

# **TTI EMISSIONS INVENTORY ESTIMATION UTILITIES USING MOVES: MOVES2014UTL**

## Technical Report

Prepared for the Texas Department of Transportation

Version FY2018

Updated August 2020

Texas A&M Transportation Institute

**TTI EMISSIONS INVENTORY ESTIMATION UTILITIES USING  
MOVES: MOVES2014aUtl**

Prepared for the  
Texas Department of Transportation  
Transportation Planning and Programming Division

Prepared by the  
Transportation Modeling Program  
Texas A&M Transportation Institute  
TTI Task No.: 409252-0641  
Task Title: Maintain, Update, and Enhance Emissions Analysis Utilities  
(TxDOT Air Quality / Conformity IAC-A)

August 2016

TEXAS A&M TRANSPORTATION INSTITUTE  
The Texas A&M University System  
College Station, Texas 77843

## TABLE OF CONTENTS

<b>LIST OF FIGURES .....</b>	<b>XIV</b>
<b>LIST OF TABLES .....</b>	<b>XVIII</b>
Overview .....	1
Installation .....	2
MOVES2014aUTL Operation.....	3
MOVES2014aUTL Inputs.....	4
MOVES2014aUTL Outputs.....	4
<b>ACTIVITY UTILITIES.....</b>	<b>5</b>
VirtualLinkVMT: HPMS-Based Link VMT and Speeds Estimation .....	7
Operation .....	7
JCF Input File .....	13
Job Identifier (/JOB) .....	14
Step Identifier (/STEP) .....	14
Listing File Output (/LST).....	14
TxDOT RIFCREC Data Input File (/RIFC) .....	15
TxDOT RIFCREC Data Input File for Toll Roads (/TRIFC).....	16
County Name and FIPS Code Input File (/CNTY).....	17
Year and Day Type Input File (/YRDY) .....	17
VMT Control Totals Input File (/CTRL).....	18
Hourly Factors Input File (/HFCT).....	18
Speed Model Data Input File (/SPDM) .....	19
Hourly Link VMT and Operational Speeds Output Files (*.Txx).....	20
Tab-Delimited Summary Output File (*.tab).....	21
TRANSVMT: Travel Model VMT and Speed Estimation Utility.....	23
Operation .....	23
Time-of-Day Volume and VMT Estimation Procedure .....	26
TTI Speed Model .....	28
Houston Speed Model.....	30
Estimating Intrazonal Travel and Speeds .....	33
JCF Input File .....	34
Job Identifier (/JOB) .....	37
Step Identifier (/STEP) .....	37
Listing File Output (/LST).....	37
Header Input File (/HDR).....	38

Hourly Factors Input File (/HRF) .....	38
Centroid Connector Road Type Code Input File (/CCRC).....	39
Directional Split Factors Input File (/SPLT) .....	39
VMT Adjustment Factors Input File (/VADJ).....	40
Intrazonal Radii Input File (/RADI) .....	41
Time Period Designation Input File (/TMPD).....	42
Coordinate Input File (/CRD).....	43
Link Data Fields Input File (/LDF, /LDFA, /LDFM, /LDFP, /LDFO) .....	44
Link Data Input File (/LNK, /LNKA, /LNKM, /LNKP, /LNKO).....	45
Intrazonal Trips Input File (/MTX, /MTXA, /MTXM, /MTXP, /MTXO).....	46
TTI Speed Model Data Input File (/SPDD).....	47
Delay Parameters Input File (/DELP).....	48
Houston Speed Model Data Input File (/SPDH).....	49
Speed Reduction Factor Input File (/SRFD).....	49
Hourly Link VMT and Operational Speeds Output Files (*.Txx).....	51
Node Coordinate Output Files (*_NodeCoordinates.txt) .....	51
Tab-Delimited Summary Output File (*.tab).....	52
Tab-Delimited TDM Summary Output File (/TMSO) .....	52
IntraTripsOutput: Intrazonal Trips Utility.....	53
Operation .....	53
JCF Input File .....	53
Job Identifier (/JOB) .....	54
Step Identifier (/STEP) .....	54
Listing File Output (/LST).....	55
Trip Table Input File (/MTX).....	55
Intrazonal Trips Output File (/IMTX).....	56
RADIICALC: Intrazonal Radii Estimation Utility .....	57
Operation .....	57
JCF Input File .....	58
Job Identifier (/JOB) .....	59
Step Identifier (/STEP) .....	59
Listing File Output (/LST).....	59
Centroid Connector Road Type Code Input File (/CCRC).....	59
Link Data Fields Input File (/LDF).....	60
Link Data Input File (/LNK).....	61
VMTtot: Travel Model VMT Summary Utility .....	63
Operation .....	63

JCF Input File .....	64
Job Identifier (/JOB) .....	66
Step Identifier (/STEP) .....	66
Listing File Output (/LST) .....	67
Header Input File (/HDR) .....	67
Centroid Connector Road Type Code Input File (/CCRC).....	67
Intrazonal Radii Input File (/RADI) .....	67
Link Data Fields Input File (/LDF, /LDFA, /LDFM, /LDFP, /LDFO) .....	68
Link Data Input File (/LNK, /LNKA, /LNKM, /LNKP, /LNKO).....	69
Intrazonal Trips Input File (/MTX, /MTXA, /MTXM, /MTXP, /MTXO).....	70
VehPopulationBuild: Vehicle Population Builder .....	73
Operation .....	73
JCF Input File .....	78
Job Identifier (/JOB) .....	80
Step Identifier (/STEP) .....	80
Listing File Output (/LST).....	80
County Name Input File (/CNTY).....	81
Age Registrations Input File (/AGEI).....	81
Diesel Heavy-Duty Registrations Input File (/DSLJ) .....	82
Gas Heavy-Duty Registrations Input File (/GASI).....	82
TxDOT District Name Input File (/DISTI).....	83
VMT Mix Input File (/VMXI).....	84
MOVES Default Database (/MVSDB).....	86
Population Factor Input File (/POPFI).....	86
Year ID Input File (/YRID) .....	86
MySQL Connection Inputs File (/MYSQL).....	87
Registration Data Summary Output File (/REGDO) .....	87
Tab-Delimited Population by SUT/Fuel Type Output File (/TPFTO) .....	87
sourcetypeyear Database Table Output (/DSTYO) .....	87
sourcetypeyear Tab-Delimited Output (/TSTYO).....	88
OffNetActCalc: Off-Network Activity Builder .....	89
Operation .....	89
JCF Input File .....	94
Job Identifier (/JOB) .....	97
Step Identifier (/STEP) .....	97
Listing File Output (/LST).....	97
County Name Input File (/CTY).....	97

Analysis VMT Road Type Designations Input File (/RDTD).....	98
Analysis 24-Hour VMT Mix Input File (/VMX).....	99
Time Period Designation Input File (/TMPD).....	100
Analysis AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/VMXA, /VMXM, /VMXP, /VMXO) .....	101
Analysis Vehicle Population Input File (/POPI).....	101
Analysis VMT Listing Input File (/VMT).....	102
Hotelling Factor Input File (/HOTF) .....	103
Base Hotelling Hours Input File (/BHHR) .....	103
Base VMT Listing Input File (/BVMT).....	104
Base VMT Road Type Designations Input File (/BRDTD) .....	104
Base 24-Hour VMT Mix Input File (/BVMX) .....	104
Base AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/BVMXA, /BVMXM, /BVMXP, /BVMXO).....	105
Age Distribution Input File (/AGED).....	105
Analysis Year Input (/YEAR).....	106
Hotelling Activity Distribuiton Input File (/HADI).....	106
Relative Mileage Accumulation Rates Input File (/RMAR) .....	107
Starts per Vehicle Input File (/STRTI) .....	108
Tab-Delimited Hourly Summary Header Input File (/HRSH) .....	108
Tab-Delimited Hourly Summary Output File (/HRSO) .....	108
Tab-Delimited SHP Output File (/SHPO) .....	109
Tab-Delimited SHI and APU Hours Output File (/EXIO) .....	109
Tab-Delimited Starts Output File (/STRTO) .....	109
OffNetActSummary: Off-Network Activity Summary.....	111
Operation .....	111
JCF Input File .....	113
Job Identifier (/JOB) .....	114
Step Identifier (/STEP) .....	114
Listing File Output (/LST).....	114
Vehicle Population Input File (/POPI).....	115
SHP Input File (/SHPI).....	115
Starts Input File (/SRTI) .....	116
Hotelling/SHI/APU Hours Input File (/HEAI).....	117
Header Input File (/HDRO) .....	117
24-Hour Summary Output File (/TABO) .....	118
Hourly Summary Output File (/HTABO).....	118
24-Hour Vehicle Type Summary Output File (/VTABO).....	118

Hourly Vehicle Type Summary Output File (/ATABO) ..... 118

VMTmixBuild: VMT Mix Builder .....	119
Operation .....	119
JCF Input File .....	121
Job Identifier (/JOB) .....	122
Step Identifier (/STEP) .....	122
Listing File Output (/LST) .....	123
VMT Mix Input File (/VMXI) .....	123
TxDOT District Name Input File (/DISTI) .....	125
Comments Input File (/CMI) .....	125
VMT Mix Output File (/VMXO) .....	125
<b>EMISSIONS RATES UTILITIES .....</b>	<b>127</b>
MOVESfleetInputBuild: MOVES Fleet Input Builder .....	129
Operation .....	129
JCF Input File .....	136
Job Identifier (/JOB) .....	138
Step Identifier (/STEP) .....	138
Listing File Output (/LST) .....	138
sourcetypeagedistribution Data Sources Input File (/STAI) .....	139
fuelengfraction Data Sources Input File (/FEFI) .....	141
Age Registrations Input File (/AGEI) .....	142
Diesel Heavy-Duty Registrations Input File (/DSLJ) .....	142
Gas Heavy-Duty Registrations Input File (/GASI) .....	143
MOVES Default Database (/MVSDB) .....	143
Default Fuel Engine Fractions Input File (/DFEFI) .....	144
Default sourcetypeagedistribution Input File (/DFSTA) .....	144
Year ID Input File (/YRID) .....	145
VMT Mix Input File (/VMX) .....	145
MySQL Connection Inputs File (/MYSQL) .....	146
sourcetypeagedistribution Database Table Output (/DSTAO) .....	147
sourcetypeagedistribution Tab-Delimited Output (/TSTAO) .....	147
fuelengfraction Database Table Output (/DFEFO) .....	147
fuelengfraction Tab-Delimited Output (/TFEFO) .....	147
MOVESoutSummary: MOVES Output Summary .....	149
Operation .....	149
JCF Input File .....	151
Job Identifier (/JOB) .....	152
Step Identifier (/STEP) .....	153



Listing File Output (/LST).....	153
MySQL Connection Inputs File (/MYSQL).....	153
Database Listing Input File (/DBASE).....	154
MOVES Run Summary Output File (/MVRO).....	154
Pollutant/Process/SUT Number of Records Output File (/PPSO).....	154
Pollutant/Process/SUT/Fuel Type/Road Type Number of Records Output File (/PPSFRO).....	155
RATESCALC: MOVES Emissions Rates Calculation Utility.....	157
Operation.....	158
JCF Input File.....	162
Job Identifier (/JOB).....	163
Step Identifier (/STEP).....	163
Listing File Output (/LST).....	163
MySQL Connection Inputs File (/MYSQL).....	164
Pollutants Input File (/POLL).....	164
MOVES Rate Output Database Name (/MRODB).....	165
rateperdistance Output Flag (/RPDOF).....	165
rateperstart Output Flag (/RPSOF).....	165
rateperhour Output Flag (/RPHOF).....	165
Evaporative Rate Calculation Output Flag (/ERCOF).....	166
SHP Calculation Database Tables Input File (/DBTIN).....	166
SHP Calculation Output Database Flag (/KCDBF).....	167
Output Database Name (/OUTDB).....	167
Tab-Delimited MOVES Run Summary Output File (/MVRO).....	168
Tab-Delimited Emission Rates Summary Output File (/MMRRO).....	168
Tab-Delimited Activity Summary Output File (/MMRAO).....	168
RATESADJ: Emissions Rates Adjustment.....	169
Operation.....	169
JCF Input File.....	172
Job Identifier (/JOB).....	173
Step Identifier (/STEP).....	174
Listing File Output (/LST).....	174
MySQL Connection Inputs File (/MYSQL).....	175
Emissions Rate Database/Table Input File (/DBTIN).....	175
Pollutants Input File (/POLL).....	176
Emissions Rate Adjustments Input File (/DBTA).....	177
Emissions Rate Output Database (/DRATO).....	179
Emissions Rate Min/Max Summary Output File (/RATMM).....	179

<b>EMISSIONS UTILITIES.....</b>	<b>181</b>
EmsCalc: MOVES-Based Emissions Calculation .....	183
Operation .....	183
JCF Input File .....	191
Job Identifier (/JOB) .....	193
Step Identifier (/STEP) .....	193
Listing File Output (/LST) .....	194
Header Input File (/THDR).....	194
County Name Input File (/CTY).....	195
VMT Listing Input File (/VMT).....	195
VMT Road Type Codes Input File (/VMTRD) .....	195
VMT Road Type Designations Input File (/RDTD).....	196
Pollutants/Tab-Delimited Output Units Input File (/POLL).....	197
24-Hour VMT Mix Input File (/VMX).....	198
Time Period Designation Input File (/TMPD).....	199
AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/VMXA, /VMXM, /VMXP, /VMXO) .....	200
ttirateperdistance Emissions Rates Database (/RDIST).....	200
ttirateperstart Emissions Rates Database (/RSTRT) .....	201
ttirateperhour Emissions Rates Database (/RHOURL).....	201
ttiratepershp Emissions Rates Database (/RSHP).....	201
SHP by SUT/Fuel Type Input File (/SHPI) .....	201
Starts by SUT/Fuel Type Input File (/STRTI).....	202
SHI and APU Hours by SUT/Fuel Type Input File (/EIAI) .....	203
MySQL Connection Inputs File (/MYSQL).....	204
SCC Input File (/SCCI).....	205
SCC Pollutants Input File (/SCCP).....	206
Tab-Delimited Output File (/TOUT) .....	207
Hourly Link Output Files Listing Input File (/LNKO).....	207
Activity and Emissions by SCC Output File (/SCCO).....	208
LinkEmsSum: Link Emissions Summary .....	209
Operation .....	209
JCF Input File .....	212
Job Identifier (/JOB) .....	214
Step Identifier (/STEP) .....	214
Listing File Output (/LST) .....	214
Header Input File (/THDR).....	214
VMT Road Type Codes Input File (/VMTRD) .....	215

Hourly Link Emissions Listing Input File (/LNKI).....	215
Tab-Delimited Input File (/TABI) .....	216
Output Units Input File (/UNIT).....	216
Tab-Delimited Output File (/TOUT) .....	217
Tab-Delimited Emissions Comparison Output File (/TCOUT).....	217
Tab-Delimited Totals, Minimum, and Maximum Comparison Summary File (/TMOUT) .....	217
Tab-Delimited 24-hour Totals, Minimum, and Maximum Comparison Summary Output File (/TDOUT).....	218
TabFileTotals: EmsCalc Tab-Delimited Output Summary.....	219
Operation .....	219
JCF Input File .....	222
Job Identifier (/JOB) .....	223
Step Identifier (/STEP) .....	223
Listing File Output (/LST).....	223
Tab-Delimited Input File Name with Wildcard (/TABI).....	224
Header Input File (/HDRO) .....	224
SUT Tab-Delimited Output File (/SOUT).....	225
Totals Tab-Delimited Output File (/TOUT) .....	225
Hourly Totals Tab-Delimited Output File (/THOUT).....	226
Hourly SUT Tab-Delimited Output File (/SHOUT).....	227
Hourly Road Type Totals Tab-Delimited Output File (/TROUT).....	228
Hourly SUT/Road Type Totals Tab-Delimited Output File (/SROUT).....	228
24-hour SUT/Road Type Totals Tab-Delimited Output File (/DSROUT).....	229
<b>MISCELLANEOUS UTILITIES.....</b>	<b>231</b>
MOVESactivityInputBuild: MOVES Activity Input Builder.....	233
Operation .....	233
JCF Input File .....	242
Job Identifier (/JOB) .....	245
Step Identifier (/STEP) .....	245
Listing File Output (/LST).....	246
County Name Input File (/CTY).....	246
VMT Listing Input File (/VMT).....	247
VMT Road Type Designations Input File (/RDTD).....	247
24-Hour VMT Mix Input File (/VMX).....	248
Time Period Designation Input File (/TMPD).....	249
AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/VMXA, /VMXM, /VMXP, /VMXO) .....	250

Day ID Input (/DAYID).....	251
Year ID Input (/YRID).....	251
Month ID Input (/MONID).....	251
VMT Road Type/Ramp Designation Input File (/RMPRD).....	251
MOVES Default Database (/MVSDB).....	252
MySQL Connection Inputs File (/MYSQL).....	252
FIPS (MOVES countyID) Input (/FIPS).....	252
sourcetypeyear Input File (/STYI).....	253
sourcetypeagedistribution Input File (/STADI).....	253
sourcetypeage Input File (/STAI).....	254
Starts Input File (/STRTI).....	254
Hotelling Input File (/HOTI).....	254
roadtypedistribution Tab-Delimited Output (/TRDTD).....	255
hourvmtfraction Tab-Delimited Output (/THRVF).....	255
avgspeeddistribution Tab-Delimited Output (/TASPD).....	255
roadtype Tab-Delimited Output (/TRDTT).....	255
Tab-Delimited Link VMT Summary Output File (/TLVMT).....	255
Tab-Delimited Link VHT Summary Output File (/TLVHT).....	255
year Tab-Delimited Output (/TYEAR).....	255
state Tab-Delimited Output (/TSTAT).....	255
zone Tab-Delimited Output (/TZONE).....	256
zoneroadtype Tab-Delimited Output (/TZRDT).....	256
monthvmtfraction Tab-Delimited Output (/TMVF).....	256
dayvmtfraction Tab-DelimitedOutput (/TDVF).....	256
hpmsvtypeday Tab-DelimitedOutput (/THVDY).....	256
sourcetypeedayvmt Tab-DelimitedOutput (/TSTDV).....	256
sourcetypeage Tab-DelimitedOutput (/TSTAO).....	256
starts Tab-DelimitedOutput (/STRTO).....	256
Hotelling Hours Tab-DelimitedOutput (/HHTO).....	257
MOVESmetInputBuild: MOVES Meteorology Input Builder.....	259
Operation.....	259
JCF Input File.....	262
Job Identifier (/JOB).....	264
Step Identifier (/STEP).....	264
Listing File Output (/LST).....	264
County Name Input File (/CTY).....	264
Month ID Input File (/MONID).....	265

MOBILE6 Input File (/M6IN) .....	265
Meteorological Data Input File (/METIN) .....	265
MOVES Default Database (/MVSDB) .....	266
MySQL Connection Inputs File (/MYSQL) .....	266
zonemonthhour Database Table Output (/DZHM) .....	267
zonemonthhour Tab-Delimited Output (/TZHM) .....	267
County Database Table Output (/DCTY) .....	267
County Tab-Delimited Output (/TCTY) .....	267
<b>APPENDIX A: MOVES CODES AND UTILITY ABBREVIATIONS .....</b>	<b>269</b>

## LIST OF FIGURES

Figure 1. Sample Batch Mode Input File.....	4
Figure 2. Sample VirtualLinkVMT JCF.....	14
Figure 3. Sample Section of RIFCREC Data Input File for Years 2006 and Later.....	16
Figure 4. Sample Section of RIFCREC Data Input File for Toll Roads for Years 2006 and Later.....	17
Figure 5. Sample County Name and FIPS Code Input File.....	17
Figure 6. Sample Year and Day Type Input File.....	18
Figure 7. Sample VMT Control Totals Input File.....	18
Figure 8. Sample Hourly Factors Input File.....	19
Figure 9. Suggested Speed Model Input Data File.....	20
Figure 10. Sample 24-hour TDM TRANSVMT jcf.....	36
Figure 11. Sample Four Time-Period TDM TRANSVMT jcf.....	37
Figure 12. Sample Header Input File.....	38
Figure 13. Sample Hourly Factors Input File.....	39
Figure 14. Sample Centroid Connector Road Type Code Input File.....	39
Figure 15. Sample Directional Splits Input File.....	40
Figure 16. Sample VMT Adjustments Input File.....	41
Figure 17. Sample Intrazonal Radii Input File.....	42
Figure 18. Sample Time Period Designation Input File.....	43
Figure 19. Sample Coordinate Input File.....	43
Figure 20. Sample Link Data Fields Input File.....	45
Figure 21. Sample Link Data Dictionary File.....	46
Figure 22. Sample Link Data Input File.....	46
Figure 23. Sample Intrazonal Trips Input File.....	47
Figure 24. Sample TTI Speed Model Data Input File.....	48
Figure 25. Sample Delay Parameters Input File.....	48
Figure 26. Sample Houston Speed Model Data Input File.....	49
Figure 27. Sample Node Coordinate Output File.....	52
Figure 28. Sample IntraTripsOutput JCF.....	54
Figure 29. Sample Trip Table Input File.....	55
Figure 30. Sample Intrazonal Trips Output File.....	56
Figure 31. Sample RADIICALC jcf.....	58
Figure 32. Sample Centroid Connector Road Type Code Input File.....	59
Figure 33. Sample Link Data Fields Input File.....	61
Figure 34. Sample Link Data Dictionary File.....	61
Figure 35. Sample Link Data Input File.....	62
Figure 36. Sample 24-hour TDM VMTtot jcf.....	66
Figure 37. Sample Four Time-Period TDM VMTtot jcf.....	66
Figure 38. Sample Header Input File.....	67
Figure 39. Sample Centroid Connector Road Type Code Input File.....	67
Figure 40. Sample Intrazonal Radii Input File.....	68

