOP _f1iter2=1000,9900,1000 by 10
_ftotals=0
LOOP _fiter4=_f1iter2,9900,100
if (_fiter4>_f1iter2+999) BREAK
if (_f1iter2<5000 | _f1iter2>5999) Loop 21.
LOOP _liter8=1,9,1
LOOP _aiter7=100000,599999,10000
_ftotiter=_aiter7+_fiter4+_liter8
_ftotals=_ftotals+_vhtall[_ftotiter]
ENDLOOP
ENDLOOP
_supertotal=_supertotal+_ftotals ;^Generate overall total for all single
digit ATYPE
ENDLOOP ;^End Loop 22.
ENDLOOP ;^End Loop 23.
ENDLOOP ;^End Loop 24.
ENDLOOP ;^End Loop 21.

_print list="",",_supertotal(10.0C),", PRINTO-1
ENDLOOP ;^End Loop 17.

_print list="",",_fttotal(10.0c), PRINTO-1
ENDLOOP ;^End Loop 16.

Print list= "------------------------------------------
Totals", PRINTO-1
_supertotal=0
LOOP _fliter2=1000,9900,1000 by 10
_ftotals=0
LOOP _fiter4=_fliter2,9900,100
if (_fiter4>_fliter2+999) BREAK
if (_fliter2<5000 | _fliter2>5999) Loop 21.
LOOP _liter8=1,9,1
LOOP _aiter7=100000,599999,10000
_ftotiter=_aiter7+_fiter4+_liter8
_FTTYPE=_fiter4<_fiter2+999
if (_fiter2<5000 | _fiter2>5999) Loop 21.
LOOP _liter8=1,9,1
_LOOP _aiter7=100000,599999,10000

Print list="",",_vhtall[_ftotiter]
ENDLOOP
ENDLOOP
endif
ENDLOOP ;^End Loop 22.
_END_LOOP 21.

_supertotal=_supertotal+_ftotals ;^Generate overall total for all single
digit ATYPE
print list="",",_ftotals(10.0C),", PRINTO-1
ENDLOOP ;^End Loop 21.

print list="",",_ftotals(10.0C),", PRINTO-1
ENDLOOP ;^End Loop 16.

print list= "------------------------
------------------------------------------
Totals", PRINTO-1

LOOP _aiter=_a1iter,599999,10000                        ;^Begin Loop 2: Cycles through ATYPE by
1   if (_aiter>_a1iter+99999) BREAK                   ; in order to get two-digit ATYPE.
   _avcheck=0                                           ;^Initialize ATYPE X checking variable.
   _avdist=0

LOOP _achkiter=_aiter,599999,1                        ;^Begin Loop 3: Cycles through Lanes
and Facility Types (FTYPE)                           ; for current ATYPE in Loop 2 and
   if (_achkiter>_aiter+9999) BREAK                    ; totals X checking variable.
   _avcheck=_avcheck+_wffspd[_achkiter]               ;^End Loop 3.
   ENDLOOP                                               ;^Begin Condition 1: If current ATYPE
   if (_avcheck>0)                                       ; has X>0 continue to report X. Else
   _supertotal=0                                       ;^Initialize ATYPE total X.
   _superdist=0

Print list= "Area Type ",_aat2(2.0), PRINTO=1       ;^Header
Print list= "                                                  Number of Lanes per
Direction                                               ", PRINTO=1
Print list= "FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1       ;^Begin Loop 4: Cycles through FTYPE
LOOP _fiter=100,9900,100                            ; by 1 in order to get two-digit FTYPE.
   _vcheck=0                                          ;^Initialize FTYPE X checking variable.
   _vdist=0

LOOP _liter=1,9,1                                  ;^Begin Loop 5: Cycles through Lanes
   _vcheck=_vcheck+_wffspd[_aiter+_fiter+_liter]    ; for current
   ENDLOOP                                            ;^End Loop 5.
   if (_vcheck>0 & (_fiter<5000 | _fiter>5999))      ;^Begin Condition 2: If current FTYPE
   _fft2=int(_fiter/100)                          ; has X>0 continue to report X. Else
   print list= _fft2(2.0),",    ", PRINTO=1          ;^Initialize FTYPE total X.
   _totvols=0                                     ;^Begin Loop 6: Cycles through Lanes to
   _totdist=0                                     ; generate ATYPE by FTYPE by Lanes total X.
   LOOP _liter2=1,9,1                             ;^End Loop 5
   _totvols=_totvols+_wffspd[_aiter+_fiter+_liter2]
   _totdist=_totdist+_dmiles[_aiter+_fiter+_liter2]
   _supertotal=_supertotal+_wffspd[_aiter+_fiter+_liter2]
   _superdist=_superdist+_dmiles[_aiter+_fiter+_liter2]
   _supertotal=0                   ;^End Loop 6
   _totdist=_totdist+" ", PRINTO=1
   _totvols=_totvols+" ", PRINTO=1
endif
endif                                             ;^End Condition 2.
ENDLOOP ;^End Loop 4.

Print list= "------------------------------------------------------------------------------
------------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
LOOP _liter3=1,9,1 ;^Begin Loop 7: Cycles through Lanes
for current ATYPE in Loop 2.
_lntotals=0 ;^Initialize Lane total X.
_lndist=0
LOOP _aiter2=_aiter,599999,100 ;^Begin Loop 8: Cycles through FTYPE
for current ATYPE in Loop 2 to generate Lane total X.
if (_aiter2>_aiter+9999) BREAK
if (_aiter2<_aiter+5000 | _aiter2>_aiter+5999)
   _lntotiter=_aiter2+_liter3
   _lntotals=_lntotals+_wffspd[_lntotiter]
   _lndist=_lndist+_dmiles[_lntotiter]
endif
ENDLOOP ;^End Loop 8
if (_lndist>0)
   _lnspd=_lntotals/_lndist
else
   _lnspd=0
endif
print list="\"," _lnspd(10.2C)", PRINTO=1
ENDLOOP ;^End Loop 7
if (_superdist>0)
   _superspd=_supertotal/_superdist
else
   _superspd=0
endif
print list="\", _superspd(10.2C), PRINTO=1
print list="", PRINTO=1
ENDLOOP ;^End Loop 2.

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES---------------------
Print list= "Total Area Types ", PRINTO=1 ;^Header
Print list= " Number of Lanes per Direction ", PRINTO=1
Print list= "FType 1 2 3 4 5 6 7 8 9 Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _fiter2=100,9900,100 ;^Begin Loop 9: Cycles through FTYPE to get
_gett2=int(_fiter2/100) ; two-digit FTYPE.
_tafvcheck=0 ;^Initialize FTYPE X checking variable.
_tavfdist=0
if (_gett2<50 | _gett2>59)
   LOOP _liter5=1,9,1 ;^Begin Loop 10: Cycles through Lanes for current
   LOOP _aiter4= 100000,599999,10000 ;^Begin Loop 11: Cycles through ATYPE for current
      _tavfcheck= _tavfcheck+_wffspd[_aiter4+_fiter2+_liter5] ; current Lanes and FTYPE in order
total X checking variable.
      _tavfdist=_tavfcheck+_dmiles[_aiter4+_fiter2+_liter5]
   endif
   ENDLOOP ;^End Loop 11
   if (_tavfdist>0)
      _tafvspd=_tavfcheck/_tavfdist
else
   _tafvspd=0
endif
print list="\", _tafvspd(10.2C), PRINTO=1
print list="", PRINTO=1
ENDLOOP ;^End Loop 1.
ENDLOOP ;^End Loop 11.

ENDLOOP ;^End Loop 10.

if (_tafvcheck>0) ;^Begin Condition 3: If current FTYPE in Loop 9 print list= _fft2(2.0),", PRINTO=1 skip FTYPE.

LOOP _liter4= 1,9,1 ;^Begin Loop 12: Cycles through Lanes for current FTYPE
_totftat=0 ;^Initialize FTYPE total X for all ATYPE.
_26708057
_LOOP aiter3= 100000,599999,10000 ;^Begin Loop 13: Cycles through ATYPE for current Lanes in Loop 12
_26708057
_totftatdist=0
LOOP _aiter3= 100000,599999,10000 ;^Begin Loop 13: Cycles through ATYPE for current Lanes in Loop 12
_26708057
_totftatdist=_totftatdist+_wffspd[_aiter3+_fiter2+_liter4] ; in order to generate total X for FTYPE by Lane for all ATYPE.
_26708057
_ENDLOOP ;^End Loop 13.

if (_totftatdist>0) ;^Begin Loop 13: Cycles through ATYPE
_26708057
_totftatspd=_totftat/_totftatdist
_26708057
else
print list="\\",",_totftatspd(10.2C),",", PRINTO=1
ENDLOOP ;^End Loop 12.

if (_tafvdist>0) ;^Begin Condition 3: If current FTYPE
_26708057
_tafvspd=_tafvcheck/_tafvdist
_26708057
else
print list="\\",",_tafvspd(10.2C), PRINTO=1
ENDLOOP ;^End Condition 3.

endif ;^End Loop 9.

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
_supertotal=0 ;^Initialize all ATYPE total X.
_superdist=0
LOOP _liter6=1,9,1 ;^Begin Loop 14: Cycles through Lanes.
_intotals=0
_26708057
_indist=0
LOOP _aiter5=100000,599999,10000 ;^Begin Loop 15: Cycles through ATYPE and

if ((_aiter5<105000 | _aiter5>105999) &
(_aiter5<115000 | _aiter5>115999) &
(_aiter5<125000 | _aiter5>125999) &
(_aiter5<135000 | _aiter5>135999) &
(_aiter5<145000 | _aiter5>145999) &
(_aiter5<155000 | _aiter5>155999) &
(_aiter5<165000 | _aiter5>165999) &
(_aiter5<175000 | _aiter5>175999) &
(_aiter5<185000 | _aiter5>185999) &
(_aiter5<195000 | _aiter5>195999) &
(_aiter5<205000 | _aiter5>205999) &
(_aiter5<215000 | _aiter5>215999) &
(_aiter5<225000 | _aiter5>225999) &
(_aiter5<235000 | _aiter5>235999) &
(_aiter5<245000 | _aiter5>245999) &
(_aiter5<255000 | _aiter5>255999) &

Page | 214
for _lntotiter=_alter5+_liter6 ; FTYPE in order to generate total X

_endif
ENDLOOP ;^End Loop 15.

if (_lndist>0)
  _lnspd=_lntotals/_lndist
else
  _lnspd=0
endif
print list="\",", _lnspd(10.2C),", PRINTO=1
_supertotal=_supertotal+_lntotals ;^Generate total X for all ATYPE.
 Superintendent=_superdist+_lndist
ENDLOOP ;^End Loop 14.
if (_superdist>0)
  _superspd=_superdist+_superdist
else
  _superspd=0
endif
print list="\",", _superspd(10.2C), PRINTO=1
print list="\n", PRINTO=1

;--------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY---------------

Print list="Total Summary Area Types by Facility Types ", PRINTO-1 ;"Header
Print list="   Single Digit Facility Types
", PRINTO-1
Print list=" AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO-1
Print list=" "}
LOOP _a1iter=100000,599999,100000
by 10 to
_aat1=int(_a1iter/100000)
print list= _aat1(1.0),"x","    ", PRINTO=1
_fttotal=0
_ftdist=0
LOOP _filter=1000,9900,1000
by 10 to
_totftlns=0
all Lanes.
_totftlnsdist=0
if (_filter<5000 | _filter>5999)
LOOP _fiter3=_filter,9900,100
if (_fiter3>_filter+999) BREAK
LOOP _aiter6=_a1iter,599999,10000
if (_aiter6>_a1iter+99999) BREAK
LOOP _liter7=1,9,1
_totftlns=_totftlns+_wffspd[_aiter6+_fiter3+_liter7]
_totftlnsdist=_totftlnsdist+_dmiles[_aiter6+_fiter3+_liter7]
ENDLOOP
ENDLOOP
ENDLOOP

if (_totftlnsdist>0)
_totftlnsspd=_totftlns/_totftlnsdist
else
_totftlnsspd=0
endif
print list="\\", ",_totftlnsspd(10.2C)," ", PRINTO=1
ENDLOOP

if (_ftdist>0)
_ftspd=_fttotal/_ftdist
else
_ftspd=0
endif
print list="\\", ",_ftspd(10.2c), PRINTO=1
ENDLOOP

Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
if (_filter4>_filter2+999) BREAK                       ; get all two-digit FTYPE for current
FTYPE in
   _if ( _filter2<5000 | _filter2>59999)
T Loop 21.
LOOP _liter8=1,9,1
   LOOP _aiter7=100000,599999,10000                    ;^Begin Loop 24: Cycles through ATYPE
   _ftotals=_ftotals+_wffspd[_ftotiter]
   _ftdist=_ftdist+_dmiles[_ftotiter]
ENDLOOP
ENDLOOP
   _superdist=_superdist+_ftdist
deft
   _superdist=0
dif (_ftdist>0)
   _ftspd=_ftotals/_ftdist
else
   _ftspd=0
dendif
print list="\"," ",_ftspd(10.2C)," ", PRINTO=1
ENDLOOP
 if (_superdist>0)
   _superspd=_superdist_/superdist
else
   _superspd=0
dendif
_print list="\"," ",_superspd(10.2C)," ", PRINTO=1
;************************************************************************
; END FREE FLOW SPEED REPORT
;************************************************************************
;======================================================================================
; BEGIN CONGESTED SPEED REPORT ------------- X = Congested Speeds
;======================================================================================
Print list=" ", PRINTO=1
Print
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="* * * * Congested Speed (MPH)
Print list="* * * * 
Print list="*******************************************************************************************