Figure 41. Sample Link Data Fields Input File.....	69
Figure 42. Sample Link Data Dictionary File.....	69
Figure 43. Sample Link Data Input File. ....	70
Figure 44. Sample Intrazonal Trips Input File.....	71
Figure 45. Sample VehPopulationBuild jcf. ....	80
Figure 46. Sample County Name Input File. ....	81
Figure 47. Sample Age Registrations Data Input File. ....	82
Figure 48. Sample Diesel Heavy-Duty Registrations Data Input File.....	82
Figure 49. Sample of an Diesel Heavy-Duty Registrations Data Input File.....	83
Figure 50. Sample TxDOT District Name Input File. ....	83
Figure 51. Sample VMT Mix Input File. ....	85
Figure 52. Sample Population Factor Input File. ....	86
Figure 53. Sample Year ID Input File. ....	86
Figure 54. Sample MySQL Connections Input File. ....	87
Figure 55. Sample OffNetActCalc jcf. ....	96
Figure 56. Sample County Input File.....	98
Figure 57. Sample VMT Road Type Designations Input File. ....	99
Figure 58. Sample 24-Hour VMT Mix Input File. ....	100
Figure 59. Sample Time Period Designation Input File. ....	101
Figure 60. Sample Tab-Delimited Population by SUT/Fuel Type Input File.....	102
Figure 61. Sample VMT Listing Input File. ....	103
Figure 62. Sample Hotelling Factor Input File. ....	103
Figure 63. Sampe Hotelling Hours Input File.....	104
Figure 64. Sample Age Distribution Input File. ....	106
Figure 65. Sample Hotelling Activity Distribution Input File.....	107
Figure 66. Sample Relative Mileage Accumulation Rates Input File. ....	107
Figure 67. Sample Starts per Vehilce Input File.....	108
Figure 68. Sample OffNetActSummary jcf. ....	114
Figure 69. Sample Vehicle Population Input File.....	115
Figure 70. Sample SHP Input File. ....	116
Figure 71. Sample Starts Input File. ....	117
Figure 72. Sample Hotelling/SHI/APU Hours Input File. ....	117
Figure 73. Sample Header Input File. ....	118
Figure 74. Sample VMTmixBuild jcf.....	122
Figure 75. Sample VMT Mix Input File. ....	124
Figure 76. Sample TxDOT District Name Input File. ....	125
Figure 77. Sample Comments Input File. ....	125
Figure 78. Sample VMT Mix Output File. ....	126
Figure 79. Sample MOVESfleetInputBuild jcf. ....	138
Figure 80. Sample sourcetypeagedistribution Data Sources Input File.....	140
Figure 81. Sample SUT Data Source File.....	141
Figure 82. Sample fuelengfraction Data Sources Input File.....	141
Figure 83. Sample Age Registrations Data Input File. ....	142
Figure 84. Sample Diesel Heavy-Duty Registrations Data Input File.....	143
Figure 85. Sample of an Gas Heavy-Duty Registrations Data Input File.....	143

Figure 86. Sample Default Fuel Engine Fractions Input File. ....	144
Figure 87. Sample of a Default sourcetypeagedistribution Input File. ....	145
Figure 88. Sample Year ID Input File. ....	145
Figure 89. Sample VMT Mix Input File. ....	146
Figure 90. Sample MySQL Connections Input File. ....	147
Figure 91. Sample MOVESoutSummary jcf. ....	152
Figure 92. Sample MySQL Connection Inputs File. ....	154
Figure 93. Sample Database Listing Input File. ....	154
Figure 94. Sample RatesCalc jcf. ....	163
Figure 95. Sample MySQL Connection Inputs File. ....	164
Figure 96. Sample Pollutants Input File. ....	165
Figure 97. Sample SHP Calculation Database Tables Input File ....	167
Figure 98. Sample RatesAdj jcf. ....	173
Figure 99. Sample MySQL Connections Input File. ....	175
Figure 100. Sample Emissions Rate Database/Table Input File. ....	176
Figure 101. Sample Pollutants Input File. ....	177
Figure 102. Sample Emissions Rate Adjustments Input File (SINGLE FACTOR Option).....	178
Figure 103. Sample Emissions Rate Adjustments Input File (PARTS Option). ....	178
Figure 104. Sample Database Emissions Rate Adjustments Input File. ....	179
Figure 105. Sample EmsCalc jcf. ....	193
Figure 106. Sample Header Input File. ....	195
Figure 107. Sample County Name Input File. ....	195
Figure 108. Sample VMT Listing Input File. ....	195
Figure 109. Sample VMT Road Type Codes Input File. ....	196
Figure 110. Sample VMT Road Type Designations Input File. ....	197
Figure 111. Sample Units Input File. ....	198
Figure 112. Sample 24-Hour VMT Mix Input File. ....	199
Figure 113. Sample Time Period Designation Input File. ....	200
Figure 114. Sample SHP by SUT/Fuel Type Input File. ....	202
Figure 115. Sample Sarts by SUT/Fuel Type Input File. ....	203
Figure 116. Sample SHI and APU Hours by SUT/Fuel Type Input File. ....	204
Figure 117. Sample MySQL Connection Inputs File. ....	205
Figure 118. Sample SCC Input File. ....	206
Figure 119. Sample SCC Pollutants Input File. ....	206
Figure 120. Sample Hourly Link Output Files Listing Input File. ....	207
Figure 121. Sample LinkemsSum jcf. ....	213
Figure 122. Sample Header Input File. ....	215
Figure 123. Sample VMT Road Type Codes Input File. ....	215
Figure 124. Sample Hourly Link Emissions Listing Input File. ....	216
Figure 125. Sample Output Units Input File. ....	217
Figure 126. Sample TabFileTotals jcf. ....	223
Figure 127. Sample Header Input File. ....	224
Figure 128. Sample SUT Tab-Delimited Output File. ....	225
Figure 129. Sample Totals Tab-Delimited Output File. ....	226
Figure 130. Sample Hourly Totals Tab-Delimited Output File. ....	227



Figure 131. Sample Hourly SUT Tab-Delimited Output File. ....	227
Figure 132. Sample Hourly Road Type Tab-Delimited Output File. ....	228
Figure 133. Sample Hourly SUT/Road Type Tab-Delimited Output File.....	228
Figure 134. Sample 24-hour SUT/Road Type Tab-Delimited Output File. ....	229
Figure 135. Sample MOVESactivityInputBuild jcf.....	245
Figure 136. Sample County Input File.....	246
Figure 137. Sample VMT and Speeds Listing Input File. ....	247
Figure 138. Sample VMT Road Type Designations Input File. ....	248
Figure 139. Sample 24-Hour VMT Mix Input File. ....	249
Figure 140. Sample Time Period Designation Input File. ....	250
Figure 141. Sample VMT Road Type/Ramp Designation Input File. ....	252
Figure 142. Sample MySQL Connection Inputs File. ....	252
Figure 143. Sample sourcetypeyear Input File. ....	253
Figure 144. Sample sourcetypeagedistribution Input File. ....	253
Figure 145. Sample sourcetypeage Input File.....	254
Figure 146. Sample Starts Input File. ....	254
Figure 147. Sample Hotelling Input File. ....	254
Figure 148. Sample MOVESmetInputBuild jcf.....	263
Figure 149. Sample County Input File.....	265
Figure 150. Sample Month ID Input File.....	265
Figure 151. Sample Meteorological Data Input File. ....	266
Figure 152. Sample MySQL Connection Inputs File. ....	266

## LIST OF TABLES

Table 1. MOVES2014aUtl Utilities.....	1
Table 2. RIFCREC Data Area Type Descriptions and Area Type Numbers.....	8
Table 3. RIFCREC Data Functional Class Descriptions and Functional Class Numbers.....	8
Table 4. Area Type/Functional Class Combinations and Corresponding Roadway Type Numbers.....	10
Table 5. VirtualLinkVMT File Type Identifiers.....	13
Table 6. Header Requirements for the TxDOT RIFCREC Data Input File for 2006 and Later.....	16
Table 7. Header Requirements for the TxDOT RIFCREC Data Input File for Toll Roads for 2006 and Later.....	17
Table 8. Volume-Delay Equation Parameters.....	29
Table 9. Sample Set of Speed Reduction Factors.....	32
Table 10. TRANSVMT File Type Identifiers.....	35
Table 11. IntraTripsOutput File Type Identifiers.....	54
Table 12. RADIICALC File Type Identifiers.....	58
Table 13. VMTtot File Type Identifiers.....	65
Table 14. Registration Categories.....	75
Table 15. SUT/Registration Category Correlation.....	76
Table 16. VehPopulationBuild File Type Identifiers.....	79
Table 17. OffNetActCalc File Type Identifiers.....	95
Table 18. VMT Road Type Designations Input File Header Line.....	98
Table 19. 24-Hour VMT Mix Input File Header Line.....	99
Table 20. Time Period Designation Input File Header Line.....	100
Table 21. Tab-Delimited Population by SUT/Fuel Type Input File Header Line.....	102
Table 22. Base Hotelling Hours Input File Header Line.....	103
Table 23. Age Distribution Input File Header Line.....	105
Table 24. Hotelling Activity Distribution Input File Header Line.....	106
Table 25. Relative Mileage Accumulation Rates Input File Header Line.....	107
Table 26. Start per Vehicle Input File Header Line.....	108
Table 27. OffNetActSummary File Type Identifiers.....	113
Table 28. Vehicle Population Input File Header Line.....	115
Table 29. SHP Input File Header Line.....	116
Table 30. Starts Input File Header Line.....	116
Table 31. Hotelling/SHI/APU Hours Input File Header Line.....	117
Table 32. VMT Mix Road Type Substitutions.....	121
Table 33. VMTmixBuild File Type Identifiers.....	122
Table 34. Registration Categories.....	133
Table 35. SUT/Registration Category Correlation For SUT/Age Distribution Array.....	133
Table 36. MOVESfleetInputBuild File Type Identifiers.....	137
Table 37. VMT Mix Input File Header Line.....	146
Table 38. MOVESoutSummary File Type Identifiers.....	152

Table 39. MOVES Emissions Process and Corresponding RatesCalc Input/Output Database Tables. ....	158
Table 40. RatesCalc File Type Identifiers. ....	162
Table 41. SHP Calculation Database Tables Input File Header Line. ....	167
Table 42. RatesAdj File Type Identifiers. ....	173
Table 43. Pollutants Input File Header Line. ....	176
Table 44. MOVES2014a Emissions Process and Corresponding Activity. ....	190
Table 45. EmsCalc File Type Identifiers. ....	192
Table 46. VMT Road Type Designations Input File Header Line. ....	196
Table 47. 24-Hour VMT Mix Input File Header Line. ....	198
Table 48. Time Period Designation Mix Input File Header Line. ....	199
Table 49. SHP by SUT/Fuel Type Input File Header Line. ....	202
Table 50. Starts by SUT/Fuel Type Input File Header Line. ....	203
Table 51. SHI and APU Hours by SUT/Fuel Type Input File Header Line. ....	204
Table 52. SCC Input File Header Line. ....	205
Table 53. SCC Pollutants Input File Header Line. ....	206
Table 54. Hourly Link Output Files Format. ....	208
Table 55. LinkEmsSum File Type Identifiers. ....	213
Table 56. MOVESabFileTotals File Type Identifiers. ....	222
Table 57. SUT/HPMS Vehicle Type Correlations. ....	240
Table 58. MOVESactivityInputBuild File Type Identifiers. ....	243
Table 59. VMT Road Type Designations Input File Header Line. ....	247
Table 60. 24-Hour VMT Mix Input File Header Line. ....	248
Table 61. Time Period Designation Input File Header Line. ....	249
Table 62. VMT Road Type/Ramp Designation Input File Header Line. ....	251
Table 63. MOVESmetInputBuild File Type Identifiers. ....	263
Table 64. MOVES Hours and Abbreviations. ....	271
Table 65. MOVES Road Types and Abbreviations. ....	272
Table 66. MOVES Pollutants and Abbreviations. ....	273
Table 67. MOVES Emissions Processes and Abbreviations. ....	277
Table 68. MOVES Average Speed Bins and Abbreviations. ....	277
Table 69. MOVES SUTs and Abbreviations. ....	278
Table 70. MOVES Fuel Types and Abbreviations. ....	278
Table 71. MOVES Activity Types and Abbreviations. ....	279



## OVERVIEW

MOVES2014aUtl is a collection of emissions inventory estimation utilities developed by the Texas A&M Transportation Institute (TTI) specifically designed for use with MOVES2014a (referred to as “MOVES” in this document, unless otherwise stated) and the associated latest available default database (movesdb20151028). Any changes to MOVES (i.e., updated version of MOVES or default databases) may affect the operation of MOVES2014aUtl or cause erroneous results. All of the utilities, including MOVES2014aUtl, were written in the Visual Basic programming language, which makes MOVES2014aUtl compatible with most computers. Some of the utilities also use the version of MySQL provided with MOVES to access and process database tables.

MOVES2014aUtl is job control file (jcf)-driven code that allows the user to access the individual utilities contained in MOVES2014aUtl based on the utility specified in the jcf. The utilities contained in MOVES2014aUtl can be divided into four main categories: activity (VMT/speeds and off-network activity), emissions rates, emissions, and miscellaneous. Table 1 shows the list of utilities.

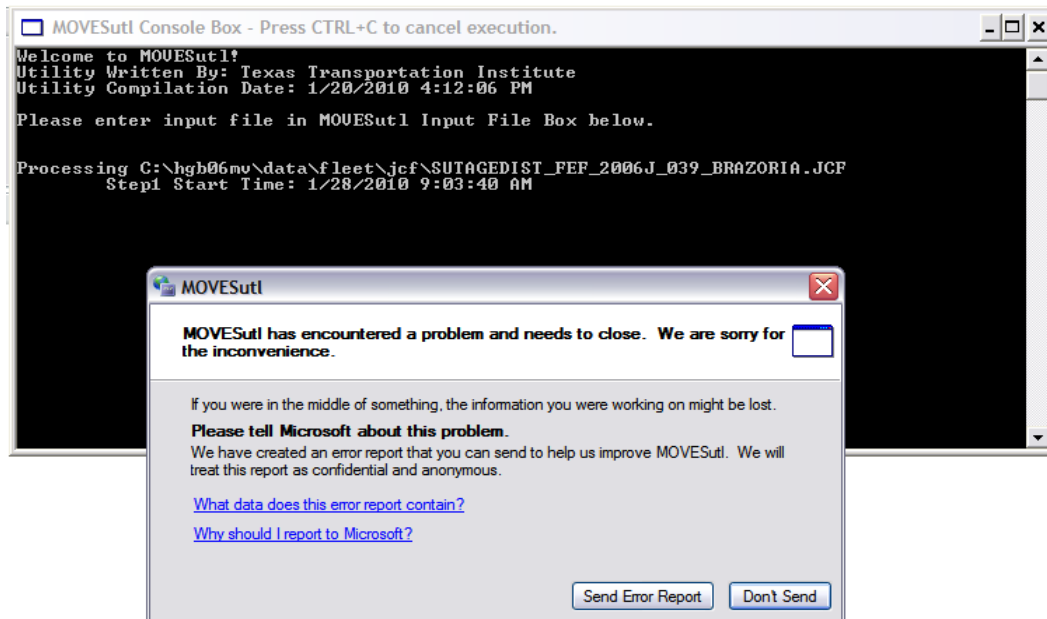
**Table 1. MOVES2014aUtl Utilities.**

Category	Utility
Activity Utilities	VirtualLinkVMT
	TRANSVMT
	IntraTripsOutput
	RADIICALC
	VMTtot
	VehPopulationBuild
	OffNetActCalc
	OffNetActSummary
	VMTmixBuild
Emissions Rates Utilities	MOVESfleetInputBuild
	MOVESoutSummary
	RatesCalc
	RatesAdj
Emissions Utilities	EmsCalc
	Linkemssum
	TabFileTotals
Miscellaneous Utilities	MOVESactivityInputBuild
	MOVESmetInputBuild

The main MOVES parameters used by these utilities have ID codes and descriptions within MOVES2014. Some of these descriptions are rather large and not suitable for direct output, so a set of abbreviations are used for some of these parameters. Appendix A contains the MOVES hours, road types, pollutants, pollutant process, average speed bins, source use types (SUTs), fuel types, activity types, and associated utility abbreviations.

## INSTALLATION

Since the utilities were all written in Visual Basic, no installation is required to execute MOVES2014aUtl. However, some of the utilities use MySQL to access databases. For this feature to operate correctly, the user must have a MySQL connector (contained in the zip file mysql-connector-net-5.2.7.zip) installed on their computer. If this connector is missing, the utility will encounter an error and an error message similar to the one shown below will appear.



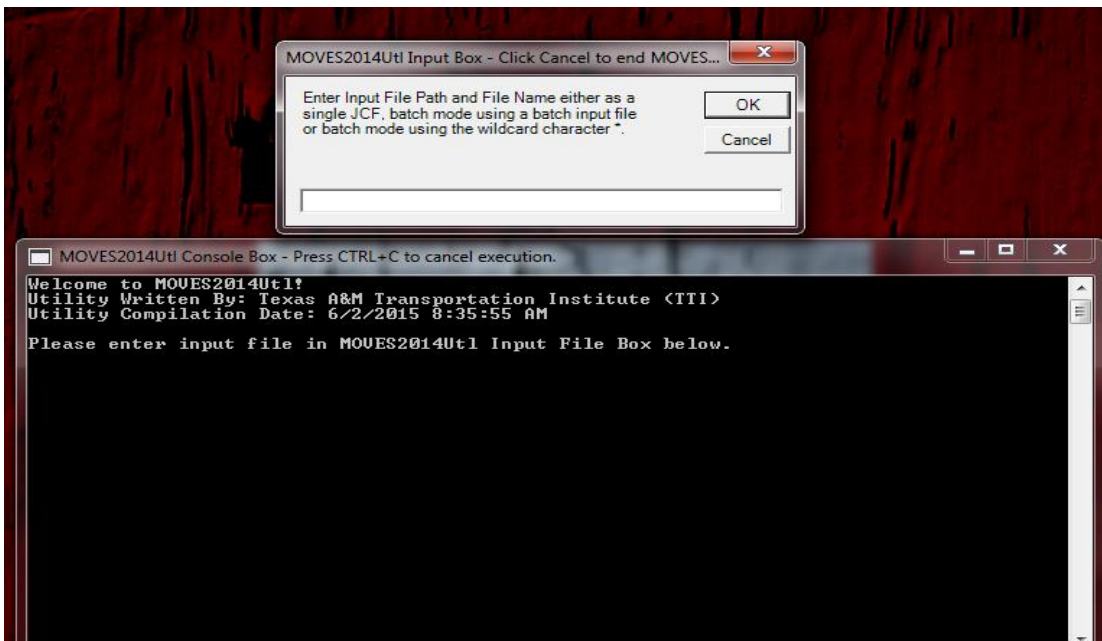
In this case, the user must install the MySQL connector so that the utilities can use MySQL to access the necessary databases. Following are instructions for installing this connector.

1. Copy the mysql-connector-net-5.2.7.zip file to the MySQL directory (i.e., C:\MySQL).
2. Open the zip file and extract the mysql.data.msi file to the MySQL directory (i.e., C:\MySQL).
3. Click on the mysql.data.msi file. This begins the installation.
4. After the installation begins, click Next.
5. Then click on install Custom.

6. Then under the core components, set the folder name to the MySQL directory, followed by the connector folder (i.e., C:\MySQL\MySQL Connector Net 5.2.7).
7. Then click Next.
8. Then click Install.
9. Then click Finish. This completes the installation of the connector and MOVES2014aUtl should work with MySQL.

## MOVES2014AUTL OPERATION

After initiating an execution of MOVES2014aUtl, two main boxes will appear on the computer screen – the MOVES2014aUtl Console Box and the MOVES2014aUtl Input Box. The console box is very similar to a DOS window. The MOVES2014aUtl Input Box is where the user inputs the batch input file or jcf (discussed in the following section). The user can cancel the execution of MOVES2014aUtl at any point by pressing CTRL+C. Below is an example of the console and input boxes.



The utility can be run using batch mode (i.e., multiple jcf files in the input file) or one jcf input file. For the batch mode, the utility reads the first input line and opens that jcf file. The utility then follows the same process as it would if the input file was only one jcf input file. The following describes this process.

When processing the jcf input file, MOVES2014aUtl reads each line of the jcf until it reaches the first step (identified by the /STEP identifier, discussed later). MOVES2014aUtl then determines which utility to run from that same line and begins that utility. After the specified utility terminates, MOVES2014aUtl then reads the jcf line-by-line until it reaches the second step. The utility then determines which utility to run from that same line and begins that utility.

The utility repeats this process until completing all of the steps in the jcf. If MOVES2014aUtl does not recognize the utility in the step identifier, an error message appears on the screen and the utility terminates the jcf.

## MOVES2014AUTL INPUTS

The inputs required for the MOVES2014aUtl utility can be in one of two formats – a single jcf or batch mode.

The batch mode option allows the user to specify multiple input files (jcf files) for processing by MOVES2014aUtl in one single run. If not using the batch mode, the user must input each of the jcf files individually. The batch mode input file consists of the path and filename of each jcf to process. Figure 1 shows a sample batch mode input file.

```
JCF C:\MOVES2014aUtl\EmsCalc\Input\jcf\EmsCalc_TDM.jcf
JCF C:\MOVES2014aUtl\EmsCalc\Input\jcf\EmsCalc_VLink.jcf
```

**Figure 1. Sample Batch Mode Input File.**

The jcf input file lists all of the inputs and outputs for the specified utility. Each line consists of a file type identifier and a text string representing either a comment, the utility to execute, or the path and filename of the input or output file. The first set of text in each line of the jcf input file is for the file type identifiers (discussed for each individual utility later in the document).