Print list=" ", PRINTO=1
Print
Print list="*******************************************************************************************
*************************************", PRINTO=1
Print list="* * * * Congested Speed (MPH)
Print list="* * * * 
Print list="*******************************************************************************************

Print list=" ", PRINTO=1
;-----------------2-DIGIT FACILITY TYPES BY 2-DIGIT AREA TYPES--------------------
LOOP _a1iter=100000,599999,100000                         ;^Begin Loop 1: Cycles through Area
   _aat1=int(_a1iter/100000)                               ; in order to get single digit ATYPE.
   print list= "Area Type ",_aat1(1.0)," Range:";
   "\n ", PRINTO=1
PRINTO=1
LOOP _alter=_a1iter,599999,10000                            ;^Begin Loop 2: Cycles through ATYPE by
   _aat2=int(_alter/10000)                                ; in order to get two-digit ATYPE.
   _avcheck=0                                             ;^Initialize ATYPE X checking variable.
_avdist=0

LOOP _achkiter=_alter,599999,1
    _avcheck=_avcheck+_wcgspd[_achkiter]
ENDLOOP

if (_avcheck>0)
in Loop 2
    skip ATYPE.
ENDLOOP

if (_avcheck>0)
in Loop 2
    skip ATYPE.
ENDLOOP

LOOP _fiter=100,9900,100
    _vcheck=0
    _vdist=0
    LOOP _liter=1,9,1
        _vcheck=_vcheck+_wcgspd[_aiter+_fiter+_liter]
    ENDLOOP

if (_vcheck>0 & (_fiter<5000 | _fiter>5999))
in Loop 4
    _fft2=int(_fiter/100)
    _totvols=0
    _totdist=0
    LOOP _liter2=1,9,1
        if (_dmiles[_aiter+_fiter+_liter2]>0)
            _spdspd=_wcgspd[_aiter+_fiter+_liter2]/_dmiles[_aiter+_fiter+_liter2]
        else
            _spdspd=0
        endif
        print list=_fft2(2.0),", ",_totvols="_totvols+_wcgspd[_aiter+_fiter+_liter2]
        _totdist=_totdist+_dmiles[_aiter+_fiter+_liter2]
        _supertotal=_supertotal+_wcgspd[_aiter+_fiter+_liter2]
        _superdist=_superdist+_dmiles[_aiter+_fiter+_liter2]
    ENDLOOP

    if (_totdist>0)
        _totspd=_totvols/_totdist
    else
        _totspd=0
    endif
    print list="\\", ", _totspd(10.2C),", PRINTO=1
ENDLOOP

print list=" Totals", PRINTO=1
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

LOOP _liter3=1,9,1
for
    _lntotals=0
    _lndist=0
LOOP _aiter2=_aiter,599999,100
for current ATYPE
    if (_aiter2>_aiter+9999) BREAK
    _lntotiter= aiter2+ liter3
    _lntotals= _lntotals+ wcgsdp[_lntotiter]
    _lndist= _lndist+ dmiles[_lntotiter]
ENDLOOP
if (_lndist>0)
    _lnspd= _lntotals/_lndist
else
    _lnspd=0
endif
ENDLOOP

if (_superdist>0)
    _superspd= _supertotal/_superdist
else
    _superspd=0
endif
print list="\",_lnspd(10.2C),",", PRINTO=1
ENDLOOP

ENDLOOP

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1
Print list= "                                                  Number of Lanes per Direction
FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _fiter2=100,9900,100
for
to get
    _fft2=int(_fiter2/100)
    _tafvcheck=0
    _tafvdist=0
if (_fft2<50 | _fft2>59)
LOOP _liter5=1,9,1
for current
    LOOP _aiter4= 100000,599999,10000
for current Lanes and FTYPE in order
to total X checking variable.
    _tafvcheck= _tafvcheck+ wcgsdp[_aiter4+_fiter2+_liter5]
    _tafvdist= _tafvdist+ dmiles[_aiter4+_fiter2+_liter5]
ENDLOOP
ENDLOOP
if (_tafvcheck>0)
ELSE
ENDLOOP
print list="\", PRINTO=1
ENDLOOP

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1
Print list= "                                                  Number of Lanes per Direction
FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _fiter2=100,9900,100
for
to get
    _fft2=int(_fiter2/100)
    _tafvcheck=0
    _tafvdist=0
if (_fft2<50 | _fft2>59)
LOOP _liter5=1,9,1
for current
    LOOP _aiter4= 100000,599999,10000
for current Lanes and FTYPE in order
to total X checking variable.
    _tafvcheck= _tafvcheck+ wcgsdp[_aiter4+_fiter2+_liter5]
    _tafvdist= _tafvdist+ dmiles[_aiter4+_fiter2+_liter5]
ENDLOOP
ENDLOOP
if (_tafvcheck>0)
ELSE
ENDLOOP
print list="\", PRINTO=1
ENDLOOP

;-----------------2-DIGIT FACILITY TYPES BY TOTAL AREA TYPES--------------------
Print list= "Total Area Types ", PRINTO=1
Print list= "                                                  Number of Lanes per Direction
FType         1           2           3           4           5           6
7           8           9       Totals", PRINTO=1
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
LOOP _fiter2=100,9900,100
for
to get
    _fft2=int(_fiter2/100)
    _tafvcheck=0
    _tafvdist=0
if (_fft2<50 | _fft2>59)
LOOP _liter5=1,9,1
for current
    LOOP _aiter4= 100000,599999,10000
for current Lanes and FTYPE in order
to total X checking variable.
    _tafvcheck= _tafvcheck+ wcgsdp[_aiter4+_fiter2+_liter5]
    _tafvdist= _tafvdist+ dmiles[_aiter4+_fiter2+_liter5]
ENDLOOP
ENDLOOP
if (_tafvcheck>0)
print list= _fft2(2.0),"    ", PRINTO=1
  ; has X>0 continue to report X. Else skip FTYPE.
LOOP _liter4= 1,9,1
  for current FTYPE
    _totfftat=0
    _totfftatdist=0
  ENDLOOP

LOOP _aiter3= 100000,599999,10000
  for current Lanes in Loop 12
    _totfftat=_totfftat+_wcgspd[_aiter3+_fiter2+_liter4]
    _totfftatdist=_totfftatdist+_dmiles[_aiter3+_fiter2+_liter4]
  ENDLOOP

if (_totfftatdist>0)
  _totftatspd=_totfftat/_totfftatdist
else
  _totftatspd=0
endif
print list="\",",_totftatspd(10.2C)," ", PRINTO=1
ENDLOOP

if (_tafvdist>0)
  _tafvspd=_tafvcheck/_tafvdist
else
  _tafvspd=0
endif
print list="\",",_tafvspd(10.2C), PRINTO=1
ENDLOOP

if (_tafvdist>0)
  _tafvspd=_tafvcheck/_tafvdist
else
  _tafvspd=0
endif
print list="\",",_tafvspd(10.2C), PRINTO=1
ENDLOOP

Print list= "------------------------------------------------------------------------------------
Totals" , PRINTO=1
print list="Totals", PRINTO=1

_printtotal=0
_printdist=0
LOOP _liter6=1,9,1
  _lntotals=0
  _lndist=0
  LOOP _aiter5=100000,599999,100
    if (_aiter5<105000 | _aiter5>105999) &
      _aiter5<115000 | _aiter5>115999) &
      _aiter5<125000 | _aiter5>125999) &
      _aiter5<135000 | _aiter5>135999) &
      _aiter5<145000 | _aiter5>145999) &
      _aiter5<155000 | _aiter5>155999) &
      _aiter5<165000 | _aiter5>165999) &
      _aiter5<175000 | _aiter5>175999) &
      _aiter5<185000 | _aiter5>185999) &
      _aiter5<195000 | _aiter5>195999) &
      _aiter5<205000 | _aiter5>205999) &
      _aiter5<215000 | _aiter5>215999) &
      _aiter5<225000 | _aiter5>225999) &
      _aiter5<235000 | _aiter5>235999) &
      _aiter5<245000 | _aiter5>245999) &
      _aiter5<255000 | _aiter5>255999) &
      _aiter5<265000 | _aiter5>265999) &
      _aiter5<275000 | _aiter5>275999) &
      _aiter5<285000 | _aiter5>285999) &
      _aiter5<295000 | _aiter5>295999) &
      _aiter5<305000 | _aiter5>305999) &
      _aiter5<315000 | _aiter5>315999) &
  and
Gainesville Urbanized Area Year 2040 Long Range Transportation Plan Update
Technical Report No. 4: 2010 Model Update and Validation

(_aiter5<325000 | _aiter5>325999) &
(_aiter5<335000 | _aiter5>335999) &
(_aiter5<345000 | _aiter5>345999) &
(_aiter5<355000 | _aiter5>355999) &
(_aiter5<365000 | _aiter5>365999) &
(_aiter5<375000 | _aiter5>375999) &
(_aiter5<385000 | _aiter5>385999) &
(_aiter5<395000 | _aiter5>395999) &
(_aiter5<405000 | _aiter5>405999) &
(_aiter5<415000 | _aiter5>415999) &
(_aiter5<425000 | _aiter5>425999) &
(_aiter5<435000 | _aiter5>435999) &
(_aiter5<445000 | _aiter5>445999) &
(_aiter5<455000 | _aiter5>455999) &
(_aiter5<465000 | _aiter5>465999) &
(_aiter5<475000 | _aiter5>475999) &
(_aiter5<485000 | _aiter5>485999) &
(_aiter5<495000 | _aiter5>495999) &
(_aiter5<505000 | _aiter5>505999) &
(_aiter5<515000 | _aiter5>515999) &
(_aiter5<525000 | _aiter5>525999) &
(_aiter5<535000 | _aiter5>535999) &
(_aiter5<545000 | _aiter5>545999) &
(_aiter5<555000 | _aiter5>555999) &
(_aiter5<565000 | _aiter5>565999) &
(_aiter5<575000 | _aiter5>575999) &
(_aiter5<585000 | _aiter5>585999) &
(_aiter5<595000 | _aiter5>595999))

_for

(\_intotiter=\_aiter5+\_liter6) ; FTYPE in order to generate total X
for
(_intotals=\_intotals+wcgspd[\_intotiter]) ; Lanes.
(\_lndist_\_lndist+\_dmiles[\_intotiter])

_endif
ENDLOOP ;^End Loop 15.

if (_lndist>0)
\_lnspd=\_intotals/_lndist
else
\_lnspd=0
endif
print list="\", "\_lnspd(10.2C), "", PRINTO=1
\_supertotal=\_intotals+_lntotals                       ;^Generate total X for all ATYPE.
\_superdist=_lndist+_superdist+
ENDLOOP ;^End Loop 14.

if (_superdist>0)
\_superspd=\_supertotal/_superdist
else
\_superspd=0
endif
print list="\", "\_superspd(10.2C), PRINTO=1
print list="\", "\"n \", PRINTO=1

;---------------1-DIGIT FACILITY TYPES BY 1-DIGIT AREA TYPES SUMMARY----------------
Print list= "Total Summary Area Types by Facility Types ", PRINTO=1 ;^Header
Print list= " Single Digit Facility Types ", PRINTO=1
Print list= " AType 1x 2x 3x 4x 5x 6x 7x 8x 9x Totals", PRINTO=1
Print list= "-------------------------------------------------------------", PRINTO=1
LOOP \_a1iter2=100000, 599999, 100000 ;^Begin Loop 16: Cycles through ATYPE
by 10 to
\_aat1=int(\_a1iter2/100000) ; get single digit ATYPE.
print list= \_aat1(1.0),"x", "", PRINTO=1
\_fttotal=0 ;^Initialize total X for all ATYPE