## MOVES2014AUTL OUTPUTS

Since MOVES2014aUtl is only an interface that allows access to the utilities, there are no outputs specific to MOVES2014aUtl. However, the console box will contain some pertinent information regarding the execution of the individual utilities, such as (but not limited to) the beginning time of the step being executed, the end time of the step being executed, and some warning/error messages regarding the step being executed. Note that the console box may not include all warnings and/or errors encountered during the execution of the step; thus, checking all listing output files from the individual utilities for warning and error messages is critical to ensure the proper execution of the step.



## **ACTIVITY UTILITIES**



## VIRTUALLINKVMT: HPMS-BASED LINK VMT AND SPEEDS ESTIMATION

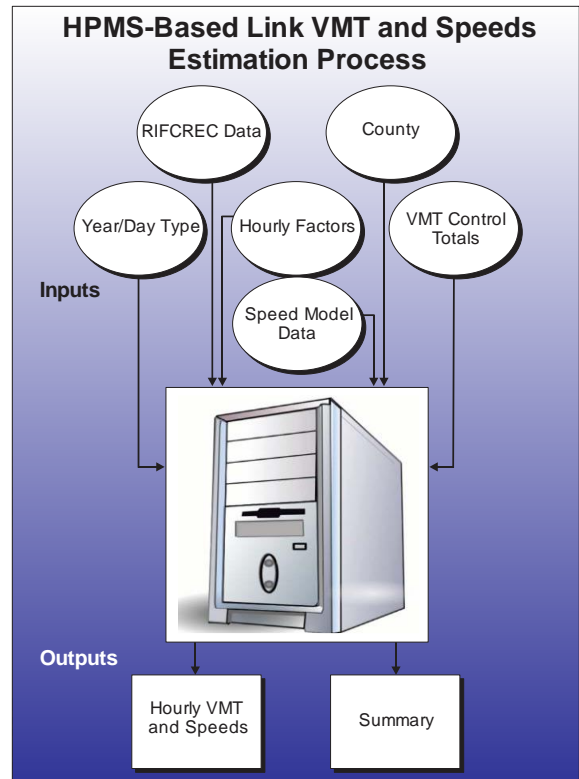
The VirtualLinkVMT utility estimates vehicle miles of travel (VMT) and speeds based on the Highway Performance Monitoring System (HPMS) average annual daily traffic (AADT) VMT, centerline miles, lane miles by functional class and area type combination (or virtual link), and user-defined VMT control totals, hourly factors, and speed model parameters. Each virtual link consists of the HPMS area type and functional class combination and is divided into primary and secondary directions, resulting in up to 42 possible links. Although the utility is mainly designed for those areas that do not have travel demand models (TDMs), the utility can be used for any of the 254 counties in Texas.

### OPERATION

The VirtualLinkVMT utility reads the jcf until it reaches the step to run. Each line of the desired step is written to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The jcf input file is then re-opened. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or VirtualLinkVMT cannot open any of these input files due to an incorrect path and/or filename or any other access error, then VirtualLinkVMT results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file processed by the utility is the county name and Federal Information Processing Standards (FIPS) code input file. This input file consists of only one line, with the FIPS code followed by the county name. This FIPS code is very important, as this code is used as the county identifier in all of the input and output files.



The next input file processed by the utility is the analysis year and day type input file. This input file is processed in a similar manner with the utility saving both the analysis year and day type. The analysis year and day type must appear the same as they appear in the other input files.

The utility then processes the Texas Department of Transportation’s (TxDOT) Roadway Inventory Functional Classification Record (RIFCREC) data input file. Otherwise, the first step is to determine the format of the RIFCREC data input file. If this input file is in the original format (i.e., year 2005 and prior), the utility then locates the desired county (based on the county FIPS code from the county name and FIPS code input file) and saves the centerline, lane, and daily VMT (AADT VMT) for each of the seven roadway type combinations and four area type combinations in the VMT array for the calculations. If this input file is in the newer format (i.e., year 2006 and later), the utility uses the first line (header line) of the input file to determine the location of the county number, functional class, area type, centerline, lane, and daily VMT data. The utility locates the desired county and saves the appropriate data to the VMT array. The utility then processes the RIFCREC data for toll roads in a similar manner, if input in the jcf, and saves that data in the VMT array as well. Regardless of the format, the total AADT VMT for all area type/functional class combinations is also calculated. Table 2 shows the RIFCREC data area types and their corresponding area type numbers followed by the RIFCREC data functional classes (Table 3) and their corresponding functional class numbers.

**Table 2. RIFCREC Data Area Type Descriptions and Area Type Numbers.**

Area Type Description	Area Type Number
Rural Miles (Pop. 1 - 4,999)	1
Small Urban Miles (Pop. 5,000 - 49,999)	2
Urbanized Miles (Pop. 50,000 - 199,999)	3
Urbanized Miles (Pop. 200,000 +) <sup>1</sup>	4

<sup>1</sup> For calculation and hourly output purposes, this area type is considered area type 3.

**Table 3. RIFCREC Data Functional Class Descriptions and Functional Class Numbers.**

Functional Class Description	Functional Class Number
Interstate	1
Urban Freeway	2
Principal Arterial	3
Minor Arterial	4
Major Collector	5
Minor Collector	6
Local	7

The utility then processes the VMT control totals input file. Using the analysis year, day type, and the county FIPS code, the utility selects the appropriate control total VMT and the TxDOT district where the county is located. The VMT control total and the total AADT VMT from the RIFCREC data are then used to calculate the VMT factor.

The next file processed by the utility is the VMT fractions input file. Using the TxDOT district obtained from the VMT control totals input file, the utility locates the appropriate TxDOT district in the VMT fractions input file and saves the 24 hourly VMT fractions in the hourly factor array. If the sum of these 24 fractions does not equal 1, then the largest fraction is adjusted so that the sum equals 1 and the utility writes a warning message to the listing file.

The final input file processed by the utility is the speed model data input file. Each line of this input file is read, saving the speed model data for each area type/functional class combination in the speed model data array.

Then utility then begins the process of estimating the hourly VMT and speeds for each virtual link. The path and filename of each hourly VMT and speeds output file is based on the listing output file, with the extension of the listing output file (i.e., .lst) replaced with the text \*.*TX*X, where *XX* represents each hour of the day (i.e., 01, 02, ..., 24). The following paragraphs describe the VMT and speed estimation process performed for each area type/functional class combination that has an AADT VMT greater than zero and for each hour of the day.

First, a roadway type number is assigned based on the area type and functional class numbers. This roadway type number is used in the hourly VMT and speeds output files.

Table 4 shows the correlation between the area type/functional class combination and the roadway type numbers.

**Table 4. Area Type/Functional Class Combinations and Corresponding Roadway Type Numbers.**

Area Type	Functional Class	Roadway Type Number	Roadway Type Description
1	1	0	Rural Interstate
1	3	2	Rural Principal Arterial
1	4	3	Rural Minor Arterial
1	5	4	Rural Major Collector
1	6	5	Rural Minor Collector
1	7	6	Rural Local
2	1	7	Small Urban Interstate
2	2	8	Small Urban Freeway
2	3	9	Small Urban Principal Arterial
2	4	10	Small Urban Minor Arterial
2	5	11	Small Urban Major Collector
2	6	12	Small Urban Minor Collector
2	7	13	Small Urban Local
3, 4	1	14	Urban Interstate
3, 4	2	15	Urban Freeway
3, 4	3	16	Urban Principal Arterial
3, 4	4	17	Urban Minor Arterial
3, 4	5	18	Urban Major Collector
3, 4	6	19	Urban Minor Collector
3, 4	7	20	Urban Local

Based on the area type/functional class combination (or virtual link), the utility selects the appropriate hourly lane capacity, free-flow speed, A parameter, B parameter, and M parameter from the speed model data array. For use in this utility, RIFCREC data for area types 3 and 4 are combined. The number of lanes (using the RIFCREC data) and the hourly capacity for the virtual link are then calculated. For the capacity calculation, a split of 50-50 is assumed. These values are calculated using the following equations:

$$\begin{aligned} \text{Lanes(A,B)} &= \text{LaneMiles(A, B)}/\text{CenterlineMiles(A, B)} \\ \text{Capacity(A,B)} &= \text{HLCapacity(A, B)} \times \text{Lanes(A, B)} \times 0.5 \end{aligned}$$

Where:

- A = area type number;
- B = functional class number;
- Lanes(A,B) = number of lanes for the virtual link(A, B);
- LaneMiles(A,B) = the lane miles for the area type combination (A, B) from the RIFCREC data;
- CenterlineMiles (A,B) = the centerline miles for the area type combination (A, B) from the RIFCREC data;
- Capacity(A,B) = the calculated hourly capacity for the virtual link (A, B); and
- HLCapacity(A,B) = the hourly lane capacity for the virtual link (A, B) from the speed model data array.

The hourly factored VMT and volume are then calculated for the primary direction. The directional splits for the hourly factored VMT and resulting volume are assumed as 60-40. These values are calculated using the following equations:

$$\begin{aligned}\text{HourlyVMT1(A,B)} &= \text{VMT(A,B)} \times \text{VMTfactor} \times \text{HourlyFactor(hr)} \times 0.6 \\ \text{HourlyVolume1(A,B)} &= \text{HourlyVMT1(A,B)}/\text{CenterlineMiles (A,B)}\end{aligned}$$

Where:

- HourlyVMT1(A,B) = the calculated hourly VMT for the virtual link (A, B) for the primary direction for hour (hr);
- VMT(A,B) = the VMT for the virtual link (A, B) from the RIFCREC data array;
- VMTfactor = the calculated VMT factor from the VMT control total and the daily AADT VMT from the RIFCREC data;
- HourlyFactor(hr) = the hourly factor from the hourly factor array for hour (hr); and
- HourlyVolume1(A,B) = the calculated hourly volume for the virtual link (A, B) for the primary direction for hour (hr).

The volume-to-capacity (v/c) ratio and directional delay (in minutes per mile) due to congestion for the primary direction are then calculated. These values are calculated using the following equations:

$$\begin{aligned}\text{VC1(A,B)} &= \text{HourlyVolume1(A,B)} / \text{Capacity(A,B)} \\ \text{Delay1(A,B)} &= \text{Min}[Ae^{B\{\text{VC1(A,B)}\}}, M]\end{aligned}$$

Where:

- VC1(A,B) = the calculated v/c ratio for the virtual link (A, B) for the primary direction;
- Delay1(A,B) = congestion delay (in minutes/mile) for the primary direction;
- A & B = volume-delay equation coefficients; and
- M = maximum minutes of delay per mile.

The congested speed is then calculated for the primary direction using the free-flow speed and the delay using the following equation:

$$\text{CSPD1(A,B)} = 60 / (\{60/\text{FSPD(A,B)}\} + \text{Delay1(A,B)})$$

Where:

CSPD1(A,B) = the congested speed for the primary direction; and  
 FSPD(A,B) = the free-flow speed for the virtual link (A, B) from the speed model data array.

The utility then writes the data for the primary direction to the appropriate hourly output file. This output consists of:

- ANODE – the RIFCREC data area type number, except for area type 4 where the area type is changed to 3;
- BNODE – the RIFCREC data functional class number;
- County number – always 1 since only one county is processed at a time;
- Roadway type – the assigned roadway type based on the area type and functional class numbers (see Table 4);
- Centerline miles – from the RIFCREC data array;
- Calculated congested speed;
- Calculated hourly VMT;
- County name;
- Area type; and
- V/C ratio.

The utility then performs the same calculations for the secondary direction, except the directional split is 0.4 instead of 0.6. The calculations for the secondary direction are:

$$\begin{aligned} \text{HourlyVMT2(A,B)} &= \text{VMT(A,B)} \times \text{VMTfactor} \times \text{HourlyFactor(hr)} \times 0.4 \\ \text{HourlyVolume2(A,B)} &= \text{HourlyVMT2(A, B)}/\text{CenterlineMiles (A, B)} \\ \text{VC2(A,B)} &= \text{HourlyVolume2(A, B)}/\text{Capacity(A, B)} \\ \text{Delay2(A,B)} &= \text{Min}[Ae^{B\{\text{VC2(A,B)}\}}, M] \\ \text{CSPD2(A,B)} &= 60 / (\{60/\text{FSPD(A,B)}\} + \text{Delay2(A, B)}) \end{aligned}$$

Where:

HourlyVMT2(A,B) = the calculated hourly VMT for the virtual link (A, B) for the primary direction for hour (hr);  
 HourlyVolume2(A,B) = the calculated hourly volume for the virtual link (A, B) for the primary direction for hour (hr);



- VC2(A,B) = the calculated v/c ratio for the virtual link (A, B) for the primary direction;
- Delay2(A,B) = congestion delay (in minutes/mile) for the primary direction; and
- CSPD2(A,B) =  $60/(\{60/FSPD(A, B)\}+Delay2(A, B))$ .

The utility then writes the hourly output for the secondary direction. This process is repeated for each area type/functional class combination that has VMT in the RIFCREC data and for each hour of the day.

## JCF INPUT FILE

The jcf input file directs the VirtualLinkVMT utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (VirtualLinkVMT in this case), or the path and filename of the input or output file. Table 5 shows the file type identifiers. Any other characters at the beginning of a line in the jcf input file are not recognized by the utility and do not affect the execution of the utility.

**Table 5. VirtualLinkVMT File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Step Identifier and Utility Name
/LST	Listing Output File	Path and Filename
/RIFC	TxDOT RIFCREC Data Input File	Path and Filename
/TRIFC	TxDOT RIFCREC Data Input File for Toll Roads <sup>2</sup>	Path and Filename
/CNTY	County Name and FIPS Code Input File	Path and Filename
/YRDY	Year and Day Type Input File	Path and Filename
/CTRL	VMT Control Totals Input File	Path and Filename
/HFCT	Hourly Factors Input File	Path and Filename
/SPDM	Speed Model Data Input File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only used for reference purposes and not accessed by the utility.

<sup>2</sup> Only if the new format (2006 or later) of the RIFCREC data are used with the /RIFC identifier.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (VirtualLinkVMT), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 2 shows a sample jcf.

/JOB	VirtualLinkVMT
/STEP	Step VirtualLinkVMT
/LST	C:\VirtualLinkVMT\Output\Wk1\VirtualLinkVMT_Wk.lst
/RIFC	C:\RIFCREC\2004\HPMS_04.DAT
/CNTY	C:\VirtualLinkVMT\Input\VirtualLinkVMTCountyFIPSCode.txt
/YRDY	C:\VirtualLinkVMT\Input\Year2005_Wk.txt
/CTRL	C:\VirtualLinkVMT\VirtualLinkVMT_DayTypeADJ.Lst
/HFCT	C:\VirtualLinkVMT\VirtualLinkVMT_Hrs.Lst
/SPDM	C:\VirtualLinkVMT\Input\SpeedModelData.txt

**Figure 2. Sample VirtualLinkVMT JCF.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. This identifier is not used by the utility but is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. This text string is not used by the VirtualLinkVMT utility, but only by MOVES2014aUtl to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part being separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. The listing file summarizes all the input and output for the specific jcf input file. The path and filename of the hourly VMT and speeds output files are generated using the path and filename of listing output file, minus the file extension. For example, if the listing file is designated as BPA2005weekday.lst, then the hourly files will be BPA2005weekday.T01, BPA2005weekday.T02, ..., BPA2005weekdayT.24, with .T01 being 12:00-to-1:00 a.m., .T02 being 1:00-to-2:00 a.m., ... , and .T24 being 11:00-to-12:00 p.m. The information in the listing file includes:

- Begin time of the step;
- Utility name (in this case VirtualLinkVMT);
- Path and filename of the jcf input file;

- JCF through the current step;
- Path, filename, and contents of the county name and FIPS code input file;
- Path, filename, and contents of the year and day type input file;
- Path and filename of the TxDOT RIFCREC input file;
- Total AADT VMT for all area type/functional class combinations;
- Path and filename of the VMT control totals input file;
- VMT control total;
- TxDOT district where the county is located;
- Calculated VMT factor;
- Path and filename of the hourly factors input file;
- Sum of the hourly factors;
- Hourly factors used (adjusted to sum to 1 if necessary);
- Path and filename of the speed model data input file;
- Names of each of the 24 hourly output files;
- Description of the data in the hourly output files;
- Hourly VMT and speeds (VMT/vehicle hours traveled[VHT]) summary; and
- End time of the step after it is completed.

### **TxDOT RIFCREC Data Input File (/RIFC)**

The /RIFC identifier (required) and its corresponding text string specify the path and filename of the TxDOT RIFCREC data input file, with one adjustment. The TxDOT RIFCREC data changed formats beginning with the 2006 data. This utility will work with either format.

For the RIFCREC data prior to 2006, the string *COUNTY:* is replaced by the county's three-digit FIPS code.

For the RIFCREC data for 2006 and later, the original TxDOT RIFCREC data file (an Microsoft® Excel spreadsheet) must be converted to a semicolon-delimited text file. This text file should begin with the header line, which indicates the position of each set of data. No other text or data should appear before the header line. This RIFCREC data does not include toll roads. If this format is used, the toll road data must be input in the jcf using the /TRIFC.

Table 6 shows the required data as it should appear in the header line and Figure 3 shows a sample section of the RIFCREC data in this format.

**Table 6. Header Requirements for the TxDOT RIFCREC Data Input File for 2006 and Later.**

Header Requirement <sup>1</sup>	Description
CO	County Number
FUNSYS	Functional Class
POP	Area Type
CENTERLINE	Centerline Miles
LANE	Lane Miles
DVM	Daily Vehicle Miles

<sup>1</sup> Capitalization of the header requirements is not required.

CO; CON;	FUNSYS;	POP;	CENTERLINE;	LANE;	DVM;	TVM
1; Anderson;	Principal Arterial;	Rural (<5,000);	28.266;	60.928;	202322.700;	59182.720
1; Anderson;	Principal Arterial;	Small Urban (5,000-49,999);	29.547;	102.388;	334307.200;	41711.171
1; Anderson;	Minor Arterial;	Rural (<5,000);	107.568;	243.257;	428283.910;	53587.396
1; Anderson;	Minor Arterial;	Small Urban (5,000-49,999);	17.291;	36.094;	33061.970;	3394.047
1; Anderson;	Major Collector;	Rural (<5,000);	168.168;	336.336;	219768.015;	37868.108
1; Anderson;	Major Collector;	Small Urban (5,000-49,999);	11.873;	23.746;	15543.430;	776.247
1; Anderson;	Minor Collector;	Rural (<5,000);	140.449;	280.898;	58283.965;	10340.072
1; Anderson;	Local;	Rural (<5,000);	887.707;	1779.625;	97223.355;	3026.794
1; Anderson;	Local;	Small Urban (5,000-49,999);	200.930;	401.860;	19691.140;	630.116

**Figure 3. Sample Section of RIFCREC Data Input File for Years 2006 and Later.**

### TxDOT RIFCREC Data Input File for Toll Roads (/TRIFC)

The /TRIFC identifier (optional, only used for RIFCREC data years 2006 and later) and its corresponding text string specify the path and filename of the TxDOT RIFCREC data input file for toll roads, which should only be used if the semicolon-delimited text file format (i.e., year 2006 and later) is used with the /RIFC identifier. The original TxDOT RIFCREC data file (an Microsoft® Excel spreadsheet) must be converted to a semicolon-delimited text file. This text file should begin with the header line, which indicates the position of each set of data. No other text or data should appear before the header line.

Table 7 shows the required data as it should appear in the header line and Figure 4 shows a sample section of the RIFCREC data for toll roads in this format.

**Table 7. Header Requirements for the TxDOT RIFCREC Data Input File for Toll Roads for 2006 and Later.**

Header Requirement <sup>1</sup>	Description
CO	County Number
PopClass	Area Type
Centerline	Centerline Miles
Lane	Lane Miles
DVM	Daily Vehicle Miles

<sup>1</sup> Capitalization of the header requirements is not required.

DI; DIN;	CO;	CON;	System;	PopClass;	Centerline;	Lane;	DVM;	TVM
2; Fort Worth;	220;	Tarrant;	Off-System;	Urban (>= 5,000);	05.971;	35.826;	557350.080;	00.000
12; Houston;	20;	Brazoria;	Off-System;	Urban (>= 5,000);	00.150;	00.600;	208.500;	00.000
12; Houston;	85;	Galveston;	Off-System;	Urban (>= 5,000);	01.457;	02.914;	842.960;	00.000
12; Houston;	102;	Harris;	Off-System;	Urban (>= 5,000);	93.344;	506.158;	5817628.010;	00.000
18; Dallas;	43;	Collin;	Off-System;	Urban (>= 5,000);	18.609;	111.654;	1940175.706;	00.000
18; Dallas;	57;	Dallas;	Off-System;	Urban (>= 5,000);	29.472;	170.156;	2088213.650;	00.000
18; Dallas;	61;	Denton;	Off-System;	Urban (>= 5,000);	02.215;	13.290;	174209.170;	00.000

**Figure 4. Sample Section of RIFCREC Data Input File for Toll Roads for Years 2006 and Later.**

### County Name and FIPS Code Input File (/CNTY)

The /CNTY identifier (required) and its corresponding text string specify the path and filename of the county name and FIPS code input file. This input file consists of the desired county's three-digit FIPS code and the desired county's name, separated by at least one space, tab, or a combination of tabs and spaces. Figure 5 shows a sample county name and FIPS code input file.

001 ANDERSON
--------------

**Figure 5. Sample County Name and FIPS Code Input File.**

### Year and Day Type Input File (/YRDY)

The /YRDY identifier (required) and its corresponding text string specify the path and filename of the year and day type input file. This file consists of the desired year and day type, separated by at least one space, tab, or a combination of tabs and spaces. Since the day type taken from this input file is used to determine the VMT control total, the day type must appear in this input file

just as it appears in the VMT control total input file. Figure 6 shows a sample year and day type input file.

2005	Wk1_VMT
------	---------

**Figure 6. Sample Year and Day Type Input File.**

### VMT Control Totals Input File (/CTRL)

The /CTRL identifier (required) and its corresponding text string specify the path and filename of the VMT control totals input file. These control totals are used to calculate the VMT factor, which in turn is applied to the AADT VMT from the RIFCREC data. This input file must have a header describing the data beginning in line three (see the following example for the header). Any data or text in the first two lines is ignored. The following data, beginning in line five and separated by at least one space, must be in the VMT control totals input file (Figure 7 shows a sample file):

- Observation number – this is not used by the utility but should be listed;
- Year – the desired year of analysis. The year must be the same as that listed in the year and day type input file;
- County – the county FIPS code;
- County name – the name of the respective county;
- TxDOT district – the TxDOT district where the county is located. This should be the exact district as it appears in the hourly fractions input file; and
- Day type – the desired day type name. The utility can read up to four day types, with each separated by one or more spaces. These day type names should be identical to those listed in the hourly fractions input file and the year and day type input file.

The SAS System					09:14 Monday, October 10, 2005 88		
Obs	Yr	Cty	County	District	Wk1_VMT	Wk2_VMT	Wk3_VMT
1	2005	1	Anderson	Tyler	1,456,212	1,445,548	1,367,805

**Figure 7. Sample VMT Control Totals Input File.**

### Hourly Factors Input File (/HFCT)

The /HFCT identifier (required) and its corresponding text string specify the path and filename of the TxDOT district-level hourly factors. The county’s TxDOT district from the VMT control totals input file is used by the utility to locate the appropriate hourly factors. The hourly factor data for each district must begin with the district name as the text *District=ZZ* where the text *ZZ*

is the district name. For example, the header for the hourly factors for the Abilene District would appear as *District=Abilene*. Before the hourly factors are listed in the input file, another header line should be listed with the first text character being *f*. The day type and 24 hourly factors should then follow, with only one factor on each line. The utility only reads those lines with a day type listed. All other lines are ignored. Figure 8 shows a sample hourly factors input file.

```

The SAS System
16:10 Monday, October 10, 2005

District=Abilene

The MEANS Procedure

Variable      Mean
ffffffffff
Wk1_Hr01      0.015980
Wk1_Hr02      0.013188
Wk1_Hr03      0.011576
Wk1_Hr04      0.011057
Wk1_Hr05      0.012500
Wk1_Hr06      0.016928
Wk1_Hr07      0.029218
Wk1_Hr08      0.048826
Wk1_Hr09      0.048331
Wk1_Hr10      0.051923
Wk1_Hr11      0.055156
Wk1_Hr12      0.057463
Wk1_Hr13      0.058808
Wk1_Hr14      0.061673
Wk1_Hr15      0.063967
Wk1_Hr16      0.066073
Wk1_Hr17      0.068843
Wk1_Hr18      0.072622
Wk1_Hr19      0.060430
Wk1_Hr20      0.047985
Wk1_Hr21      0.041345
Wk1_Hr22      0.035388
Wk1_Hr23      0.028883
Wk1_Hr24      0.021838
ffffffffff

```

**Figure 8. Sample Hourly Factors Input File.**

**Speed Model Data Input File (/SPDM)**

The /SPDM identifier (required) and its corresponding text string specify the path and filename of the speed model parameters input file used to calculate the output speeds. This input file consists of the following data:

- HPMS functional class – for this input file, only functional classes 1 through 3 are listed. Functional class 4 uses the same parameters as functional class 3;
- HPMS area type - area types 1 through 7;
- Hourly lane capacities by HPMS functional class and area type combination;
- Free-flow speeds by functional class and area type combination;
- A parameter for the TTI Speed Model;
- B parameter for the TTI Speed Model; and
- M parameter for the TTI Speed Model.

The data in the speed model parameters input file can be separated by spaces, tabs, or any combination of tabs and spaces. Although the utility allows these to be changed, it is highly suggested that the data in Figure 9 be used.

1	1	2102	70.0	0.015	3.5	1.0
1	2	2102	70.0	0.015	3.5	1.0
1	3	1018	60.0	0.050	3.0	2.0
1	4	1018	50.0	0.050	3.0	2.0
1	5	744	40.0	0.050	3.0	2.0
1	6	744	30.0	0.050	3.0	2.0
1	7	664	30.0	0.050	3.0	2.0
2	1	2271	70.0	0.015	3.5	1.0
2	2	2176	60.0	0.015	3.5	1.0
2	3	954	50.0	0.050	3.0	2.0
2	4	954	40.0	0.050	3.0	2.0
2	5	774	35.0	0.050	3.0	2.0
2	6	774	30.0	0.050	3.0	2.0
2	7	636	30.0	0.050	3.0	2.0
3	1	2328	70.0	0.015	3.5	1.0
3	2	2231	60.0	0.015	3.5	1.0
3	3	753	40.0	0.050	3.0	2.0
3	4	753	35.0	0.050	3.0	2.0
3	5	708	30.0	0.050	3.0	2.0
3	6	708	30.0	0.050	3.0	2.0
3	7	593	30.0	0.050	3.0	2.0

**Figure 9. Suggested Speed Model Input Data File**

(These factors are from *Virtual Link Methodology Review, Refinement, and Update*, TTI, February 2007.)

### Hourly Link VMT and Operational Speeds Output Files (\*.Txx)

Although they are not listed directly in the jcf, the VirtualLinkVMT utility also produces 24 hourly link VMT and operational speeds output files. The path and filenames for these 24 files are created using the path and filename of the listing output file, minus the extension. For each hour, the extension of the listing file is replaced by the text *.Txx* where the *xx* ranges from 1 to 24, depending upon the hour. For example, if the listing output file is defined as



C:\VirtualLinkVMT\output\2007\bpa2007swk.lst, then the hourly link VMT and operational speeds output files would be C:\VirtualLinkVMT\output\2007\bpa2007swk.T01, C:\VirtualLinkVMT\output\2007\bpa2007swk.T02,....., C:\VirtualLinkVMT\output\2007\bpa2007swk.T24. These output files are in the proper format for use with any of the TTI emissions inventory development utilities. The data within each of these output files are (columns are in the following order):

- ANODE – the RIFCREC data area type number, except for area type 4 where the area type is changed to 3;
- BNODE – the RIFCREC data functional class number;
- County number – always 1 since only one county is processed at a time;
- Roadway type – the assigned roadway type based on the area type and functional class numbers (see Table 4);
- Centerline miles – from the RIFCREC data array;
- Calculated congested speed;
- Calculated hourly VMT;
- County name;
- Area type; and
- V/C ratio.

#### **Tab-Delimited Summary Output File (\*.tab)**

The VirtualLinkVMT utility also produces a tab-delimited summary output file. This output file is not listed directly in the jcf. The path and filename for this output file is created using the path and filename of the listing output file, minus the extension and adding the text Summary and the .tab extension. For example, if the listing output file is defined as C:\VirtualLinkVMT\output\2007\bpa2007swk.lst, then the tab-delimited summary output file would be C:\VirtualLinkVMT\output\2007\bpa2007swkSummary.tab. This tab-delimited output file includes a summary of VMT, VHT, and speed (VMT/VHT) by county, hour (including a total for all 24-hours), functional class, and area type.



## TRANSVMT: TRAVEL MODEL VMT AND SPEED ESTIMATION UTILITY

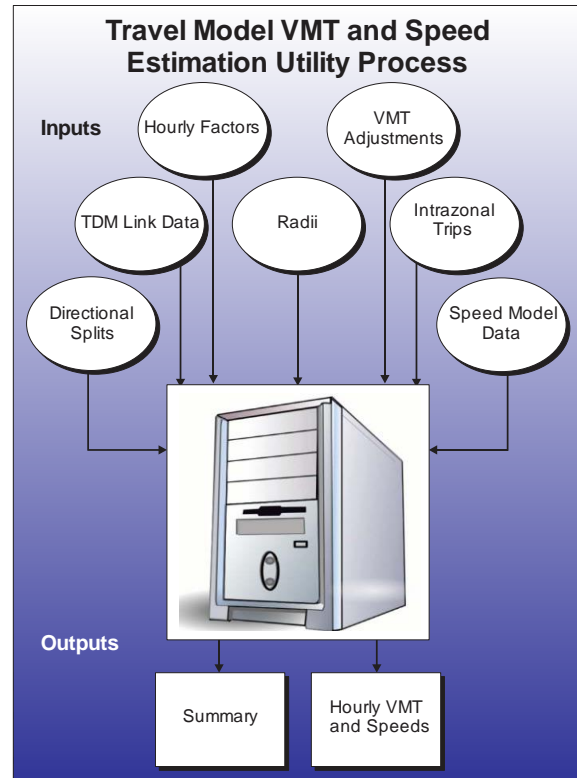
The TRANSVMT utility estimates VMT and estimated operational speeds based on a TDM and trip matrix (or matrices if the assignment consists of multiple time-of-day assignments) for a user-specified day type and season. This utility can be used with 24-hour TDMs as well as four time-period TDMs. For the four time-period TDMs, the utility also has the option of estimating weekend travel based on the TDM and user-defined weekend profile factors. The TRANSVMT utility also has two options for calculating the estimated operational speeds – the TTI Speed Model or the Houston Speed Model.

### OPERATION

The TRANSVMT utility reads the jcf until it reaches the step to run. Each line of the desired step is written to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The jcf input file is then re-opened. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or TRANSVMT cannot open any of these input files due to an incorrect path and/or filename or any other access error, then TRANSVMT results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

As the utility processes the jcf, two main characteristics of the input files are also determined based on the identifiers used in the jcf. The first characteristic is the type of link data input file, either a 24-hour TDM or a four time-period TDM. The default is set to a four time-period TDM. If the /LNK identifier is in the jcf, the utility determines that the link data input file is for a 24-hour TDM. Otherwise, it is assumed that the link data input file is for a four time-period assignment. The other main characteristic is the speed model used to estimate the link



operational speeds. If the /SPDD identifier is in the jcf, then the speed model is set to the TTI Speed Model. If the /SPDH identifier is in the jcf, then the speed model is set to the Houston Speed Model. If neither of these identifiers appears in the jcf, the utility will not operate properly.

The first input file processed by the utility is the header input file. This input file consists of one line, which is saved as the header.

The next input file processed by the utility is the VMT adjustment factors input file. The first line of this input file is used to determine the location of the appropriate data in the input file.

If the TDM input file is a 24-hour TDM (as defined in the jcf by the /LNK identifier), the utility determines the location of the county ID, the HPMS factor, the VMT factor, and the seasonal adjustment factor. If the TDM input file is a four time-period TDM, the utility also determines the location of the AM Peak factor, the Midday factor, the PM Peak factor, and the Overnight factor. If the headers for these time period factors are not found by the utility, then these factors are not saved and no factors time period will be applied. For each line (excluding the header line), the county ID and associated VMT adjustment factors are then saved in the County ID and VMT factors array.

The next input file processed by the utility is the directional splits input file. The first line is used to determine the location of the required data in the input file (area type, road type, AM Peak directional split, Midday directional split, PM Peak directional split, and Overnight directional split). For each remaining line, the appropriate data are saved in the directional splits array.

If the TTI Speed Model is used to estimate link operational speeds, the utility then processes the TTI Speed Model data input file and the delay parameters input file. For the The TTI Speed Model data input file, the first line is used to determine the location of the required data in the input file (area type, road type, speed factor, and capacity factor data for a 24-hour TDM and area type, road type, speed factor, AM Peak capacity factor, Midday capacity factor, PM Peak capacity factor, and the Overnight capacity factor for a four time-period TDM). For each remaining line in the input file, the appropriate data are saved in the capacity and speed factor data array.

For the delay parameters input file, the first line is used to determine the location of the required data (county, road type, A coefficient, B coefficient, and M factor). For each remaining line in the input file, the appropriate data are then saved in the delay parameters array.

If the Houston Speed Model is used to estimate link operational speeds, the utility then processes the Houston Speed Model data input file and the speed reduction factor (SRF) input file. For the Houston Speed Model data input file, the first line is used to determine the location of the required data (area type, road type, SRF area type, SRF functional class, free-flow speed factor, level-of-service [LOS] E speed factor, AM Peak capacity factor, Midday capacity factor, PM Peak capacity factor, and the Overnight capacity factor). For each remaining line in the input file, the appropriate data are saved in the capacity and speed factor data array.

The utility then processes the SRF input file. The first line is used to determine the location of the required data (SRF area type, SRF functional class, and the SRFs for the following v/c ratio factor data in the input file: 0.00, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, and 1.00). For each remaining line in the input file, the appropriate data are saved in the SRF data array.

The utility then processes the time period designations input file. The first line is used to determine the location of the required hour index and the time period (either AM Peak, Midday, PM Peak, or Overnight). For each remaining line in the input file, the appropriate data are saved in the time period designations array.

The utility then processes the centroid connector road type code input file. This input file consists of one line, which is the centroid connector road type code.

The utility then processes the coordinate input file. This input file also consists of one line, which is the coordinate flag (either *Yes* or *No*).

The utility then processes the radii input file. For each line in this input file (consisting of the text "RADII," the zone number, and the zone's respective radii), the zone number and zone's respective radii is saved in the radii array.

If the TDM input file is a 24-hour TDM (as defined in the jcf by the /LNK identifier), the utility then processes the link data fields input file, the link data input file, and the intrazonal trips input file. For the link data fields input file, each line is used to determine the link data field in the link data input file that will be used for the required link data (county code, area type code, road type code, A-Node, B-Node, TAZ, volume, capacity, length, speed, from latitude [optional], from longitude [optional], to latitude [optional], and to longitude [optional] data), with the required data and associated link data field saved in the link data fields array.

For the link data input file, the utility uses the data in the link data fields array and the dictionary file that must accompany the link data input file, to determine the location of the required link data in the link data input file. For each line in the link data input file, the utility saves the required data listed previously in the link data array. As each line is read, the VMT for each link is also calculated and added to the link data VMT array for summary purposes later in the process. If TRANSVMT has been directed to report the node coordinate data (coordinate flag of *Yes* in the coordinate input file), the utility also saves the longitude and latitude for both the ANODE and BNODE in the coordinate array. After processing the entire TDM link data input file, the utility then writes the node coordinate data to the node coordinate data output file, if TRANSVMT has instructions to write the data to the file.

For the intrazonal trips input file, each line is used to determine the zone and the zone's respective intrazonal trips. The intrazonal trips are saved in the intrazonal data array according to the zone number related to the intrazonal trips.

For each zone, the utility then calculates the intrazonal VMT for that zone. The utility calculates the average speed of the centroid connector within the zone and saves the data in the

intrazonal data array, along with the county code where the zone resides. Then, using the intrazonal trips (from the intrazonal data array) and the radii (from the radii array) and the average speed of the centroid connectors within the given zone, the utility calculates the intrazonal VMT and saves that data in the intrazonal data array. The VMT is also saved in the link data VMT array for summary purposes. For a more detailed description of the calculation, see the “Time-of-Day Volume and VMT Estimation Procedure” in the following section. A summary of the link data VMT and the intrazonal VMT by county is calculated and written to the listing output file.

For the four time-period TDM, the same process detailed previously is followed for the AM Peak, Midday, PM Peak, and Overnight set of input files (link data fields input file, link data input file and intrazonal trips input file). The main difference in the processes is that the data are saved in arrays respective to the time period for the four time-period TDM (i.e., the AM Peak link data are saved in the AM Peak link data array, the Midday link data are saved in the Midday link data array, etc.). As each time period is processed, the summary of the time period link data VMT and intrazonal VMT by county is calculated and written to the listing file, as well as the node coordinates to the respective node coordinates output file.

The utility then processes the hourly factors input file. The first line is used to determine the location of the required data (hour index and the hourly factor data). For each remaining line in the input file, the appropriate data are saved in the hourly factors array. For a 24-hour TDM, the sum of the hourly factors is calculated. If this sum does not equal to 1, the largest fraction is adjusted and the utility writes a warning message to the listing file. For the four time-period TDM, the sum of the hourly factors for each time period (as designated in the time period designations array) is calculated and the largest time period fraction is adjusted when the sum of the time period hourly factors does not sum to 1.

The utility then calculates the VMT and operational speed for each link in the link data array (or arrays) and the intrazonal VMT for each zone. The following sections describe the link level VMT and speed calculation.

### **Time-of-Day Volume and VMT Estimation Procedure**

The directional volumes and VMT for each link are calculated as:

$$\begin{aligned}
 \text{VOL1(A,B)} &= \text{VOL24(A,B)} \times \text{HPMFAC(CNTY)} \times \text{SEAFAC(CNTY)} \times \\
 &\quad \text{PERFAC(HR)} \times \text{VMTFAC(CNTY)} \times (\text{SPLIT(AT,FC)}/100.0) \\
 \text{VOL2(A,B)} &= \text{VOL24(A,B)} \times \text{HPMFAC(CNTY)} \times \text{SEAFAC(CNTY)} \times \\
 &\quad \text{PERFAC(HR)} \times \text{VMTFAC(CNTY)} \times ((100- \\
 &\quad \text{SPLIT(AT,FC)}/100.0) \\
 \text{VMT1(A,B)} &= \text{VOL1(A,B)} \times \text{LENGTH(A,B)} \\
 \text{VMT2(A,B)} &= \text{VOL2(A,B)} \times \text{LENGTH(A,B)}
 \end{aligned}$$

Where:

A,B	=	the A-Node and B-Node of the link obtained from the link data;
CNTY	=	the county number obtained from the link data;
AT	=	the area type number obtained from the link data;
FC	=	the functional classification (or facility type) number obtained from the link data;
LENGTH(A,B)	=	the link length for link A,B in miles obtained from the link data;
HR	=	the hour of the day;
VOL24(A,B)	=	the link's 24-hour assigned volume for link A,B obtained from the link data;
VOL1(A,B)	=	the estimated time-of-day volume in one direction;
VOL2(A,B)	=	the estimated time-of-day volume in the other direction. This volume is calculated only if the assignment is non-directional (i.e., the directional split factor is not 100.00);
VMT1(A,B)	=	the estimated time-of-day VMT in one direction;
VMT2(A,B)	=	the estimated time-of-day VMT in the other direction. This VMT is calculated only if the assignment is non-directional (i.e., the directional split factor is not 100.00);
HPMFAC(CNTY)	=	the HPMS factor for CNTY;
SEAFAC(CNTY)	=	the seasonal adjustment factor for CNTY;
PERFAC(HR)	=	the time-of-day factor applied to the 24-hour volume to estimate the volume for the time period;
SPLIT(AT,FC)	=	the directional split factor specifying the percentage of the time-of-day volume expected to occur in one direction. If the assignment is directional, the directional split factor should be 100;
100-SPLIT(AT,FC)	=	specifies the percentage of the time-of-day volume expected to occur in the other direction. If the assignment is directional, this value will be 0; and
VMTFAC(CNTY)	=	the VMTFAC factor for CNTY.

For the four time-period TDM and Weekend day type combination, the weekend profile factor is included in the VOL1(A,B) and VOL2(A,B) calculations. For this utility, the VOL1(A,B) and VOL2(A,B) equations are expressed as:

$$\text{VOL1(A,B)} = \text{VOL24(A,B)} \times \text{HPMFAC(CNTY)} \times \text{SEAFAC(CNTY)} \times \text{PERFAC(HR)} \times \text{VMTFAC(CNTY)} \times \text{TPFACT(CNTY)} \times (\text{SPLIT(AT,FC)}/100.0)$$

$$\text{VOL2(A,B)} = \text{VOL24(A,B)} \times \text{HPMFAC(CNTY)} \times \text{SEAFAC(CNTY)} \times \text{PERFAC(HR)} \times \text{VMTFAC(CNTY)} \times \text{TPFACT(CNTY)} \times ((100-\text{SPLIT(AT,FC)})/100.0)$$

Where:

TPFACT(CNTY) = the appropriate time period factor for CNTY.

### TTI Speed Model

The TTI Speed Model estimates the directional time-of-day speeds using the directional v/c ratio on a link to calculate the average delay per mile (in minutes) due to the level of congestion reflected in the v/c ratio. The congested directional speed is calculated using the estimated directional delay and the estimated free-flow speed. This process is used for all links excluding the centroid connectors and the created intrazonal links. The following provides a more detailed description of the congested speed estimation process.

The directional v/c ratios and free-flow speed are calculated for a non-directional assignment as:

$$VC1(A,B) = VOL1(A,B) / (CAP24_{ndir}(A,B) \times CAPFAC(AT,FC) \times 0.5)$$

$$VC2(A,B) = VOL2(A,B) / (CAP24_{ndir}(A,B) \times CAPFAC(AT,FC) \times 0.5)$$

$$FSPD(A,B) = SPD24 \times SPDFAC(AT,FC)$$

Where:

- A,B = the A-Node and B-Node of the link obtained from the link data;
- AT = the area type number obtained from the link data;
- FC = the functional classification (or facility type) number obtained from the link data;
- VC1(A,B) = the estimated time-of-day v/c ratio in one direction;
- VC2(A,B) = the estimated time-of-day v/c ratio in the other direction;
- VOL1(A,B) = the estimated time-of-day volume in one direction;
- VOL2(A,B) = the estimated time-of-day volume in the other direction;
- CAP24<sub>ndir</sub>(A,B) = the link's 24-hour non-directional capacity from the assignment data set;
- CAPFAC(AT,FC) = the analyst-supplied factor used to estimated time-of-day non-directional capacity from the 24-hour non-directional capacity. Half of the non-directional time-of-day capacity is used for each direction;
- FSPD(A,B) = estimated free-flow speed on link A,B. The free-flow speed is assumed to be the same in both directions for a non-directional assignment;
- SPD24(A,B) = the input speed for the link data (i.e., the 24-hour input link-data speed); and
- SPDFAC(AT,FC) = the analyst-supplied factor used to estimate time-of-day free-flow speed from the input link-data speed.