Page | 221
_ftdist=0
LOOP _f1iter=1000,9900,1000 by 10 to
_aligns=0
all Lanes.
_totftlns=0
if (_f1iter<5000 | _f1iter>5999)
LOOP _fiter3=_f1iter,9900,1000
_digit FTYPE
if (_fiter3>_f1iter+999) BREAK
Loop 17.
LOOP _a1iter6=_a1iter2,599999,1000
_digit ATYPE
if (_a1iter6>_a1iter2+99999) BREAK
Loop 16.
LOOP _liter7=1,9,1
_fttotal=_fttotal+_totftlns
_ftdist=_ftdist+_totftlnsdist
if (_totftlnsdist>0)
_ftlnsspd=_totftlns/_totftlnsdist
else
_ftlnsspd=0
endif
print list="\\", _ftlnsspd(10.2c), PRINTO=1
ENDLOOP
^End Loop 20.
ENDLOOP
^End Loop 19.
ENDLOOP
^End Loop 18.
 ENDLOOP
^End Loop 17.
if (_ftdist>0)
_ftspd=_fttotal/_ftdist
else
_ftspd=0
endif
print list="\\", _ftspd(10.2c), PRINTO=1
ENDLOOP
^End Loop 16.
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
print list="Totals", PRINTO=1
 Superintendent=0
_superuserdist=0
LOOP _f1iter2=1000,9900,1000
by 10
_fttotal=0
_fttotaldist=0
if (_f1iter2<5000 | _f1iter2>5999)
LOOP _fiter4=_f1iter2,9900,1000
by 1 to
_digit FTYPE
if (_fiter4>_f1iter2+999) BREAK
Loop 21.
LOOP _liter8=1,9,1
for current single digit FTYPE in
through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_wcgspd[_a1iter6+_fiter3+_liter7]
_totftlnsdist=_totftlnsdist+_dmiles[_a1iter6+_fiter3+_liter7]
ENDLOOP
^End Loop 20.
ENDLOOP
^End Loop 19.
ENDLOOP
^End Loop 18.
 ENDLOOP
^End Loop 17.
if (_ftdist>0)
_ftspd=_fttotal/_ftdist
else
_ftspd=0
endif
print list="\\", _ftspd(10.2c), PRINTO=1
ENDLOOP
^End Loop 16.
Print list= "------------------------------------------------------------------------------------
------------------------------------------", PRINTO=1
print list="Totals", PRINTO=1
print list="Totals", PRINTO=1
 Superintendent=0
_superuserdist=0
LOOP _f1iter2=1000,9900,1000
by 10
_fttotal=0
_fttotaldist=0
if (_f1iter2<5000 | _f1iter2>5999)
LOOP _fiter4=_f1iter2,9900,1000
by 1 to
_digit FTYPE
if (_fiter4>_f1iter2+999) BREAK
Loop 21.
LOOP _liter8=1,9,1
for current single digit FTYPE in
through Lanes for current FTYPE and ATYPE
_totftlns=_totftlns+_wcgspd[_a1iter6+_fiter3+_liter7]
_totftlnsdist=_totftlnsdist+_dmiles[_a1iter6+_fiter3+_liter7]
ENDLOOP
^End Loop 20.
ENDLOOP
^End Loop 19.
ENDLOOP
^End Loop 18.
LOOP _aiter7=100000,599999,10000 ; Begin Loop 24: Cycles through ATYPE in order
  _ftotitmer=_aiter7+_filter4+_liter8
  _ftotals=_ftotals+_wcgspd[_ftotiter]
  _ftdist=_ftdist+_dmiles[_ftotiter]
ENDLOOP ; End Loop 24.

ENDLOOP ; End Loop 23.

ENDLOOP ; End Loop 22.

if (_ftdist>0)
  _ftspd=_ftotals/_ftdist
else
  _ftspd=0
endif
PRINT list="\","_ftspd(10.2C)", PRINTO=1
ENDLOOP ; End Loop 21.

if (_superdist>0)
  _superspd=_supertotal/_superdist
else
  _superspd=0
endif
_PRINTO=1
print list=" \",_superspd(10.2C), PRINTO=1

ENDLOOP ; End Loop 20.

;************************************************************************
; END CONGESTED SPEED REPORT
;************************************************************************

;======================================================================================
; BEGIN SCREENLINE SUMMARY REPORT ------------- X = SCREENLINE Volume over Count
;======================================================================================

PRINT list="", PRINTO=1
PRINT list="*******************************************************************************************
*************************************", PRINTO=1
PRINT list="*

"Volume over Count

 Screenline Volume over Count

*******************************************************************************************", PRINTO=1
PRINT list="", PRINTO=1

LOOP _sliter=1,99,1
  if (_slvol[_sliter]>0)
    PRINT list= "Screenline ",_sliter(2.0)," Volume/Count Ratio:           ", (_slvol[_sliter]/_slcnt[_sliter])(4.2),
    "Screenline ",_sliter(2.0)," Volume:           ",_slvol[_sliter](10.0C),
    "Screenline ",_sliter(2.0)," Count:           ",_slcnt[_sliter](10.0C), PRINTO=1
  endif
ENDLOOP ;*********************************************************

; BEGIN SUMMARY REPORT
;************************************************************************
Overall Summary

Total Number of Links: 
Total Lane Miles: 
Total Directional Miles: 
Total VMT using Volumes: 
Total VMT using Counts: 
Total VMT Volume over Counts: 
Total VHT using Volumes: 
Total VHT using Counts: 
Total VHT Volume over Counts: 
Total Volumes All Links: 
Total VMT All Links: 
Total VHT All Links: 
Original Speed (MPH): 
Congested Speed (MPH):
Appendix E: Input and Output Network Format

Input Network Format (HNET20{YEAR}.NET)
Note: {YEAR} represents the last two digits of the scenario year. The file name will be HNET2010.NET if it is base year 2010 scenario or HNET2040.NET if it is 2040 Existing plus Committed scenario.

Attribute List for HNET20{YEAR}.NET

Link Attributes
A – A node
B – B node
SCRN – FSUTMS screenline code
DIR – Direction code (0=two-way, 1=one-way)
FTYPE – FSUTMS two-digit facility type. It also should be noted that any link present in the network with FTYPE=0 will not be carried through the model.
ATYPE – FSUTMS two-digit area type
LANES – Directional number of lanes
ROAD_NAME_ – Street name
ROAD_NAME2 – Alternate street name
TYPE – Represents if it is a U.S., state or county road if applicable
RCIFCLASS – RCI functional classification
DISTANCE – Link length in miles
BK_LNS – Bike lanes code (0 = no bike lanes, 1 = in street bike lanes, 2 = wide buffers for biking, 3 = off-street multipurpose facilities)
MOCF – Model output conversion factor that is found from FDOT Traffic Info DVD
AADT07 – Year 2007 two-way average annual daily traffic estimate, only for links where the count was available. For I-75 this is the sum of both directions.
COUNT07 – Directional traffic count with MOCF applied. This is used when VC (volume-to-count ratio) is calculated in the output network.
AADT10 – Year 2010 two-way average annual daily traffic estimate, only for links where the count was available. For I-75 this is the sum of both directions.
COUNT10 – Directional traffic count with MOCF applied. This is used when VC (volume-to-count ratio) is calculated in the output network.

Node Attributes
N – Node number
X – X coordinate
Y – Y coordinate
PNRDESCRIP – Bus park-and-ride lot description (text)
PNRSVCAREA – Maximum park-and-ride service area (highway access distance), in miles.
PARKINGSPA – Number of park-and-ride lot parking spaces. This value is optional because the model does not constrain the auto access mode by the number of spaces.
PNRTERMTIM – Park-and-ride terminal time (walk time from the auto to the bus stop).
KNRTERMTIM – Kiss-and-ride (auto drop-off) terminal time (walk time from the auto to the bus stop).
AMUSEFLAG – Flag to turn the lot on or off for the AM or peak network. If “1”, the lot is used, if “0”, the model ignores the lot.
AMPNRCOST – Cost in cents to park for AM (peak) park-and-ride trips.
MDUSEFLAG – Flag to turn the lot on or off for the MD or off-peak network. If “1”, the lot is used, if “0”, the model ignores the lot.
MDPNRCOST – Cost in cents to park for MD (off-peak) park-and-ride trips

Output Network Format (COMBINEDLOADED.NET)
Note: All the input attributes that were included in the input network of HNET20{YEAR}.NET are carried over to the output network.