For directional assignments, the same process discussed previously is used except only one volume and one v/c ratio exist. Since the capacity for the link is also directional, the capacity is not split in half. For a directional assignment, the v/c ratio is calculated as:

$$VC1(A,B) = VOL1(A,B) / (CAP24dir(A,B) \times CAPFAC(AT,FC))$$

$$FSPD(A,B) = SPD24 \times SPDFAC(AT,FC)$$

Where:

CAP24dir(A,B) = the link's 24-hour directional capacity from the assignment data set; and

FSPD(A,B) = SPD24 × SPDFAC(AT,FC).

The directional delay (in minutes per mile) due to congestion is calculated using a volume-delay equation for each link and each direction (or the only direction for a directional assignment). The following is the general form of the volume-delay equation used in the model:

$$Delay1(A,B) = \text{Min}[Ae^{B\{VC1(A,B)\}}, M]$$

$$Delay2(A,B) = \text{Min}[Ae^{B\{VC2(A,B)\}}, M]$$

Where:

A,B = the A-Node and B-Node of the link obtained from the link data;

Delay1(A,B) = congestion delay (in minutes/mile) for one direction;

Delay2(A,B) = congestion delay (in minutes/mile) for the other direction (non-directional assignment only);

A & B = volume-delay equation coefficients;

M = maximum minutes of delay per mile;

VC1(A,B) = the estimated time-of-day v/c ratio in one direction; and

VC2(A,B) = the estimated time-of-day v/c ratio in the other direction.

The delay model parameters (A, B, and M) were developed for the Dallas/Fort Worth area and verified by application in other Texas urban areas. Table 8 shows the volume-delay equation parameters.

**Table 8. Volume-Delay Equation Parameters.**

Parameters	Parameter Values	
	High-Capacity Facilities	Low-Capacity Facilities
A	0.015	0.050
B	3.500	3.000
M	5.000	10.000

After the delay is calculated for each direction, the congested speed is then calculated for each direction. As with the delay, only one congested speed is calculated for a directional assignment. The congested speed is calculated for each direction when a non-directional assignment is used. The directional congested speed can be calculated as:

$$\text{CSPD1(A,B)} = 60 / (\{60/\text{FSPD}\} + \text{Delay1(A,B)})$$

$$\text{CSPD2(A,B)} = 60 / (\{60/\text{FSPD}\} + \text{Delay2(A,B)})$$

Where:

- A,B = the A-Node and B-Node of the link obtained from the link data;
- CSPD1(A,B) = the congested speed for one direction;
- CSPD2(A,B) = the congested speed for the other direction (non-directional assignment only);
- FSPD(A,B) = the estimated free-flow speed on link A,B. The free-flow speed is assumed to be the same in both directions for a non-directional assignment;
- Delay1(A,B) = the delay in one direction; and
- Delay2(A,B) = the delay in the other direction (non-directional assignment only).

Capacity data are not used for centroid connectors. For centroid connectors, the operational speed (i.e., congested speed) is assumed to be the speed input from the link data.

### **Houston Speed Model**

The original Houston Speed Model is based on the speed estimation procedures suggested in a report, *Highway Vehicle Speed Estimation Procedures for Use in Emissions Inventories* (a draft report prepared for the U.S. Environmental Protection Agency [EPA] by Cambridge Systematics Inc., September 1991). The original Houston Speed Model is described in the technical memorandum, *Implementation and Calibration of a Speed Model for the Houston-Galveston Region*, prepared by TTI for the Houston-Galveston Area Council (H-GAC), March 1993. The model approach used to estimate freeway speeds in the original Houston Speed Models could be described as an SRF approach. This approach is used for Freeways, Arterials, and Collectors.

Using the SRF approach requires estimates of both the free-flow speed (i.e., the speed at a v/c ratio approaching 0) and the LOS E speed (i.e., the LOS E speed or the speed at a v/c ratio of 1.0). The analyst provides pairs of speed factors for each functional class and area type that can be applied to the link-data input speed to estimate a link's free-flow speed and LOS E speed. The analyst-supplied SRFs describe the general shape of the speed curve for v/c ratios varying from 0.0 to 1.0. These estimate the speeds for v/c ratios between 0.0 and 1.0. The extensions of the models for v/c ratios exceeding 1.0 are based on the traditional Bureau of Public Records (BPR) impedance adjustment function. The following provides a more detailed description of the congested speed estimation process.

The directional v/c ratios, free-flow speeds, and LOS E speeds for a non-directional assignment are calculated as:

$$VC1(A,B) = VOL1(A,B) / (CAP24(A,B) \times CAPFAC(AT,FC) \times 0.5)$$

$$VC2(A,B) = VOL2(A,B) / (CAP24ndir(A,B) \times CAPFAC(AT,FC) \times 0.5)$$

$$SPD0(A,B) = SPD24 \times SPD0FAC(AT,FC)$$

$$SPD1(A,B) = SPD24 \times SPD1FAC(AT,FC)$$

Where:

- A,B = the A-Node and B-Node of the link obtained from the link data;
- AT = the area type number obtained from the link data;
- FC = the functional classification number obtained from the link data;
- VC1(A,B) = the estimated time-of-day v/c ratio in one direction;
- VC2(A,B) = the estimated time-of-day v/c ratio in the other direction. If the assignment is directional, the VC2 will be 0.0;
- VOL1(A,B) = the estimated time-of-day volume in one direction;
- VOL2(A,B) = the estimated time-of-day volume in the other direction. If the assignment is directional, the VOL2(A,B) will be 0.0;
- CAP24ndir(A,B) = the link's 24-hour non-directional capacity from the assignment data set;
- CAPFAC(AT,FC) = the analyst-supplied factor used to estimate time-of-day non-directional capacity from the 24-hour non-directional capacity. Half of the non-directional time-of-day capacity is used for each direction;
- SPD0(A,B) = estimated free-flow speed on link A,B;
- SPD1(A,B) = estimated LOS speed (i.e., the expected speed at a v/c ratio of 1.0) on link A,B;
- SPD24(A,B) = the input speed for the link data (i.e., the 24-hour input link-data speed);
- SPD0FAC(AT,FC) = the analyst-supplied factor used to estimate time-of-day free-flow speed from the input link-data speed; and
- SPD1FAC(AT,FC) = the analyst-supplied factor used to estimate time-of-day LOS E speed from the link-data input speed.

For directional assignments, the same process discussed previously is used except only one volume and one v/c ratio exist. Since the capacity for the link is also directional, the capacity is not split in half. For a directional assignment, the v/c ratio is calculated as:

$$VC1(A,B) = VOL1(A,B) / (CAP24dir(A,B) \times CAPFAC(AT,FC))$$

Where:

- CAP24dir(A,B) = the link's 24-hour directional capacity from the assignment data set.

The speed factors are applied to the link's TDM coded speed to estimate the link's free-flow speed (i.e., the speed for a v/c ratio approaching 0.0) and the LOS E speed (i.e., the speed for a v/c ratio of 1.0). The SRFs, which essentially describe the shape of the speed curve, are by area type and functional group. The factors are input for v/c ratios from 0.0 to 1.0 in increments of 0.05. The analyst-supplied SRFs describe the decay from a free-flow speed to a LOS E speed for a v/c ratio of 1.0. The values of the SRFs vary from 0.0 to 1.0. Table 9 provides an example of a set of SRFs for a freeway. The SRFs for v/c ratios between these points are estimated by linear interpolation.

**Table 9. Sample Set of Speed Reduction Factors.**

V/C Ratio	Speed Reduction Factor	V/C Ratio	Speed Reduction Factor
0.00	0.00000	0.55	0.00735
0.05	0.00005	0.60	0.00860
0.10	0.00010	0.65	0.00930
0.15	0.00025	0.70	0.01000
0.20	0.00040	0.75	0.06750
0.25	0.00095	0.80	0.12500
0.30	0.00150	0.85	0.27250
0.35	0.00250	0.90	0.42000
0.40	0.00350	0.95	0.60000
0.45	0.00480	1.00	1.00000
0.50	0.00610		

The speed model (for v/c ratios from 0.00-to-1.00) may be described as:

$$SV/C = S0.0 - SRFV/C \times (S0.0 - S1.0)$$

Where:

- SV/C = estimated directional speed for the forecast v/c ratio on the link in the selected direction;
- S0.0 = estimated free-flow speed for the v/c ratio equal to 0.0;
- S1.0 = estimated LOS E speed for the v/c ratio equal to 1.0;
- SRFV/C = SRF for the forecast v/c ratio; and
- V/C = the forecast v/c ratio on the link. The v/c ratio can be 0.0 to 1.0.

Because traffic assignments can produce v/c ratios greater than 1.0, a model extension similar to that used in the Houston Speed Model is used. The extension is based on the BPR model. For links with a v/c ratio greater than 1.0 and less than 1.5, the following model extension is used to estimate the link's speed:

$$SV/C = S1.0 \times (1.15 / (1.0 + (0.15 \times (V/C)^4)))$$

Where:

- SV/C = estimated directional speed for the forecast v/c ratio on the link in the selected direction;
- S1.0 = estimated LOS E speed for the v/c ratio equal to 1.0; and
- V/C = the forecast v/c ratio on the link. The v/c ratio can be 1.0 to 1.5.

For v/c ratios greater than 1.5, the speed is calculated using the model extension shown above for the v/c ratio of 1.5.

Capacity data are not used for centroid connectors and intrazonal links. Thus, for local streets, which these represent, the free-flow speed factors and LOS E speed factors should be defined as 1.0, and the speed reduction factors should be set to 0 for all v/c entries. The operational speed (i.e., assignment speed) for centroid connectors is assumed to be the speed input from the link data.

### Estimating Intrazonal Travel and Speeds

Intrazonal travel cannot be estimated only from the TDM, since intrazonal trips are not assigned to the network. The TRANSVMT utility outputs a link record (for subsequent input to EMSCALC) for the intrazonal trips for each zone. For the intrazonal link output, the zone number is used for both the A-Node and B-Node.

In the speed and VMT summaries, the intrazonal travel is summarized as a separate functional classification code. In the hourly link output, the intrazonal records are given a functional classification code of 40.

To calculate the intrazonal VMT, the intrazonal speed and average trip length are calculated. The intrazonal speed is estimated by averaging the speeds on the zone's centroid connectors. The average trip length and intrazonal VMT are calculated using the speed and average trip length in minutes as:

$$ATL2 = (SPD/60) \times ATL1$$

$$VMT(\text{intra}) = INTRA \times ATL2 \times HPMFAC(\text{CNTY}) \times SEAFAC(\text{CNTY}) \times PERFAC(\text{HR}) \times VMTFAC(\text{CNTY})$$

Where:

- ATL2 = average trip length (in miles);
- SPD = the estimated intrazonal speed;
- ATL1 = average trip length (in minutes). This average trip length is assumed to be equal to the zone's radii value used in the trip distribution;
- INTRA = number of intrazonal trips for the selected zone for the time-of-day and season. The total number of intrazonal trips for the zone is read from the assignment trip table; and
- VMT(intra) = VMT for the zone.

For the four time-period TDM and Weekend day type combination, the weekend profile factor is included in the VMT(intra) calculation, but the average trip length (ATL2 above) remains the same. This VMT(intra) calculation can be expressed as:

$$\text{VMT(intra)} = \text{INTRA} \times \text{ATL2} \times \text{HPMFAC(CNTY)} \times \text{SEAFAC(CNTY)} \times \text{PERFAC(HR)} \times \text{VMTFAC(CNTY)} \times \text{WEPROF(CNTY)}$$

## JCF INPUT FILE

The jcf input file directs the TRANSVMT utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (TRANSVMT in this case), or the path and filename of the input or output file.

Table 10 shows the file type identifiers. Any other characters at the beginning of a line in the jcf input file are not recognized by the utility and do not affect the execution of the utility.

**Table 10. TRANSVMT File Type Identifiers.**

<b>Identifier</b>	<b>Description</b>	<b>Text String Description</b>
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/HDR	Header Input File	Path and Filename
/HRF	Hourly Factors Input File	Path and Filename
/CCRC	Centroid Connector Road Type Code Input File	Path and Filename
/SPLT	Directional Splits Input File	Path and Filename
/VADJ	VMT Adjustment Factors Input File	Path and Filename
/RADI	Intrazonal Radii Input File	Path and Filename
/TMPD	Time Period Designations Input File	Path and Filename
/LDF	24-Hour TDM Link Data Fields Input File <sup>2</sup>	
/LDFA, /LDFM, /LDFP, /LDFO	AM Peak, Midday, PM Peak, and Overnight TDM Link Data Fields Input Files <sup>3</sup> , respectively	Path and Filename
/LNK	24-Hour TDM Link Data Input File <sup>2</sup>	Path and Filename
/LNKA, /LNKM, /LNKP, /LNKO	AM Peak, Midday, PM Peak, and Overnight TDM Link Data Input Files <sup>3</sup> , respectively	Path and Filename
/MTX	24-Hour TDM Intrazonal Trips Input File <sup>2</sup>	Path and Filename
/MTXA, /MTXM, /MTXP, /MTXO	AM Peak, Midday, PM Peak, and Overnight TDM Intrazonal Trips Input Files <sup>3</sup> , respectively	Path and Filename
/SPDD	TTI Speed Model Data Input File <sup>4</sup>	Path and Filename
/DELP	Delay Parameters Input File <sup>4</sup>	Path and Filename
/SPDH	Houston Speed Model Data Input File <sup>5</sup>	Path and Filename
/SRFD	SRF Input File <sup>5</sup>	Path and Filename
/CRD	Coordinates Input File	Path and Filename
/TMSO	Travel Model Summary Output File <sup>6</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only used for reference purposes and not accessed by the utility.

<sup>2</sup> Only used for 24-hour TDMs.

<sup>3</sup> Only used for four time-period TDMs.

<sup>4</sup> Only used with the TTI Speed Model.

<sup>5</sup> Only used with the Houston Speed Model.

<sup>6</sup> Optional.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (TRANSVMT), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 10 shows a sample jcf for a 24-hour TDM using the TTI Speed Model, followed by a sample jcf for four time-period TDM using the Houston speed model in Figure 11.

```

/JOB      TRANSVMT
/STEP     Step1 TRANSVMT
/LST      C:\transvmt\output\2007\bpa2007swk.lst
/TMSO     C:\transvmt\output\2007\bpa2007swk_TDMsummary.tab
/HDR      C:\transvmt\input\bpa07mtp_2007_Header.txt
/HRF      C:\transvmt\input\bpa07mtp_HOURLYFACTORS.txt
/DELP     C:\transvmt\input\BPA_DELAY_FY07.txt
/CCRC     C:\transvmt\input\BPA_ccCode.txt
/SPLT     C:\transvmt\input\BPA_DirSplits_fy05.txt
/TMPD     C:\transvmt\input\bpa_TimePerDesignation.txt
/VADJ     C:\transvmt\input\bpa07mtp_COUNTYVMTFACFS.txt
/SPDD     C:\transvmt\input\BPA_fy05_Speed&capfactor.txt
/LNK      C:\transvmt\input\TDMs\2007\Network\Out\bpa07Network_TRANSVMT.asc
/RADI     C:\transvmt\input\TDMs\2007\Radii\bpa07rad.asc
/MTX      C:\transvmt\input\TDMs\2007\TripMatrix\BPA2007IntraTrips.txt
/CRD      C:\transvmt\input\Coordinates.txt
/LNKF     C:\transvmt\input\TDMs\2007\Network\Out\LinkDataFields.txt

```

**Figure 10. Sample 24-hour TDM TRANSVMT jcf.**



```

/JOB      TRANSVMT
/STEP     Step1 TRANSVMT
/LST      C:\transvmt\output\2007\hgb2007swk.lst
/TMSO     C:\transvmt\output\2007\hgb2007swk_TDMsummary.tab
/HDR      C:\transvmt\input\hgb_2007swk_Header.txt
/HRF      C:\transvmt\input\HOURLYFACTORS.txt
/CCRC     C:\transvmt\input\ccCode.txt
/SPLT     C:\transvmt\input\DirSplits.txt
/TMPD     C:\transvmt\input\TimePerDesignation.txt
/VADJ     C:\transvmt\input\COUNTYVMFACTS.txt
/SPDH     C:\transvmt\input\HGB_CAPFACT_SPDFACT.txt
/LNKA     C:\transvmt\input\TDMs\2007\Network\Out\hgb07Network_AMPK_TRANSVMT.asc
/LNKM     C:\transvmt\input\TDMs\2007\Network\Out\hgb07Network_MDAY_TRANSVMT.asc
/LNKP     C:\transvmt\input\TDMs\2007\Network\Out\hgb07Network_PMPK_TRANSVMT.asc
/LNKO     C:\transvmt\input\TDMs\2007\Network\Out\hgb07Network_OVN_TRANSVMT.asc
/RADI     C:\transvmt\input\TDMs\2007\Radii\hgb07rad.asc
/MTXA     C:\transvmt\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_AMPK.txt
/MTXM     C:\transvmt\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_MDAY.txt
/MTXP     C:\transvmt\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_PMPK.txt
/MTXO     C:\transvmt\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_OMPK.txt
/CRD      C:\transvmt\input\Coordinates.txt
/SRFD     C:\transvmt\input\HGAC_srfdata.txt
/LDFA     C:\transvmt\input\TDMs\2007\Network\Out\LinkDataFields_AMPK.txt
/LDFM     C:\transvmt\input\TDMs\2007\Network\Out\LinkDataFields_MDAY.txt
/LDFP     C:\transvmt\input\TDMs\2007\Network\Out\LinkDataFields_PMPK.txt
/LDFO     C:\transvmt\input\TDMs\2007\Network\Out\LinkDataFields_OVN.txt

```

**Figure 11. Sample Four Time-Period TDM TRANSVMT jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. This identifier is not used by the utility but is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. This text string is not used by the TRANSVMT utility, but only by MOVES2014aUtl to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part being separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file basically summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- JCF through the step being executed;
- Path and filename of all input files;
- TDM and intrazonal VMT summary;
- Begin and end time of each hourly VMT and speed calculation step;
- Hourly VMT and speed (VMT/VHT) summary for each county;
- 24 hour VMT and speed (VMT/VHT) summary; and
- Date and time of the end of the utility execution.

### **Header Input File (/HDR)**

The /HDR identifier (required) and its corresponding text string specify the path and filename of the header input file. This text file lists the header that is written to the listing file to assist in naming the specific TRANSVMT execution. Figure 12 shows a sample header input file.

BPA 2007 - bpa07mtp

**Figure 12. Sample Header Input File.**

### **Hourly Factors Input File (/HRF)**

The /HRF identifier (required) and its corresponding text string specify the path and filename of the hourly factors input file. This text file lists each hour and corresponding hourly factor, which is used to distribute the VMT over each hour of the day. The first line of this input file is a header consisting of:

- ID – hour ID of 1, 2, ..., 24 representing each hour of the day (hour 1 is 12:00 a.m.-to-12:59 a.m., hour 2 is 1:00 a.m.-to-1:59 a.m., etc.); and
- FACTOR – the hourly factor for the corresponding hour.

For the header line and subsequent data, the hour listed in the tab-delimited file and the corresponding hour index can be separated by spaces, tabs, or any combination of spaces and tabs. All 24 hours must appear in the hourly factors input file. If the sum of the hourly factors does not sum to 1 (all factors for a 24-hour TDM or by time-period for a four time-period TDM), the largest factor is adjusted so that the sum equals 1. Figure 13 shows a sample hourly factors input file.

ID	FACTOR
01	0.011604
02	0.008800
03	0.007727
04	0.007389
05	0.011695
06	0.024937
07	0.044381
08	0.058250
09	0.049817
10	0.048140
11	0.051565
12	0.056114
13	0.059230
14	0.060272
15	0.062415
16	0.069199
17	0.073674
18	0.076507
19	0.058359
20	0.045415
21	0.038476
22	0.032995
23	0.024646
24	0.018393

**Figure 13. Sample Hourly Factors Input File.**

### **Centroid Connector Road Type Code Input File (/CCRC)**

The /CCRC identifier (required) and its corresponding text string specify the path and filename of the centroid connector road type code input file. This text file lists the road type code of the centroid connectors as it appears in the link data. Figure 14 shows a sample centroid connector road type code input file.



**Figure 14. Sample Centroid Connector Road Type Code Input File.**

### **Directional Split Factors Input File (/SPLT)**

The /SPLT identifier (required) and its corresponding text string specify the path and filename of the directional split factors input file. This text file lists the directional split factors by area type/road type combination for all four time periods (AM Peak, Midday, PM Peak, and Overnight). The first line of this input file is a header row consisting of:

- ATYPE – area type;
- RDTYPE – road type code;
- AMPKSPLT – AM Peak directional split factor;
- MDAYSPLT – Midday directional split factor;
- PMPKSPLT – PM Peak directional split factor; and
- OVNSPLT – Overnight directional split factor.

The order of the columns of data does not matter as long as the data are separated by spaces, tabs, or any combination of spaces and tabs. Each area type/road type combination in the link data must appear in the directional split factors input file. The directional split factors should be listed as non-percentage numbers (i.e., a directional split of 50 percent would result in a directional split factor 50). If the TDM is directional (i.e., a link appears in the link data for both directions), then the directional split factors would be set at 100. Figure 15 shows a sample directional split factor input file.

ATYPE	RDTYPE	AMPKSPLT	MDAYSPLT	PMPKSPLT	OVNSPLT
01	01	50.00	51.00	54.00	51.00
01	02	50.00	51.00	54.00	51.00
01	03	50.00	51.00	54.00	51.00
01	04	50.00	51.00	54.00	51.00
01	05	50.00	51.00	54.00	51.00

**Figure 15. Sample Directional Splits Input File.**

### **VMT Adjustment Factors Input File (/VADJ)**

The /VADJ identifier (required) and its corresponding text string specify the path and filename of the VMT adjustment factors input file, which lists each county in the link data input file and its respective VMT adjustment factors. The first line of this text file is header line consisting of:

- COUNTYID - county number as shown in the link data;
- HPMSFACT - HPMS factor. Each county must have its own HPMSFACT, even if the values are all the same. If the user does not wish to apply an HPMS factor, then use a value of 1.0 for this input;
- VMFACT - VMT factor. Each county must have its own VMFACT, even if the values are all the same. If the user does not wish to apply a VMT factor, then use a value of 1.0 for this input;
- SEAFAC - seasonal adjustment factor. Each county must have its own SEAFAC, even if the values are all the same. If the user does not wish to apply a seasonal adjustment factor, then use a value of 1.0 for this input;

- AMPKFACT – AM Peak factor. This column should only appear if processing a four time-period assignment where the AM Peak assignment has a unique VMT factor;
- MDAYFACT – Midday factor. This column should only appear if processing a four time-period assignment where the Midday assignment has a unique VMT factor;
- PMPKFACT – PM Peak factor. This column should only appear if processing a four time-period assignment where the PM Peak assignment has a unique VMT factor ;
- OVNFACT – Overnight factor. This column should only appear if processing a four time-period assignment where the Overnight assignment has a unique VMT factor; and
- COUNTYNAME - county name corresponding to the appropriate COUNTYID (not used by the utility, only listed as a comment and must appear as the last column of data).

The order of the columns does not matter (except for the *COUNTYNAME*, which must appear as the last column of data), as long as the user separates the data within this input file by spaces, tabs, or any combination of spaces and tabs. The user must list each county in the link data input in the VMT adjustments input file or the utility will not function properly. Figure 16 shows a sample VMT adjustments input file.

COUNTYID	HPMSFACT	VMFACT	SEAFAC	COUNTYNAME
01	1.047589089	1.000000000	1.10424	Jefferson
02	1.060350825	1.000000000	1.10424	Orange
03	0.945155488	1.000000000	1.10424	Hardin
04	1.000000000	1.000000000	1.00000	Other

**Figure 16. Sample VMT Adjustments Input File.**

### **Intrazonal Radii Input File (/RADI)**

The /RADI identifier (required) and corresponding text string specify the path and file name of the intrazonal radii input file. This text file lists the text RADII, the zone number, and the intrazonal radii. This file is the exact output from the RADICALC utility used to calculate the radii. Figure 17 shows a sample intrazonal radii input file.

RADII	1	4.6400
RADII	2	2.3600
RADII	3	1.7400
RADII	4	3.5000
RADII	5	3.5300
RADII	6	4.8100
RADII	7	4.0600
RADII	8	2.2900
RADII	9	0.8500
RADII	10	0.9600

**Figure 17. Sample Intrazonal Radii Input File.**

### **Time Period Designation Input File (/TMPD)**

The /TMPD identifier (required) and its corresponding text string specify the path and filename of the time period designation input file, which assigns each hour of the day a time period of either AM Peak, Midday, PM Peak, or Overnight. The time periods must be spelled as shown or the utility will not operate properly. The first line of this input file is the header row consisting of:

- TIMEPER – the hours of the day; and
- TIMEPERIOD – the time period for the respective hour, either AM Peak, Midday, PM Peak, or Overnight.

The columns can be in any order, as long as the header and the data following the header are separated by spaces, tabs, or any combination of tabs and spaces. Each hour of the day must be listed in this input file but the order of the hours does not matter. For this utility, the hours of the day should be listed as 1, 2, 3, ..., 24, with hour 1 being 12:00 a.m.-to-12:59 a.m., hour 2 being 1:00 a.m.-to-1:59 a.m., etc. Figure 18 shows a sample time period designation input file.

TIMEPER	TIMEPERIOD
01	Overnight
02	Overnight
03	Overnight
04	Overnight
05	Overnight
06	Overnight
07	Overnight
08	AM Peak
09	Midday
10	Midday
11	Midday
12	Midday
13	Midday
14	Midday
15	Midday
16	Midday
17	Midday
18	PM Peak
19	Overnight
20	Overnight
21	Overnight
22	Overnight
23	Overnight
24	Overnight

**Figure 18. Sample Time Period Designation Input File.**

### Coordinate Input File (/CRD)

The /CRD identifier (required) and its corresponding text string specify the path and filename of the coordinate input file, which directs TRANSVMT on the option of reporting the node coordinates. Either the text *Yes* or *No* must appear in the first line. Although capitalization does not matter, any other text will result in an error and TRANSVMT will terminate the current execution. Since the lines following the first line are not read by TRANSVMT, these lines may be used as comments. If *Yes* is listed in the first line, utility will search the link data input file for the node coordinate data and report the node coordinates to the node coordinates output file. If *No* is listed, the node coordinates will not be reported and are not required in the link data input file. Figure 19 shows a sample coordinate input file.

Yes * Value must be "Yes" or "No" or utility will terminate.
---

**Figure 19. Sample Coordinate Input File.**

## Link Data Fields Input File (/LDF, /LDFA, /LDFM, /LDFP, /LDFO)

The /LDF, /LDFA, /LDFM, /LDFP, and /LDFO identifiers and corresponding text strings specify the 24-hour TDM link data fields input file, the four time-period AM Peak link data fields input file, the four time-period Midday link data fields input file, the four time-period PM Peak link data fields input file, and the four time-period Overnight link data fields input file, respectively. For a 24-hour TDM, only the /LNK identifier (required) should be used in the jcf. For a four time-period TDM, only the /LNKA, /LNKM, /LNKP, and /LNKO identifiers (required) should be used. All five of these input files have the same format, a text file listing each set of required data and associated link data field in the link data input file. The required data consists of:

- ANODE - the A-Node (or begin node) of the link;
- BNODE - the B-Node (or end node) of the link;
- County - the county code where the link resides;
- Area Type - the area type code for the respective link;
- Roadtype - the road type code for the respective link;
- Volume - the volume for the respective link;
- Capacity - the capacity for the respective link;
- Speed - the speed to be input into the speed model;
- TAZ - the associated zone for the respective link;
- Length - the length of the respective link;
- From Latitude – latitude of ANODE (optional, only if node coordinates are to be output);
- From Longitude – longitude of ANODE (optional, only if node coordinates are to be output);
- To Latitude – latitude of BNODE (optional, only if node coordinates are to be output);  
and
- To Longitude – longitude of BNODE (optional, only if node coordinates are to be output).

Only one set of required data can appear on each line, with the associated link data field in the link data input file, separated by the equals (=) sign. Each set of required data must appear in the link data fields input file as shown above. The order of the required data listed does not matter. However, the link data fields must be titled as shown in the associated link data dictionary file (discussed in the next section) because the dictionary file is used by the utility to determine the location of the data in the link data input file. If any of the columns of data other than the coordinate data (i.e., From Latitude, From Longitude, To Latitude, and To Longitude) are missing, the utility will write an error to the listing file and terminate the current execution. If the coordinate input file directs TRANSVMT to report the coordinate data and any of the



coordinate data are missing, the utility will write an error to the listing file and terminate the current execution. Figure 20 shows a sample link data fields input file.

```
ANODE = ANODE
BNODE = BNODE
County = COUNTY
Area Type = ATYPE
Roadtype = FUNCL
Volume = TOT_FLOW
Capacity = TOTCAP
Speed = SPEED
TAZ = TAZ
Length = LENGTH
From Latitude = FromLat
From Longitude = FromLong
To Latitude = ToLat
To Longitude = ToLong
```

**Figure 20. Sample Link Data Fields Input File.**

### **Link Data Input File (/LNK, /LNKA, /LNKM, /LNKP, /LNKO)**

The /LNK, /LNKA, /LNKM, /LNKP, and /LNKO identifiers and corresponding text strings specify the 24-hour TDM link data input file, the four time-period AM Peak link data input file, the four time-period Midday link data input file, the four time-period PM Peak link data input file, and the four time-period Overnight link data input file, respectively. For a 24-hour TDM, only the /LNK identifier should be used in the jcf and is required. For a four time-period TDM, only the /LNKA, /LNKM, /LNKP, and /LNKO identifiers should be used and are required. All five of these link data input files have the same format, an ASCII file with a corresponding dictionary file. The link data input file must contain the columns of data specified in the link data input file's respective link data fields input file. Figure 21 shows a sample link data dictionary file, and Figure 22 shows a sample link data input file, both with the optional coordinate data included.

```

86
"ANODE",I,1,10,0,10,0,,,"",Blank,
"BNODE",I,11,10,0,10,0,,,"",Blank,
"COUNTY",I,21,10,0,10,0,,,"",Blank,
"ATYPE",I,31,2,0,2,0,,,"",Copy,
"FUNCL",I,33,8,0,8,0,,,"",Blank,
"TOT_FLOW",I,41,8,0,8,0,,,"",Blank,
"TOTCAP",I,49,10,0,10,0,,,"",Blank,
"SPEED",R,59,10,0,10,2,,,"",Copy,
"TAZ",I,69,8,0,8,0,,,"",Copy,
"Length",R,77,10,0,10,6,,,"",Blank,
"FromLat",I,67,10,0,10,0,,,"",Blank,
"FromLong",I,77,10,0,10,0,,,"",Blank,
"ToLat",I,87,10,0,10,0,,,"",Blank,
"ToLong",I,97,10,0,10,0,,,"",Blank,

```

**Figure 21. Sample Link Data Dictionary File.**

8961	25295	8	3	12	1861	4650	37	2845	0.64	29369116	-94949057	29360065	-94946720
10203	14544	2	5	13	44	6105	51	1975	0.90	29364998	-95428119	29352043	-95427996
10606	10893	2	5	13	446	4500	50	1974	2.47	29354190	-95462306	29385941	-95480471
10616	10617	2	5	14	72	1950	49	1975	2.46	29387645	-95442717	29351990	-95440705
10617	10616	2	5	14	33	1950	49	1975	2.46	29351990	-95440705	29387645	-95442717
10622	14542	2	5	10	2710	12897	59	1975	1.89	29380520	-95427021	29353167	-95426898

**Figure 22. Sample Link Data Input File.**

**Intrazonal Trips Input File (/MTX, /MTXA, /MTXM, /MTXP, /MTXO)**

The /MTX, /MTXA, /MTXM, /MTXP, and /MTXO identifiers and corresponding text strings specify the 24-hour TDM intrazonal trips input file, the four time-period AM Peak intrazonal trips input file, the four time-period Midday intrazonal trips input file, the four time-period PM Peak intrazonal trips input file, and the four time-period Overnight intrazonal trips input file, respectively. For a 24-hour TDM, only the /MTX identifier should be used in the jcf and is required. For a four time-period TDM, only the /MTXA, /MTXM, /MTXP, and /MTXO identifiers should be used and are required. All five of these intrazonal trips input files have the same format. These text files begin with the following header:

- ZONE – zone number; and
- TRIPS – intrazonal trips for the respective zone.

The order of the columns does not matter, as long as the header and the data following the header are separated by spaces, tabs, or any combination of spaces and tabs. Each zone listed in the link data input file must be in the intrazonal trips input file. If no intrazonal trips exist for a

zone, a value of 0 should be placed in the trips column. Figure 23 shows a sample intrazonal trips input file.

Zone	Trips
1	141
2	251
3	2218
4	143
5	73
6	422
7	1
8	75
9	5

**Figure 23. Sample Intrazonal Trips Input File.**

### **TTI Speed Model Data Input File (/SPDD)**

The /SPDD identifier (required only for TTI Speed Model) and its corresponding text string specify the path and filename of the TTI Speed Model data input file. This identifier should only be used when the TTI Speed Model is the basis for estimating the hourly link speeds. This input file lists the capacity factor and speed factor for each area type/road type combination in the link data input file. The first line is a header line consisting of:

- RDTYPE – road type code;
- ATYPE – area type code;
- CAPFACT – capacity factor; and
- SPDFACT – speed factor.

Each area type and functional class combination in the link data input file must be listed or the utility will write an error to the listing file and terminate the current execution. The order of the columns does not matter, as long as the data in this input file is separated by spaces, tabs, or any combination of spaces and tabs. Figure 24 shows a sample TTI Speed Model data input file.

RDTYPE	ATYPE	CAPFACT	SPDFACT
01	01	0.093750	1.441300
01	02	0.097884	1.398264
01	03	0.098684	1.278228
01	04	0.112069	1.292247
01	05	0.126582	1.325758
02	01	0.093750	1.441300
02	02	0.097884	1.398264
02	03	0.098684	1.278228
02	04	0.112069	1.292247

**Figure 24. Sample TTI Speed Model Data Input File.**

### Delay Parameters Input File (/DELP)

The /DELP identifier (required only for TTI Speed Model) and its corresponding text string specify the path and filename of the delay parameters input file. This identifier should only be used when the TTI Speed Model is the basis for estimating the hourly link speeds. This input file lists the A coefficient, B coefficient, and M factor for each county and road type combination in the link data input file. The first line is a header line consisting of:

- COUNTY – county code;
- RDTYPE – road type code;
- ACOEFF – A coefficient;
- BCOEFF – B coefficient; and
- MFACTOR – M factor.

Each county and road type combination listed in the link data input file must be listed or the utility will write an error to the listing file and terminate the current execution. The order of the columns does not matter, as long as the data in this input file is separated by spaces, tabs, or any combination of spaces and tabs. Figure 25 shows a sample delay parameters input file.

COUNTY	RDTYPE	ACOEFF	BCOEFF	MFACTOR
01	01	0.015	3.5	5
02	01	0.015	3.5	5
03	01	0.015	3.5	5
01	02	0.015	3.5	5
02	02	0.015	3.5	5
03	02	0.015	3.5	5
01	03	0.015	3.5	5
02	03	0.015	3.5	5
03	03	0.015	3.5	5

**Figure 25. Sample Delay Parameters Input File.**

## Houston Speed Model Data Input File (/SPDH)

The /SPDH identifier (required only for Houston Speed Model) and its corresponding text string specify the path and filename of the Houston Speed Model data input file. This identifier should only be used when the Houston Speed Model is the basis for estimating the hourly link speeds. This input file lists the AM Peak capacity factor, Midday capacity factor, PM Peak capacity factor, Overnight capacity factor, the free-flow speed factor, and the LOS E speed factor for each area type and road type combination in the link-data input file. This input file also lists the SRF area type and SRF functional class codes for the respective area types and functional classes. The first line is a header line consisting of:

- ATYPE – area type code;
- RDTYPE – road type code;
- SRFATYPE – SRF area type code;
- SRF FUNCL – SRF functional class code;
- CAPFACTAMPK – AM Peak capacity factor;
- CAPFACTMDAY – Midday capacity factor;
- CAPFACTPMPK – PM Peak capacity factor;
- CAPFACTOVN – Overnight capacity factor;
- SPD0FACT – free-flow speed factor; and
- SPD1FACT – LOS E speed factor.

Each area type and road type combination in the link data input file must be listed or the utility will write an error to the listing file and terminate the current execution. The order of the columns does not matter, as long as the data in this input file is separated by spaces, tabs, or any combination of spaces and tabs. Figure 26 shows a sample Houston Speed Model data input file.

ATYPE	RDTYPE	SRFATYPE	SRFFUNCL	CAPFACTAMPK	CAPFACTMDAY	CAPFACTPMPK	CAPFACTOVN	SPD0FACT	SPD1FACT
01	01	01	01	0.3333333	0.1666667	0.2500000	0.0909091	1.147400	0.700000
01	02	01	01	0.3333333	0.1666667	0.2500000	0.0909091	1.147400	0.700000
01	10	01	01	0.3333333	0.1666667	0.2500000	0.0909091	1.147400	0.700000
01	11	01	01	0.3333333	0.1666667	0.2500000	0.0909091	1.147400	0.700000
02	01	02	01	0.3333333	0.1666667	0.2500000	0.0909091	1.177508	0.768691
02	02	02	01	0.3333333	0.1666667	0.2500000	0.0909091	1.177508	0.768691
02	10	02	01	0.3333333	0.1666667	0.2500000	0.0909091	1.177508	0.768691
02	11	02	01	0.3333333	0.1666667	0.2500000	0.0909091	1.177508	0.768691
03	01	03	01	0.3333333	0.1666667	0.2500000	0.0909091	1.157272	0.737693

**Figure 26. Sample Houston Speed Model Data Input File.**

## Speed Reduction Factor Input File (/SRFD)

The /SRFD identifier (required only for Houston Speed Model) and its corresponding text string specify the path and filename of the SRF input file. This identifier should only be used when the

Houston Speed Model is the basis for estimating the hourly link speeds. This text file lists the SRFs for each v/c ratio from 0 to 1.0 in 0.05 increments for each SRF area type and SRF functional class combination. The first line is a header line consisting of:

- SRFATYPE – SRF area type;
- SRF FUNCL – SRF functional class;
- VC000 – SRF for v/c ratio 0.00;
- VC005 – SRF for v/c ratio 0.05;
- VC010 – SRF for v/c ratio 0.10;
- VC015 – SRF for v/c ratio 0.15;
- VC020 – SRF for v/c ratio 0.20;
- VC025 – SRF for v/c ratio 0.25;
- VC030 – SRF for v/c ratio 0.30;
- VC035 – SRF for v/c ratio 0.35;
- VC040 – SRF for v/c ratio 0.40;
- VC045 – SRF for v/c ratio 0.45;
- VC050 – SRF for v/c ratio 0.50;
- VC055 – SRF for v/c ratio 0.55;
- VC060 – SRF for v/c ratio 0.60;
- VC065 – SRF for v/c ratio 0.65;
- VC070 – SRF for v/c ratio 0.70;
- VC075 – SRF for v/c ratio 0.75;
- VC080 – SRF for v/c ratio 0.80;
- VC085 – SRF for v/c ratio 0.85;
- VC090 – SRF for v/c ratio 0.90;
- VC095 – SRF for v/c ratio 0.95; and
- VC100 – SRF for v/c ratio 1.00.

Each SRF area type and SRF functional class combination must be listed or the utility will write an error to the listing file and terminate the current execution. The order of the columns does not matter, as long as the data in this input file is separated by spaces, tabs, or any combination of spaces and tabs.

## Hourly Link VMT and Operational Speeds Output Files (\*.Txx)

Although they are not listed directly in the jcf, the TRANSVMT utility also produces 24 hourly link VMT and operational speeds output files. The path and filenames for these 24 files are created using the path and filename of the listing output file, minus the extension. For each hour, the extension of the listing file is replaced by the text *.Txx*, where the *xx* ranges from 1 to 24, depending upon the hour. For example, if the listing output file is defined as C:\transvmt\output\2007\bpa2007swk.lst, then the hourly link VMT and operational speeds output files would be C:\transvmt\output\2007\bpa2007swk.T01, C:\transvmt\output\2007\bpa2007swk.T02...., C:\transvmt\output\2007\bpa2007swk.T24. These output files are in the proper format for use with any of the TTI emissions inventory development utilities. The data within each of these output files are (columns are in the following order):

- A-Node (or begin node) of the link;
- B-Node (or end node) of the link;
- County code where the link resides;
- Functional class (or facility type if desired) code for the respective link;
- Length of the respective link;
- Operational speed for the respective link;
- VMT for the respective link;
- Associated zone for the respective link;
- Area type code for the respective link; and
- V/C ratio for the respective link.

## Node Coordinate Output Files (\*\_NodeCoordinates.txt)

If the coordinate input file directs TRANSVMT to report the node coordinate data, a node coordinate output file is written for each of the TDM link data input files. The path and filename of the node coordinate output file is generated using the path and filename of the listing output file, minus the extension. If only one TDM link data input file is found in the jcf, the extension of the listing output file is replaced with the text *\_NodeCoordinates.txt*. If four TDM link data input files are used, the extension is replaced with the text *\_AMPK\_NodeCoordinates.txt* for the AM Peak TDM link data input file, the text *\_MDAY\_NodeCoordinates.txt* for the Midday TDM link data input file, the text *\_PMPK\_NodeCoordinates.txt* for the PM Peak TDM link data input file, and the text *\_OVN\_NodeCoordinates.txt* for the Overnight TDM link data input file; thus producing four node coordinate output files. These output files list each node in the respective TDM link data input file, as well as the longitude and latitude for each node. Figure 27 shows a sample node coordinates output file.

Node	Longitude	Latitude
1	-95373329	29758730
2	-95372774	29757730
3	-95373725	29756462
4	-95372191	29756666
5	-95373558	29757778

**Figure 27. Sample Node Coordinate Output File.**

**Tab-Delimited Summary Output File (\*.tab)**

The TRANSVMT utility also produces a tab-delimited summary output file. This output file is not listed directly in the jcf. The path and filename for this output file is created using the path and filename of the listing output file, minus the extension and adding the text Summary and the .tab extension. For example, if the listing output file is defined as C:\TRANSVMT\output\2007\bpa2007swk.lst, then the tab-delimited summary output file would be C:\TRANSVMT\output\2007\bpa2007swkSummary.tab. This tab-delimited output file includes a summary of VMT, VHT, and speed (VMT/VHT) by county (including an all counties total), hour (including a total for all 24-hours), functional class, and area type.

**Tab-Delimited TDM Summary Output File (/TMSO)**

The /TMSO identifier (optional) and its corresponding text string specify the path and filename of the TDM summary output file. This tab-delimited output file lists a VMT summary by county, road type, and area type for the 24-hour TDM data or by time-period for a four-time period TDM.