Attribute List for COMBINEDLOADED.NET
NONMOTORVOL – Total nonmotorized volumes
CGSPEED – Congested speed
CGTIME – Congested travel time (minutes)
SELZONE_MOTOR – Select zone volumes if ZoneData{YEAR}.DBF included the value of one in the SELECTZONE attribute.
UF_MOTOR – Light plus heavy vehicles with a UF trip end
LIGHTVEHICLES – Total light vehicles
HEAVYTRUCKS – Total heavy trucks
MOTORIZEDVOL – Light vehicles plus heavy trucks (MOTORIZEDVOL is most important because it is directional assigned auto volume that is used for highway evaluation)
VMT – Total motorized vehicle miles of travel.
VHT – Total motorized vehicle hours of travel.
PEDESTRIANS – Pedestrian volumes.
BICYCLISTS – Bike volumes.
VOL_CAP – Motorized volume/ (FSUTMS LOS C capacity)
DAILYCAPE– Daily FSUTMS LOS E capacity
VOL_CAPE – Motorized volume/ (FSUTMS LOS E capacity)
TranVol – Total transit volume (daily persons)
VC – 2010 Volume-to-Count Ratio (VC=MOTORIZEDVOL/COUNT10) This is only available in the base year 2010 scenario.
CONFAC – percentage of daily traffic occurring in the peak hour from VFACTORS.
CAPACITY – Hourly link capacity from the FSUTMS Speed-Capacity table, multiplied by the number of lanes
DAILYCAP – Daily capacity for roadway assignment
SPEED – Free-flow speed from the FSUTMS Speed-Capacity table. If needed, free-flow travel time in minutes can be calculated as: TIME=60*DISTANCE/SPEED
WALKTIME – Travel time in minutes for walk trips at 2.5 miles per hour.
BK_SPD – Bicycle speed
BK_TIME – Bicycle travel time in minutes

Notes Regarding True Shape Display of Networks
HNET20{YEAR}.NET and COMBINEDLOADED.NET in the Gainesville MTPO 2010 can be shown with True Shapefile Display in Cube software to be shown with curved line shapes. Please use True Shape polyline GIS shapefile which is available in model data in the following file location.
…\Gainesville_2007_2035\Media\Street\HNET2010TrueShp.shp

Step 1. Make sure that True Shape GIS shapefile is correctly selected and overlaid underneath the input or output network, as shown in the next screenshot.
Step 2 Go to Tool Bar, and find and select “True Shape Display” function under “GIS Tools”
Step 3 “A” should be selected for A-Node, “B” for B-Node and “N” for Node Number. Click Ok.

Now, the network should be displayed with curved shapes instead of straight line shapes.
Appendix F: Glossary and Abbreviations

(Source: FSUTMS CUBE Comprehensive Workshop Front Matter Documentation – April 2010)

Access – Connectivity between a TAZ and the network. Access can be distinguished between highway and transit networks, and between automobile and pedestrian modes.

Advanced Traffic Management Systems (ATMS) - ATMS is the application of information and telecommunications technologies to the management of freeway and surface street facilities to maximize the use of existing roadway capacity, improve safety, reduce congestion, and provide predictable services.

Advanced Traveler Information Systems (ATIS) - The collection, aggregation and dissemination of information to assist surface transportation travelers in moving from an origin to a destination.

Advanced Public Transportation Systems (APTS) - The use of information and communication technologies to improve the performance of transit services and level of service provided to customers.

Alightings - The number of persons getting off a transit vehicle.

Area Type - Network link code representing the type of land use in the area.

Attraction - The desirability of a zone. For non-home-based trips, attractions in a zone can be considered synonymous with trip destinations in that zone.

Auto Occupancy Rate - Average number of persons per vehicle.

Best Path - One of many paths between a specific origin and destination pair in a transit network determined to be the most efficient means of traveling from the origin to the destination. The default transit path methodology used in Florida.

Boardings - The number of persons getting on a transit vehicle.

Calibration - A process where models are adjusted to simulate trip-making characteristics of households in the model study area to match observed traffic activity in the study area.

Capacity - The maximum number of vehicles that can pass over a given section of a lane or roadway in one direction (or in both directions for a two-lane or three-lane highway). It is the maximum rate of flow that has a reasonable expectation of occurring. The terms “capacity” and “possible capacity” are synonymous. In the absence of a time modifier, capacity is an hourly volume. In expressing capacity, it is essential to state the prevailing roadway and traffic conditions under which the capacity is applicable. The capacity would not normally be exceeded without changing one or more of the conditions that prevail.

Centroid - Centroids are nodes used to identify the center of activity within a traffic analysis zone.

Centroid Connector - The Centroid Connector connects the traffic analysis zone centroid to the surrounding network links.

Cordon Line - An imaginary line encircling a study area. Traffic counts, travel origins and destinations, and other traffic data are collected at the locations where the imaginary line intersects the roads entering and leaving the study area. Used in modeling to estimate traffic entering and exiting the study area.

Commercial Vehicle Operations (CVO) - ITS technologies that uniquely support commercial vehicle operations to promote safe, economical, and efficient truck transportation.

Cube Voyager - A modeling software, developed by Citilabs, used as a modeling engine for the Florida Standard Model.

Demand - A desire for travel from an origin to a destination. Demand is not a fixed amount of travel, but a function of level of service.
Destination - Location to which trips are made, variously identified as a zone of specified area (in aggregate travel forecasting) or a location with a specified “attraction power,” measured by things such as employees (for work trips) or square feet of sales area (for shopping trips).

Desire Line - Lines on a map representing the number of trips between zones. The thicker the line, the larger the number of trips.

EE Trips - External-External trips represent trips that travel through but have both trip ends outside of the model study area.

Facility Type - A network link code representing the type of service a roadway provides, such as principal arterial, minor arterial, collector, etc. The facility type does not always match the functional classification, as the facility type is used for modeling purposes only to simulate actual conditions.

Friction Factors (F-Factors, FF) - Reflects the regional sensitivities toward certain trip lengths for certain trip purposes. For example, home-based shopping trips may tend to be shorter than home-based work trips. Used to modify impedance during trip distribution.

Gravity Model - A mathematical model of trip distribution based on the premise that trips produced in any given area will distribute themselves in accordance with the accessibility of other areas and the opportunities they offer.

Headway - The amount of wait time between arrivals at a given transit stop for a given transit line.

Highway-Only Model - A model that only includes a roadway network thereby excluding transit.

Home-Based Trip - A trip with one end at the residence of the person making the trip.

HOV Trips - High Occupancy Vehicle trips, or carpool trips, represent the number of trips with usually two or more persons in the vehicle, including the driver.

Impedance - More general than Friction Factors, impedance shows the effect that various levels of time and cost will have on travel between zones. Impedance can include various types of time (walking, waiting, riding, etc.) and cost (fares, operating costs, tolls, parking costs, etc.). Other factors, such as comfort, convenience, personal safety, etc., may also be included.

IE Trips - Internal-External trips represent trips that have one end inside the model study area and one end outside the model study area.

II Trips - Internal-Internal trips represent trips that have both ends inside the model study area.

Incident Management Systems - These systems manage both predicted and unexpected incidents so that the impact to the transportation network and traveler safety is minimized. Incident management involves five major phases. These are incident detection, incident verification, incident response, incident clearance, and queue dissipation.

Intelligent Transportation Systems (ITS) - The application of information and telecommunications technologies to the management and operation of transportation systems.

IntelliDrive/Vehicle-Infrastructure Integration (VII) - The establishment of vehicle to vehicle and vehicle to roadside communication capability nationwide to enable a number of new services that provide significant safety, mobility, and commercial benefits.

Intrazonal Trip - A trip with both its origin and destination in the same zone.

Kiss-and-Ride (KNR) - A type of transit trip characterized by a transit rider being dropped off at a transit station by automobile and boarding a transit line.

Level of Service (LOS) - Multidimensional characteristics of the transportation service provided that are usually identified specifically by the location of the origin and destination of a trip and that are divided into those that are quantifiable (travel time, travel cost, number of transfers) and those that are difficult to quantify (comfort, mode image).

Link - A basic component of a network representing a segment of roadway. This component is a primary unit of analysis and carries data pertaining to roadway characteristics, traffic volumes, and performance measures.
Managed Lanes - Managed lanes help maximize the use of existing highway capacity by using price and/or occupancy restrictions to manage the number of vehicles traveling on them. Managed lanes maintain volumes consistent with acceptable levels of service even during peak travel periods.

Micro-coding - A transit modeling technique used to introduce a higher level of detail at transit stations by separating access points between modes and introducing links connecting them. Allows a more realistic representation of transferring between modes.

Mode Choice - Mode choice models calculate which trips will use the highway network and which will use the transit network. The model predicts how the trips will be divided among variable modes of travel.

Mode of Travel - Means of travel such as auto driver, vehicle passenger, mass transit passenger, walking or bicycle.

Nested Logit Model (NLM) - Analytical form for demand modeling that is suited to modeling of multiple travel choice situations by grouping different modes of travel according to their likelihood for direct competition.

Network - Set of nodes and connecting links that represent transportation facilities in an area. Attributes normally associated with links are distances, levels of service, capacities, and volumes.

Node - A point where two links join in a network, usually representing a decision point for route choice but sometimes indicating only a change in some important link attribute.

Occupancy Model - Converts person trips to vehicle trips using auto occupancy factors.

Origin - The location of the beginning of a trip or the zone in which a trip begins.

Park-and-Ride (PNR) - A type of transit trip characterized by the act of parking at a transit station and boarding a transit line.

Path - A set of links representing a possible route between an origin and a destination. There can be a number of paths between any specific origin and destination pair.

Peak Period - The period during which the maximum amount of travel occurs. This may be one or more hours. Generally, there is a morning peak and an afternoon peak and traffic assignments may be made for each period.

Productions - The number of home-based trip ends in the zone of residence. For all non-home based trips, productions are synonymous with origins.

Ramp Metering - The application of signal control devices to regulate the number of and/or how vehicles merge into the freeway mainline lanes with the objective in most cases to balance flow and demand.

Ridership - Number of individuals using a transit line. Used as an assessment of a transit line’s attractiveness.

RMSE - Root Mean Square Error is a measure of total error defined as the square root of the sum of the variance and the square of the bias. It assumes that larger forecast errors are of greater importance than smaller ones; hence they are given a more than proportionate penalty.

Road-Weather Information Systems (RWIS) - RWIS provides information to travelers and also to agencies for better deployment of resources. They use combinations of weather information services and data collected from environmental sensors.

Screenline - An imaginary line, usually along a physical barrier such as a river or railroad tracks, splitting the study area into parts. Traffic counts and possibly interviews are conducted along this line, and the crossings are compared to those calculated from the home interview data as a check of survey accuracy. Crossing may also be compared with model estimates as part of calibration.

Selected Link Analysis - Traces the entire length of each trip passing through a particular link or set of links along the network to determine where such trips are coming from and going to.

Selected Zone Analysis - Traces the entire length of each trip traveling to or from a particular zone or set of zones.
**Shortest Path** - A path representing the least cost option of traveling between any specific origin and destination pair.

**Signal Preemption** - Traffic signal preemption is a type of system that allows the normal operation of traffic lights to be preempted, often to assist emergency vehicles. The most common use of these systems is to provide emergency vehicles priority by changing traffic signals in the path of the vehicle to green and stopping conflicting traffic.

**Smart work zones (SWZ)** - SWZ are automated systems that provide real-time information on work zone traffic conditions. In recent years, transportation agencies across the nation have deployed portable ITS technologies to monitor traffic and manage mobility and safety during construction and maintenance of highways.

**Socioeconomic Data** - Demographic data, such as household, population, and employment characteristics, that are input into the model to determine the impact on trip-making patterns.

**SOV Trips** - Single Occupancy Vehicle trips, or drive-alone trips, represent the number of trips with only one person in the vehicle, including the driver.

**Special Generators** - Concentrations of activities of such size or unusual nature to warrant special consideration in trip generation analysis.

**Station** - A node in the transit network that offers an opportunity for automobile access. FSUTMS

**Stop Node** - A node along a transit line that represents an opportunity for boardings and alightings.

**Study Area Boundary** - The area that is expected to take on urban characteristics in the next 20 to 30 years (by the end of the planning period).

**TAZ** - Traffic Analysis Zone - a small geographic area that serves as the primary unit of analysis in a travel forecasting model.

**Traffic Count** - The observed number of trips collected at a specific location. Used to assist with model validation.

**Transit Legs** - Distinct units of a transit line representing a segment from one stop to the next. Transit paths are built by assessing the relative costs of available transit legs.

**Transit Line** - A collection of transit stops arranged into a route along which public transport vehicles travel. A system of interacting transit lines is a transit network.