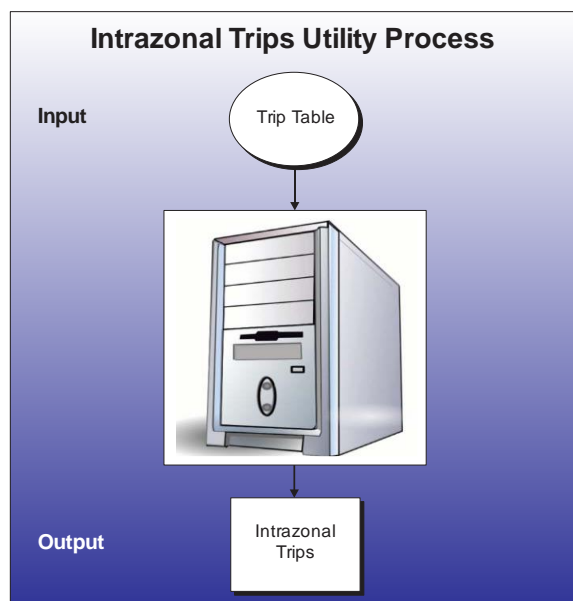


## INTRATRIPSOUTPUT: INTRAZONAL TRIPS UTILITY

The IntraTripsOutput utility extracts the intrazonal trips from a trip table and formats them for use with the TRANSVMT and VMTtot utilities. Only the zone number and the number of trips are output. This utility will work with any trip table that is a text file and is in the proper format.

### OPERATION

The IntraTripsOutput utility reads the jcf until it reaches the step to run. Each line of the desired step is written to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The jcf input file is then re-opened. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.



If any of the required input files are missing from the jcf, or IntraTripsOutput cannot open any of these input files due to an incorrect path and/or filename or any other access error, then IntraTripsOutput results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The utility reads each line of the trip table input file (each line consists of the “from” zone number, the “to” zone number, and the number of trips) and if the “from” zone number is equal to the “to” zone number, then the zone number and the number of trips is written to the intrazonal trips output file. The total number of trips and the total number of intrazonal trips are also calculated as each line is read.

### JCF INPUT FILE

The jcf input file directs the IntraTripsOutput utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (IntraTripsOutput in this case), or the path and filename of the input or output file. Table 11 shows the file type identifiers. Any other characters at the beginning of a line in the jcf input file are not recognized by the utility and do not affect the execution of the utility.

**Table 11. IntraTripsOutput File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Step Identifier and Utility Name
/LST	Listing Output File	Path and Filename
/MTX	Trip Table Input File	Path and Filename
/IMTX	Intrazonal Trips Output File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only used for reference purposes and not accessed by the utility.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (IntraTripsOutput), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 28 shows a sample jcf.

/JOB	IntraTripsOutput
/STEP	Step IntraTripsOutput
/LST	C:\ IntraTripsOutput\Output\IntraTrips_Listing.lst
/MTX	C:\ IntraTripsOutput\Input\TriptTable.asc
/IMTX	C:\ IntraTripsOutput\Output\IntraTrips_Trips.txt

**Figure 28. Sample IntraTripsOutput JCF.**

### **Job Identifier (/JOB)**

The /JOB identifier and its corresponding text string are comments used only as a description of the jcf. This identifier is not used by the utility but is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier and its corresponding text string name the utility to run. This text string is not used by the IntraTripsOutput utility, but only by MOVES2014aUtl to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### Listing File Output (/LST)

The /LST identifier and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Begin time of the step;
- Utility name (in this case IntraTripsOutput);
- Path and filename of the jcf input file;
- JCF through the current step;
- The path and filename of the trip table input file;
- The path and filename of the intrazonal trips output file;
- The total number of trips from the trip table input file;
- The total number of intrazonal trips from the trip table input file; and
- End time of the step after it is completed.

### Trip Table Input File (/MTX)

The /MTX identifier (required) and its corresponding text string specify the path and filename of the trip table input file. The trip table input file lists each zone-to-zone combination and the associated number of trips from the trip matrix. Each line consists of the “from” zone, “to” zone, and the number of trips, with each set of data separated by at least one space, tab, or combination spaces and tabs. Figure 29 shows a sample trip table input file.

1	1	6.480487
1	2	12.531475
1	3	4.447803
1	4	4.813423
1	5	9.065815
1	6	2.899225
1	7	2.631721
1	8	2.538330
1	9	5.385235
1	10	2.055096

**Figure 29. Sample Trip Table Input File.**

### Intrazonal Trips Output File (/IMTX)

The /IMTX identifier (required) and its corresponding text string specify the path and filename of the intrazonal trips output file. This text file lists each zone that has intrazonal trips in the trip table and the number of trips associated with that zone. Figure 30 shows a sample intrazonal trips output file.

Zone	Trips
1	6.480487
2	40.172031
3	20.169640
4	45.869946
5	65.609650
6	7.834467
7	9.676954
8	7.265394
9	26.384670

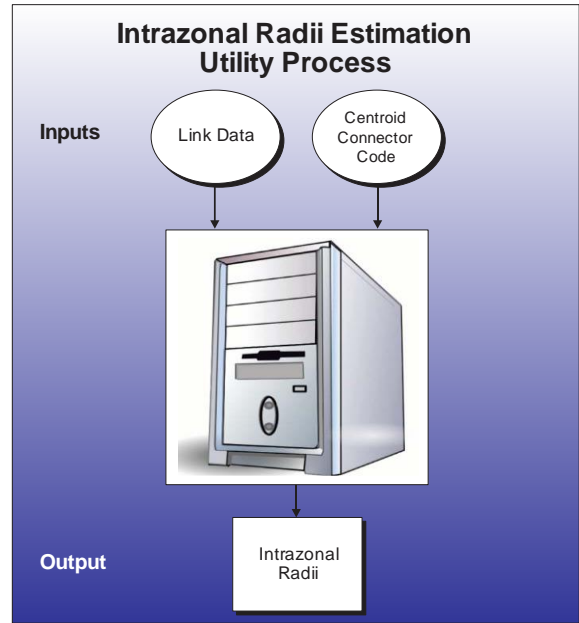
**Figure 30. Sample Intrazonal Trips Output File.**

## RADIICALC: INTRAZONAL RADII ESTIMATION UTILITY

The RADIICALC utility estimates the intrazonal radii for a TDM link data set. The output for this utility is in the proper format for use with the TRANSVMT and VMTtot utilities. Since the inputs required for the RADIICALC utility are also used in the TRANSVMT and VMTtot utilities, no new inputs are required.

### OPERATION

The RADIICALC utility reads the jcf until it reaches the step to run. Each line of the desired step is written to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The jcf input file is then re-opened. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs and outputs.



If any of the required input files are missing from the jcf, or RADIICALC cannot open any of these input files due to an incorrect path and/or filename or any other access error, then RADIICALC results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file processed by the utility is the centroid connector road type code input file. This input file consists of one line, which is the centroid connector road type code.

The utility then processes the link data fields input file. For the link data fields input file, each line is used to determine the link data field in the link data input file that will be used for the required link data (road type code, TAZ, and Time data), with the required data and associated link data field saved in the link data fields array.

The utility then processes the link data input file. For each line in the link data input file, the utility sets the road type code, TAZ, and time using the data in the link data fields array and the location data from the dictionary file. If the road type code is the same as the centroid connector road type code, then the utility saves the time based on the TAZ for that line in the radii data array. The utility maintains a counter for each TAZ in the radii data array.

The utility then begins the process of writing the intrazonal radii output. The utility writes the text RADII, the TAZ number, and the average time (calculated by dividing the time for the TAZ by the counter for the TAZ from the radii data array) to the intrazonal radii output file for each TAZ in the radii data array.

## JCF INPUT FILE

The jcf input file directs the RADIICALC utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (RADIICALC in this case), or the path and filename of the input or output file. Table 12 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 12. RADIICALC File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/CCRC	Centroid Connector Road Type Code Input File	Path and Filename
/LDF	TDM Link Data Fields Input File	Path and Filename
/LNK	TDM Link Data Input File	Path and Filename
/RADO	Intrazonal Radii Output File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (RADIICALC), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 31 shows a sample jcf.

/JOB	RADIICALC Sample JCF
/STEP	Step1 RADIICALC
/LST	C:\RADIICALC\Output\Intrazonal_Radii.lst
/CCFC	C:\RADIICALC\Input\Centroid_Conn_Code.TXT
/LNK	C:\RADIICALC\Input\Linkdata_TRANSVMT.asc
/LNK	C:\RADIICALC\Input\Linkdata_TRANSVMT.asc
/RADO	C:\RADIICALC\Output\Intrazonal_Radii.txt

**Figure 31. Sample RADIICALC jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The RADIICALC utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- JCF through the step being executed;
- Path, filename, and contents of the centroid connector code road type input file;
- Path and filename of the link data input file;
- The maximum zone number from the link data;
- Path and filename of the intrazonal radii output file; and
- Date and time of the end of the utility execution.

### **Centroid Connector Road Type Code Input File (/CCRC)**

The /CCRC identifier (required) and its corresponding text string specify the path and filename of the centroid connector road type code input file. This text file lists the road type code of the centroid connectors as it appears in the link data. This input file is the same centroid connector road type code input file that is used with the TRANSVMT and VMTtot utilities. Figure 32 shows a sample centroid connector road type code input file.



**Figure 32. Sample Centroid Connector Road Type Code Input File.**

## Link Data Fields Input File (/LDF)

The /LDF identifier (required) and corresponding text string specifies the TDM link data fields input file. This input file is a text file listing each set of required data and associated link data field in the link data input file. This input file is the same link data input file used with the TRANSVMT and VMTtot utilities, except that the user must include the “TIME” column of data for this utility to function properly. The user should use the exact link data input file used with TRANSVMT or VMTtot as the input for this utility with the added “TIME” column of data. The required data consists of:

- ANODE - the A-Node (or begin node) of the link;
- BNODE - the B-Node (or end node) of the link;
- County - the county code where the link resides;
- Area Type - the area type code for the respective link;
- Roadtype - the road type code for the respective link;
- Volume - the volume for the respective link;
- Capacity - the capacity for the respective link;
- Speed - the speed to be input into the speed model;
- TAZ - the associated zone for the respective link;
- Length - the length of the respective link;
- TIME – the time for the respective link;
- From Latitude – latitude of ANODE (optional, only if node coordinates are to be output);
- From Longitude – longitude of ANODE (optional, only if node coordinates are to be output);
- To Latitude – latitude of BNODE (optional, only if node coordinates are to be output);  
and
- To Longitude – longitude of BNODE (optional, only if node coordinates are to be output).

Only one set of required data can appear on each line, with the associated link data field in the link data input file, separated by the equals (=) sign. Each set of required data must appear in the link data fields input file as shown previously. The order of the required data listed does not matter. However, the link data fields must be titled as shown in the associated link data dictionary file (discussed in the next section) because the dictionary file is used by the utility to determine the location of the data in the link data input file. If any of the columns of data other than the coordinate data (i.e., From Latitude, From Longitude, To Latitude, and To Longitude) are missing, the utility will write an error to the listing file and terminate the current execution. Figure 33 shows a sample link data fields input file.



```

ANODE = ANODE
BNODE = BNODE
County = COUNTY
Area Type = ATYPE
Roadtype = FUNCL
Volume = TOT_FLOW
Capacity = TOTCAP
Speed = SPEED
TAZ = TAZ
Length = LENGTH
Time = TIME
From Latitude = FromLat
From Longitude = FromLong
To Latitude = ToLat
To Longitude = ToLong

```

**Figure 33. Sample Link Data Fields Input File.**

**Link Data Input File (/LNK)**

The /LNK identifier and corresponding text string specify the link data input file. This input file is an ASCII file with a corresponding dictionary file. This input file is the same link data input file used with the TRANSVMT and VMTtot utilities, except that the user must include the “TIME” column of data for this utility to function properly. The user should use the exact link data input file used with TRANSVMT or VMTtot as the input for this utility with the added “TIME” column of data. The link data input file must contain the columns of data specified in the link data input file’s respective link data fields input file. Figure 34 shows a sample link data dictionary file, and Figure 35 shows a sample link data input file, both with the optional coordinate data included.

```

168
“ANODE”,I,1,10,0,10,0,,,””,Blank,
“BNODE”,I,11,10,0,10,0,,,””,Blank,
“COUNTY”,I,21,10,0,10,0,,,””,Blank,
“ATYPE”,I,31,10,0,10,0,,,””,Blank,
“FUNCL”,I,41,10,0,10,0,,,””,Blank,
“TOT_FLOW”,R,51,20,0,20,6,,,””,Blank,
“TOTCAP”,I,71,10,0,10,0,,,””,Blank,
“SPEED”,R,81,10,0,10,2,,,””,Blank,
“TAZ”,I,91,8,0,8,0,,,””,Blank,
“LENGTH”,R,99,10,0,10,6,,,””,Blank,
“TIME”,R,109,20,0,20,6,,,””,Blank,
“FromLat”,I,129,10,0,10,0,,,””,Blank,
“FromLong”,I,139,10,0,10,0,,,””,Blank,
“ToLat”,I,149,10,0,10,0,,,””,Blank,
“ToLong”,I,159,10,0,10,0,,,””,Blank,

```

**Figure 34. Sample Link Data Dictionary File.**

7492	8306	2	3	12	26210.101529	20700	41	187	0.336537	0.492493	30439794	-97779066	30437070	-97774392
8308	10223	2	3	12	13242.824405	20700	41	187	0.489614	0.716508	30438092	-97777948	30434464	-97770904
8306	7563	2	3	11	15881.552541	19550	35	187	0.211934	0.363315	30437070	-97774392	30435330	-97771464
8306	11702	2	3	12	10328.548989	20700	41	187	0.092148	0.134850	30437070	-97774392	30436418	-97773044
8266	8277	2	3	7	11524.991617	12400	38	168	0.236106	0.372799	30441421	-97773458	30438247	-97771980
188	8297	1	3	0	414.742504	199998	25	188	0.281668	0.676004	30427591	-97772370	30424121	-97769874

**Figure 35. Sample Link Data Input File.**

## VMTTOT: TRAVEL MODEL VMT SUMMARY UTILITY

The VMTtot utility estimates the VMT by county based on a TDM and trip matrix. This utility can be used with 24-hour TDMs as well as four time-period TDMs.

### OPERATION

The VMTtot utility reads the jcf until it reaches the step to run. Each line of the desired step is written to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The jcf input file is then re-opened. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

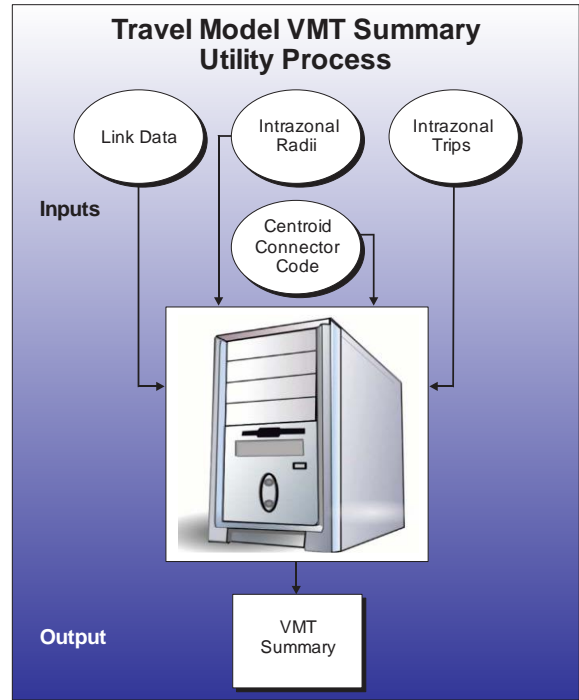
If any of the required input files are missing from the jcf, or VMTtot cannot open any of these input files due to an incorrect path and/or filename or any other access error, then VMTtot results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

As the utility processes the jcf, it determines the type of TDM based on the identifiers used in the jcf, either a 24-hour TDM or a four time-period TDM. The default TDM is set to a four time-period TDM. If the /LNK identifier is in the jcf, the utility determines that the link data input file is for a 24-hour TDM. Otherwise, the program assumes that the link data input file is for a four time-period assignment.

The first input file processed by the utility is the header input file. This input file consists of one line, which is saved as the header.

The utility then processes the centroid connector road type code input file. This input file consists of one line, which is the centroid connector road type code.

If the TDM input file is a 24-hour TDM (as defined in the jcf by the /LNK identifier), the utility then processes the link data fields input file, the link data input file, and the the intrazonal trips input file. For the link data fields input file, each line is used to determine the link data field in the link data input file that will be used for the required link data (county code, area type code,



road type code, A-Node, B-Node, TAZ, volume, capacity, length, speed, from latitude [optional], from longitude [optional], to latitude [optional], and to longitude [optional] data), with the required data and associated link data field saved in the link data fields array.

For the link data input file, the utility uses the data in the link data fields array and the dictionary file that must accompany the link data input file, to determine the location of the required link data in the link data input file. For each line in the link data input file, the utility saves the required data listed above in the link data array. As each line is read, the VMT for each link is also calculated and added to the link data VMT array for summary purposes later in the process.

For the intrazonal trips input file, each line is used to determine the zone and the zone's respective intrazonal trips. The intrazonal trips are saved in the intrazonal data array according to the zone number related to the intrazonal trips.

For each zone, the utility then calculates the intrazonal VMT for that zone. The utility calculates the average speed of the centroid connector within the zone and saves the data in the intrazonal data array, along with the county code where the zone resides. Then, using the intrazonal trips (from the intrazonal data array), the radii (from the radii array) and the average speed of the centroid connectors within the given zone, the utility calculates the intrazonal VMT and saves that data in the intrazonal data array. The VMT is also saved in the link data VMT array for summary purposes.

For the four time-period TDM, the same process detailed previously is followed for the AM Peak, Midday, PM Peak and Overnight set of input files (link data fields input file, link data input file and intrazonal trips input file). The main difference in the processes is that the data are saved in arrays respective to the time period for the four time-period TDM (i.e., the AM Peak link data are saved in the AM Peak link data array, the Midday link data are saved in the Midday link data array, etc.). As each time period is processed, the summary of the time period link data VMT and intrazonal VMT by county is calculated and written to the listing file.

## **JCF INPUT FILE**

The jcf input file directs the VMTtot utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (VMTtot in this case), or the path and filename of the input or output file. Table 13 shows the file type identifiers. Any other characters at the beginning of a line in the jcf input file are not recognized by the utility and do not affect the execution of the utility.

**Table 13. VMTtot File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/HDR	Header Input File	Path and Filename
/CCRC	Centroid Connector Road Type Code Input File	Path and Filename
/RADI	Intrazonal Radii Input File	Path and Filename
/LDF	24-Hour TDM Link Data Fields Input File <sup>2</sup>	Path and Filename
/LDFA, /LDFM, /LDFP, /LDFO	AM Peak, Midday, PM Peak, and Overnight TDM Link Data Fields Input Files, <sup>3</sup> respectively	Path and Filename
/LNK	24-Hour TDM Link Data Input File <sup>2</sup>	Path and Filename
/LNKA, /LNKM, /LNKP, /LNKO	AM Peak, Midday, PM Peak, and Overnight TDM Link Data Input Files, <sup>3</sup> respectively	Path and Filename
/MTX	24-Hour TDM Intrazonal Trips Input File <sup>2</sup>	Path and Filename
/MTXA, /MTXM, /MTXP, /MTXO	AM Peak, Midday, PM Peak, and Overnight TDM Intrazonal Trips Input Files, <sup>3</sup> respectively	Path and Filename
/TMSO	Travel Model Summary Output File <sup>4</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only used for reference purposes and not accessed by the utility.

<sup>2</sup> Only used for 24-hour TDMs.

<sup>3</sup> Only used for four time-period TDMs.

<sup>4</sup> Optional.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (VMTtot), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 36 shows a sample jcf for a 24-hour TDM, followed by a sample jcf for a four time-period TDM in Figure 37.

```

/JOB      VMTtot
/STEP     Step1 VMTtot
/LST      C:\VMTtot\output\2007\bpa2007swk.lst
/TMSO    C:\VMTtot\output\2007\hgb2007_TDMsummary.tab
/HDR      C:\VMTtot\input\bpa07mtp_2007_Header.txt
/CRFC     C:\VMTtot\input\BPA_ccCode.txt
/LNK      C:\VMTtot\input\TDMs\2007\Network\Out\bpa07Network_VMTtot.asc
/RADI     C:\VMTtot\input\TDMs\2007\Radii\bpa07rad.asc
/MTX      C:\VMTtot\input\TDMs\2007\TripMatrix\BPA2007IntraTrips.txt

```

**Figure 36. Sample 24-hour TDM VMTtot jcf.**

```

/JOB      VMTtot
/STEP     Step1 VMTtot
/LST      C:\VMTtot\output\2007\hgb2007_VMTtot.lst
/TMSO    C:\VMTtot\output\2007\hgb2007_TDMsummary.tab
/HDR      C:\VMTtot\input\hgb_2007_VMTtot_Header.txt
/CCRC     C:\VMTtot\input\ccCode.txt
/LNKA     C:\VMTtot\input\TDMs\2007\Network\Out\hgb07Network_AMPK_TRANSVMT.asc
/LNKM     C:\VMTtot\input\TDMs\2007\Network\Out\hgb07Network_MDAY_TRANSVMT.asc
/LNKP     C:\VMTtot\input\TDMs\2007\Network\Out\hgb07Network_PMPK_TRANSVMT.asc
/LNKO     C:\VMTtot\input\TDMs\2007\Network\Out\hgb07Network_OVN_TRANSVMT.asc
/RADI     C:\VMTtot\input\TDMs\2007\Radii\hgb07rad.asc
/MTXA     C:\VMTtot\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_AMPK.txt
/MTXM     C:\VMTtot\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_MDAY.txt
/MTXP     C:\VMTtot\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_PMPK.txt
/MTXO     C:\VMTtot\input\TDMs\2007\TripMatrix\hgb2007IntraTrips_OMP.txt
/LDFA     C:\VMTtot\input\TDMs\2007\Network\Out\LinkDataFields_AMPK.txt
/LDFM     C:\VMTtot\input\TDMs\2007\Network\Out\LinkDataFields_MDAY.txt
/LDFP     C:\VMTtot\input\TDMs\2007\Network\Out\LinkDataFields_PMPK.txt
/LDFO     C:\VMTtot\input\TDMs\2007\Network\Out\LinkDataFields_OVN.txt

```

**Figure 37. Sample Four Time-Period TDM VMTtot jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. This identifier is not used by the utility but is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. This text string is not used by the VMTtot utility, but only by MOVES2014aUtl to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part being separated by a space, a tab, or any combination of tabs and spaces.