**Transit Signal Priority** - Transit Signal Priority (TSP) is an operational strategy that facilitates the movement of in-service transit vehicles through traffic signal controlled intersections. Signal priority modifies the normal signal operation process to better accommodate transit vehicles.

**Transportation Model** - A mathematical description of a transportation system’s characteristics including traffic volumes, and use, roadway type and population. After a mathematical relationship is established, the model is used to predict traffic volumes based on anticipated changes in the other characteristics.

**Trip Assignment** - The process of determining route or routes of travel and allocating the zone-to-zone trips to these routes.

**Trip Distribution** - The process by which the movement of trips between zones is estimated. The data for each distribution may be measured or estimated by a growth factor process, or by synthetic model.

**Trip End** - Either a trip origin or a trip destination.

**Trip Generation** - A general term describing the analysis and application of the relationships that exist among the trip makers, the urban area, and trip making. It is used to determine the number of trip ends in any part of the urban area.

**Trip Purpose** - The reason for making a trip, normally one of several possible purposes. Each trip may have a purpose at each end; (e.g., home to work) or may be classified by the purpose at the non-home end (e.g. home to shop).

**Trip Table** - A table showing trips between zones -- either directionally or total two-way. The
trips may be separated by mode, by purpose, by time period, by vehicle type, or other classification. **Trip Rate** - The average number of trips per household for specific trip purposes. In Florida, trip rates are usually applied by household size and auto availability within each zone by trip purpose. **Validation** - The procedure used to adjust models to simulate base year traffic conditions. A preliminary step that must be undertaken before models may be reasonably used to forecast future traffic conditions. **VHT** - Vehicle hours of travel. **VMT** - Vehicle miles of travel. **Volume-to-Capacity Ratio** - The number of trips simulated in the model divided by the capacity of the link. A volume-to-capacity ratio of 1.0 represents 100 percent of the capacity. **Volume-to-Count Ratio** - The number of trips simulated in the model divided by the count on the link. A volume-to-count ratio of 1.0 represents an exact match between the simulated volumes and the observed counts. Typically assessed only during validation.

**ACRONYMS** (Sourced from FDOT Project Forecasting Handbook 2002)
- ADT Average Daily Traffic
- AADT Annual Average Daily Traffic
- D Directional traffic split
- D30 Proportion of traffic in the peak direction for the 30th highest hour
- DHV Design Hour Volume
- DDHV Directional Design Hour Volume
- DHT Design Hour Truck Percentage
- ESAL Equivalent Single Axle Load
- FDOT Florida Department of Transportation
- FHWA Federal Highway Administration
- FM Financial Management
- FPI Financial Project Identifier
- FSUTMS Florida Standard Urban Transportation Model Structure computer program
- HCM Highway Capacity Manual
- K30 Ratio of DHV to AADT for the 30th highest hour
- Lf Lane Factor
- LGCP Local Government Comprehensive Plan
- LOS Level of Service
- MOCF Model Output Conversion Factor
- MPO Metropolitan Planning Organization
- PD&E Project Development and Environment
- PHF Peak Hour Factor
- PTMS Portable Traffic Monitoring Site
- PSWADT Peak Season Weekday Average Daily Traffic
- RCI Roadway Characteristics Inventory database
- SF Seasonal Factor
- T Truck Factor
- TCI Traffic Characteristics Inventory database
- TTMS Telemetric Traffic Monitoring Site
- V/C Volume to Capacity Ratio
- WPA Work Program Administration
- WPI Work Program Item (First 6 digits of FPI)
Appendix G: Scenario Manager / Running the Model

1. To run the Alachua County model, you first have to open up the Cube Application window by navigating to it or clicking the Cube desktop icon.

2. This brings up a dialog asking for the catalog location of the model.

3. Navigate to the location for the catalog on your computer/network and open it.

   The second way to open the model is to double click Gainesville_2010_2040.cat in Windows Explorer.
4. This brings up the Gainesville model application window.

- **Scenarios (shows base scenario and all siblings)**
- **Applications (shows applications and displays flow)**
- **Data (shows input and output files)**
- **Keys (Shows parameter keys set up by the**
The figure above shows the various parts of the Cube catalog application window.

5. To run a particular scenario, for example the Base Year, you would select **Base 2010 in the Scenarios section of the catalog.**

![Scenarios.png](image.png)

When a scenario is double-clicked, the Scenario dialog box is displayed as illustrated below.

![Scenario Dialog Box.png](image.png)

This dialog box allows you to change model runtime options and set model parameters. These parameters are stored as Cube Catalog Keys which are variables which are referenced in the model script during the relevant model processing phase. The Catalog Key values can also be observed in Part 4 of the Catalog File diagram above.

6. To run the model with the indicated variables, simply press **Run** and then **OK** and the model will be launched. When the model run is completed, press **OK**. You may then analyze the results. The **Applications** section of the Catalog file shows the model processes and may be used to open the Cube Application Flowchart for a specific step or for the overall model.
The **Data** section of the Catalog file shows the input files, output files and any reports generated by the model run processes.

For further details on the Scenario Manager and running FSUTMS models, you may refer to the FSUTMS Comprehensive Modeling Workshops held periodically throughout the year in various locations across the state.
Year 2040 Long Range Transportation Plan Update Planning Team

Metropolitan Transportation Planning Organization for the Gainesville Urbanized Area

Scott R. Koons, AICP, Executive Director
Marlie Sanderson, AICP, Director of Transportation Planning
Michael Escalante, AICP, Senior Planner
Steve Dopp, Senior Planner
Suwan Shen (former employee)
Lauren Yeatter, AICP, Principal Planner
Michael DePalma (former employee)

Atkins

Wiley Page, AICP, Transportation Planning East Manager
Wiatt Bowers, AICP, Senior Transportation Planning Project Manager
Daniel Beaty, AICP, Senior Transportation Planner & Travel Demand Modeling Lead
Philip Shad, AICP, Senior Transportation Planner
Michael Adams, AICP, Senior Transportation Planner
Allison Deffenbaugh, AICP, Transportation Planner
Audrey Postal, Transportation Planner
Todd Flemister, Senior Graphic Designer

HDR Inc.

Santanu Roy, PTP – Florida Transportation Planning Manager & Travel Demand Modeling Practice Lead
Mike Rose – Transit Planner III
Mia Gao, EIT – Travel Demand Modeler
Stephane Gros, PhD – Principal Economist
Hicham Haboussi – Senior Economist
Michael Schmedt, GISP – GIS Manager

Renaissance Planning

Nick Lepp, AICP, CTP, Project Manager
Caleb Van Nostrand, Senior Planner

Quest Corporation of America

Karen Harrell- Business Unit Manager
Micaela Harris- Public Information Associate
Eileen LaSeur- Senior Associate
Dylan Conway- National IT / Web Manager