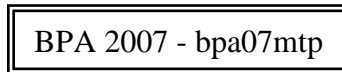
### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file basically summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- JCF through the step being executed;
- Path and filename of all input files;
- TDM and intrazonal VMT summary; and
- Date and time of the end of the utility execution.

### **Header Input File (/HDR)**

The /HDR identifier (required) and its corresponding text string specify the path and filename of the header input file. This text file lists the header that is written to the listing file to assist in naming the specific VMTtot execution. Figure 38 shows a sample header input file.



BPA 2007 - bpa07mtp

**Figure 38. Sample Header Input File.**

### **Centroid Connector Road Type Code Input File (/CCRC)**

The /CCRC identifier (required) and its corresponding text string specify the path and filename of the centroid connector road type code input file. This text file lists the road type code of the centroid connectors as it appears in the link data. Figure 39 shows a sample centroid connector road type code input file.



39

**Figure 39. Sample Centroid Connector Road Type Code Input File.**

### **Intrazonal Radii Input File (/RADI)**

The /RADI identifier and corresponding text string specify the path and file name of the intrazonal radii input file. This text file lists the text RADI, the zone number, and the intrazonal radii. This file is the exact output of the RADIICALC utility used to calculate the radii. Figure 40 shows a sample intrazonal radii input file.

RADII	1	4.6400
RADII	2	2.3600
RADII	3	1.7400
RADII	4	3.5000
RADII	5	3.5300
RADII	6	4.8100
RADII	7	4.0600
RADII	8	2.2900
RADII	9	0.8500
RADII	10	0.9600

**Figure 40. Sample Intrazonal Radii Input File.**

**Link Data Fields Input File (/LDF, /LDFA, /LDFM, /LDFP, /LDFO)**

The /LDF, /LDFA, /LDFM, /LDFP, and /LDFO identifiers and corresponding text strings specify the 24-hour TDM link data fields input file, the four time-period AM Peak link data fields input file, the four time-period Midday link data fields input file, the four time-period PM Peak link data fields input file, and the four time-period Overnight link data fields input file, respectively. For a 24-hour TDM, only the /LNK identifier (required) should be used in the jcf. For a four time-period TDM, only the /LNKA, /LNKM, /LNKP, and /LNKO identifiers (required) should be used. All five of these input files have the same format, a text file listing each set of required data and associated link data field in the link data input file. The required data consists of:

- ANODE - the A-Node (or begin node) of the link;
- BNODE - the B-Node (or end node) of the link;
- County - the county code where the link resides;
- Area Type - the area type code for the respective link;
- Roadtype - the road type code for the respective link;
- Volume - the volume for the respective link;
- Capacity - the capacity for the respective link;
- Speed - the speed to be input into the speed model;
- TAZ - the associated zone for the respective link; and
- Length - the length of the respective link;

Only one set of required data can appear on each line, with the associated link data field in the link data input file, separated by the equals (=) sign. Each set of required data must appear in the link data fields input file as shown above. The order of the required data listed does not matter. However, the link data fields must be titled as shown in the associated link data dictionary file (discussed in the next section) because the dictionary file is used by the utility to determine the location of the data in the link data input file. If any of the columns of data are



missing, the utility will write an error to the listing file and terminate the current execution. Figure 41 show a sample link data fields input file.

```

ANODE = ANODE
BNODE = BNODE
County = COUNTY
Area Type = ATYPE
Roadtype = FUNCL
Volume = TOT_FLOW
Capacity = TOTCAP
Speed = SPEED
TAZ = TAZ
Length = LENGTH

```

**Figure 41. Sample Link Data Fields Input File.**

### Link Data Input File (/LNK, /LNKA, /LNKM, /LNKP, /LNKO)

The /LNK, /LNKA, /LNKM, /LNKP, and /LNKO identifiers and corresponding text strings specify the 24-hour TDM link data input file, the four time-period AM Peak link data input file, the four time-period Midday link data input file, the four time-period PM Peak link data input file, and the four time-period Overnight link data input file, respectively. For a 24-hour TDM, only the /LNK identifier should be used in the jcf and is required. For a four time-period TDM, only the /LNKA, /LNKM, /LNKP, and /LNKO identifiers should be used and are required. All five of these link data input files have the same format, an ASCII file with a corresponding dictionary file. The link data input file must contain the columns of data specified in the link data input file's respective link data fields input file. Figure 42 shows a sample link data dictionary file and Figure 43 shows a sample link data input file.

```

168
"ANODE",I,1,10,0,10,0,,,"",Blank,
"BNODE",I,11,10,0,10,0,,,"",Blank,
"COUNTY",I,21,10,0,10,0,,,"",Blank,
"ATYPE",I,31,10,0,10,0,,,"",Blank,
"FUNCL",I,41,10,0,10,0,,,"",Blank,
"TOT_FLOW",R,51,20,0,20,6,,,"",Blank,
"TOTCAP",I,71,10,0,10,0,,,"",Blank,
"SPEED",R,81,10,0,10,2,,,"",Blank,
"TAZ",I,91,8,0,8,0,,,"",Blank,
"LENGTH",R,99,10,0,10,6,,,"",Blank,
"TIME",R,109,20,0,20,6,,,"",Blank,
"FromLat",I,129,10,0,10,0,,,"",Blank,
"FromLong",I,139,10,0,10,0,,,"",Blank,
"ToLat",I,149,10,0,10,0,,,"",Blank,
"ToLong",I,159,10,0,10,0,,,"",Blank,

```

**Figure 42. Sample Link Data Dictionary File.**

3884	3622	1	5	1	26119	31600	52.799999	205	0.680150
3630	3631	1	5	1	26419	31600	52.799999	208	0.609656
4173	909	2	5	1	22192	31600	52.799999	598	0.404420
4034	1177	2	5	1	24090	31600	52.799999	517	0.168046
3937	3933	1	5	5	4606	31600	52.799999	336	0.252206
3931	3935	1	5	5	7476	31600	52.799999	375	0.473176
3875	2034	1	5	5	22007	47400	52.799999	173	0.131383
2054	2055	1	5	5	11653	15800	52.799999	173	0.225235
909	3682	4	5	1	22192	31600	52.799999	598	0.690815
3682	3683	2	5	1	22153	31600	52.799999	598	0.249837

**Figure 43. Sample Link Data Input File.**

### **Intrazonal Trips Input File (/MTX, /MTXA, /MTXM, /MTXP, /MTXO)**

The /MTX, /MTXA, /MTXM, /MTXP, and /MTXO identifiers and corresponding text strings specify the 24-hour TDM intrazonal trips input file, the four time-period AM Peak intrazonal trips input file, the four time-period Midday intrazonal trips input file, the four time-period PM Peak intrazonal trips input file, and the four time-period Overnight intrazonal trips input file, respectively. For a 24-hour TDM only the /MTX identifier should be used in the jcf. For a four time-period TDM, only the /MTXA, /MTXM, /MTXP, and /MTXO identifiers should be used. Either the /MTX identifier or the four time-period identifiers must be included in the jcf or VMTtot results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2010butl code to check the jcf for more steps to execute. All five of these intrazonal trips input files have the same format. These text files begin with the following header:

- ZONE – zone number; and
- TRIPS – intrazonal trips for the respective zone.

The order of the columns does not matter, as long as the header and the data following the header are separated by spaces, tabs, or any combination of spaces and tabs. Each zone listed in the link data input file must be in the intrazonal trips input file. If no intrazonal trips exist for a zone, a value of 0 should be placed in the trips column. Figure 44 shows a sample intrazonal trips input file.

Zone	Trips
1	141
2	251
3	2218
4	143
5	73
6	422
7	1
8	75
9	5

**Figure 44. Sample Intrazonal Trips Input File.**

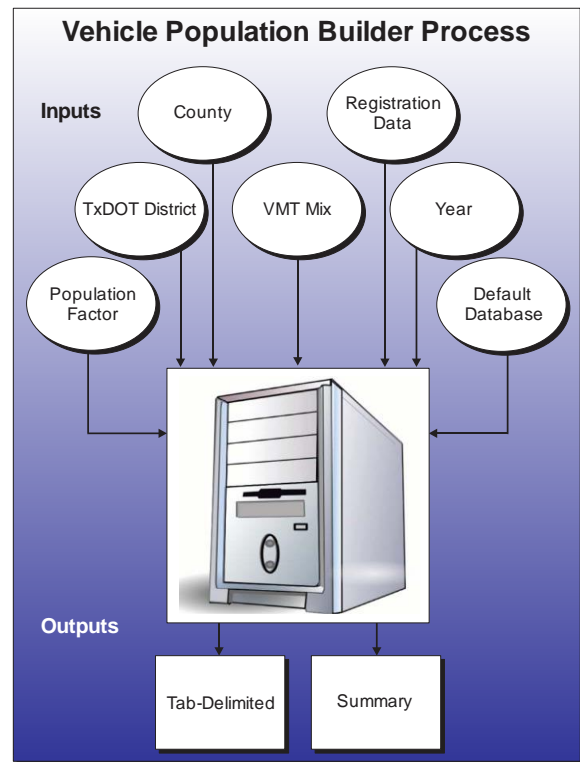


## VEHPOPULATIONBUILD: VEHICLE POPULATION BUILDER

The VehPopulationBuild utility develops sourcetypeyear output in a format consistent with MOVES (both tab-delimited format and database table format options) and the tab-delimited population by MOVES SUT/fuel type combination that may be used as input to EmsCalc or OffNetActCalc utilities. Vehicle registration data supplied by the Texas Department of Motor Vehicles (TxDMV) and an all roadway types VMT mix are the basis for these outputs.

### OPERATION

The VehPopulationBuild utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.



If any of the required input files are missing from the jcf, or TRANSVMT cannot open any of these input files due to an incorrect path and/or filename or any other access error, then TRANSVMT results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the county name input file. The utility reads each line until it locates the text *GRAND* or *COUNTY* (capitalization does not matter) as the first word in the line and sets the county name as this line.

The next input file the utility processes is the TxDOT district name input file. The utility reads each line until it locates the text *DISTRICT* as the first word and sets the district name as the text following the “=” sign.

The next input file the utility processes is the all roadway types VMT mix input file. The utility then creates the VMT mix array, which is by SUT/fuel type. The utility reads each line

until locating the text *Obs* in the first segment, which signifies the header line for the first block of data. Beginning with the fourth column of data, the utility uses the last three characters to set the SUT code (set as the first two characters) and the fuel type (set as the last character). The utility sets the fuel type code based on the fuel type: G is fuel type 1, D is fuel type 2, C is fuel type 3, and E is fuel type 9. For example, if the segment of the header line for the first block of data is the text *P\_PV21D*, then the SUT code is 21, the fuel type is D and the fuel type code is 2.

The utility saves the SUT code and fuel type code for each column of data in the code segment of the VMT mix array. The utility then begins the process of reading the first block of data. The utility then reads each line until locating the appropriate district (as defined by the district name from the TxDOT district name input file) and saves the observation number and the VMT mix values to the VMT mix array based on the SUT code and fuel type code. This process is repeated until the utility reaches the end of the file or locates the text *Obs*. If the text *Obs* is found, then the utility recognizes this as a new block of data and uses a process similar to the first block of data to read the VMT mix data (sets the SUT code and fuel type code and uses the observation number instead of the district name). The utility then reads each line until it reaches the correct observation number and saves the VMT mix data in the VMT mix array based on the SUT code and fuel type code.

This utility repeats this process until it reaches the end of the. The utility then checks the sum of the VMT mix data across all of the SUT/fuel type combinations. If this sum is not equal to the value of 1, then the utility adjusts the largest VMT mix data value so that the total VMT mix is equal to the value of 1. In the event that the VMT mix does not sum to the value of 1, the utility writes a warning to the listing output file.

The next input file the utility processes is the age registrations input file. The utility reads each line until it locates the appropriate county. After the appropriate county has been located, the utility reads each line until locating the text *MODEL* and the utility skips the next line, which is part of the data header. The utility then reads the next 32 lines and saves the registration data (Passenger, Motorcycles, Trucks <=6000, Trucks >6000 <=8500, Total Trucks <=8500 age registrations vehicle categories) in the age registration data array (by vehicle type and year with the *OLDER* year model saved in year 31 and the *TOTAL* year model saved in year 32).

The next input file the utility processes is the diesel heavy-duty registrations input file. The utility reads each line until it locates the appropriate county. After the appropriate county has been located, the utility reads each line until locating the text *MODEL* and the utility skips the next line, which is part of the data header. The utility then reads the next 32 lines and saves the registration data (Diesel > 8500, Diesel > 10000, Diesel > 14000, Diesel > 16000, Diesel > 19500, Diesel > 26000, Diesel > 33000, Diesel > 60000 diesel heavy-duty registrations vehicle categories) in the diesel portion of the diesel/gas registration data array (by diesel/gas, vehicle type and year with the *OLDER* year model saved in year 31 and the *TOTAL* year model saved in year 32).

The next input file the utility processes is the gas heavy-duty registrations input file. The utility reads each line until it locates the appropriate county. After the appropriate county has been located, the utility reads each line until locating the text *MODEL* and the utility skips the

next line, which is part of the data header. The utility then reads the next 32 lines and saves the registration data (Gas > 8500, Gas > 10000, Gas > 14000, Gas > 16000, Gas > 19500, Gas > 26000, Gas > 33000, Gas > 60000 gas heavy-duty registrations vehicle categories) in the gas portion of the diesel/gas registration data array (by diesel/gas, vehicle type and year with the *OLDER* year model saved in year 31 and the *TOTAL* year model saved in year 32).

The next input file the utility processes is the population factor input file. If this input file is missing from the jcf, then the utility does not process this input file and will not apply the population factor during the calculation process. Otherwise, the utility reads each line until locating the text *POPFACTOR* (capitalization does not matter) as the first word and sets the population factor as the number following the “=” sign.

If the population factor input file was found in the jcf, the utility then processes the year ID input file. The utility reads each line until locating the text *YEAR* (capitalization does not matter) as the first word and sets the year output as the number following the “=” sign.

If the database format of the sourcetypeyear output is found in the jcf, the utility then processes the MySQL connection inputs file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

The utility then begins the calculation process. The first step of the calculation process is to create the registration data category array (by registration data category and year, including total). The utility combines the age registration data array and the diesel/gas registration data array to form the registration category data array using the combinations in Table 14.

**Table 14. Registration Categories.**

Registration Category	Vehicle Category	Data Location
1	Passenger	Age registration data array
2	Motorcycles	
3	Total Trucks <=8500	
4	Diesel > 8500, Diesel > 10000, Diesel > 14000, Diesel > 16000	Diesel/gas registration data array
5	Diesel > 19500, Diesel > 26000, Diesel > 33000, Diesel > 60000	
6	Gas > 8500, Gas > 10000, Gas > 14000, Gas > 16000	
7	Gas > 19500, Gas > 26000, Gas > 33000, Gas > 60000	

The next step in the calculation process is to create the SUT population array (by SUT/fuel type). The utility assigns each SUT/fuel type combination (excluding long-haul trucks) the total registrations from one or more of the registration categories in the registration category data array. Table 15 shows the SUTs and their associated registration category in the registration category data array.

**Table 15. SUT/Registration Category Correlation.**

SUT	Registration Category
11	2
21	1
31, 32	3
41, 42, 43, 51, 52, 54	4 + 6
61	5 + 7

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

The next step in the calculation process is to create the SUT population factors array (by SUT and fuel type). The utility calculates SUT population factors for all SUTs except motorcycles (SUT 11) and the long-haul trucks (SUTs 53 and 62) using the VMT mix (from the VMT mix array) and saves these SUT population factors in the SUT population factors array.

For SUT 21, the utility divides the fuel type VMT mix by the total VMT mix for SUT 21. For SUT 31, the utility divides the fuel type VMT mix by the total VMT mix for SUTs 31 and 32. The same process applies to SUT 32. For SUT 41, the utility divides the fuel type VMT mix by the total VMT mix for SUTs 41, 42, 43, 51, 52, and 54. The same process applies to SUTs 42, 43, 51, 52, and 54. For SUT 61, the utility divides the fuel type VMT mix by the total VMT mix for SUT 61.

For SUT 11, the SUT population factor for fuel type 1 (gasoline) is set to 1 with all other factors set to 0. For SUT 53, the utility calculates the SUT population factors by fuel type by dividing the fuel type VMT mix for SUT 53 by the fuel type VMT mix for SUT 52. For SUT 62, the utility calculates the SUT population factors by fuel type by dividing the fuel type VMT mix for SUT 62 by the fuel type VMT mix for SUT 61; thus creating a ratio of long-haul trucks and short-haul trucks.



The final step in the calculation process is to calculate the SUT population by SUT and fuel type. For all SUTs, except the long-haul trucks (SUT 53 and 62), the utility applies the SUT population factors and the population factor (if input in the jcf) to the SUT population array to calculate the SUT population. After calculating the SUT population for these SUTs, the utility calculates the SUT population for the long-haul trucks.

For SUT 53, the utility applies the SUT population factors for SUT 53 to the SUT population array for SUT 52. For SUT 62, the utility applies the SUT population factors for SUT 62 to the SUT population array for SUT 61.

The utility then writes the data summaries. The utility writes a summary of the total registrations from the registration data input files to the listing output file. If the text for the tab-delimited registration data summary output file is in the jcf, the utility attempts to create the output file. If VehPopulationBuild cannot create this output file due to an incorrect path and/or filename or any other access error, then VehPopulationBuild results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute.

Otherwise, the utility opens the output file and writes the header line, which consists of the text *YEAR MODEL*, followed by each vehicle category in Table 14 (Registration Categories), with each set of text separated by a tab. For each year model (including OLDER and TOTAL), the utility writes the registration data from the age registration data array and the diesel/gas registration data array to the output file. The utility writes a summary of the VMT mix (from the VMT mix array) and the SUT population factors (from the SUT population factors array) to the listing output file as well.

The utility then writes the population output files and summary. If the text for the tab-delimited population by SUT/fuel type output file is in the jcf, the utility attempts to create the output file. If VehPopulationBuild cannot create this output file due to an incorrect path and/or filename or any other access error, then VehPopulationBuild results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute.

Otherwise, the utility opens the output file and writes the header line, which consists of the text *yearID*, *MOVES\_STcode*, *MOVES\_FTcode*, *Population*, *MOVES\_STdesc*, and *MOVES\_FTdesc*, with each set of text separated by a tab. For each SUT/fuel type combination in the SUT population array, the utility writes the year ID (either the first year from the registration data or the year ID input file), the MOVES SUT, the MOVES fuel type code, the population (from the SUT population array), the MOVES SUT description (from the MOVES base data array), and the MOVES fuel type description (from the MOVES base data array) to this output file. Regardless of whether the text for this output file is in the jcf or not, the utility writes this same data to the listing output file as well.

The utility then writes the *sourcetypeyear* database table and tab-delimited output file. If the text for either the *sourcetypeyear* database table output or the tab-delimited *sourcetypeyear* output is in the jcf, the utility aggregates the SUT population array by fuel type. If the text for the tab-delimited *sourcetypeyear* output is in the jcf, the utility attempts to create the output file. If VehPopulationBuild cannot create this output file due to an incorrect path and/or filename or any other access error, then VehPopulationBuild results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute.

Otherwise, the utility opens the output file and writes the header line, which consists of the text *yearID*, *sourceTypeID*, *sourceTypePopulation*, *salesGrowthFactor*, and *migrationrate*, with each set of text separated by a tab. For each SUT where the fuel type aggregated SUT population (from the SUT population array) is greater than 0, the utility writes the year ID (either the first year from the registration data or the year ID input file), the MOVES SUT code, the SUT population (the fuel type aggregated SUT population from the SUT population array), the sales growth factor (automatically set to 1 by the utility), and the migration rate (automatically set to 1 by the utility) to this output file. If the text for the *sourcetypeyear* database table output is in the jcf, then the utility uses the appropriate MySQL scripts to create a database table similar to the MOVES default *sourcetypeyear* database table. Using MySQL scripts, the utility adds the data to the database table in the same manner as writing the tab-delimited output.

## JCF INPUT FILE

The jcf input file directs the VehPopulationBuild utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (VehPopulationBuild in this case), or the path and filename of the input or output file. Table 16 shows the file type identifiers. Any other characters at the beginning of a line in the jcf input file are not recognized by the utility and do not affect the execution of the utility.

**Table 16. VehPopulationBuild File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/CNTY	County Name Input File	Path and Filename
/AGEI	Age Registrations Input File	Path and Filename
/DSLII	Diesel Heavy-Duty Registrations Input File	Path and Filename
/GASI	Gasoline Heavy-Duty Registrations Input File	Path and Filename
/DISTI	District Name Input File	Path and Filename
/VMXI	VMT Mix Input File	Path and Filename
/MVSDB	MOVES Default Database	Database Name
/POPFI	Population Factor Input File <sup>2</sup>	Path and Filename
/YRID	Year ID Input File <sup>2</sup>	Path and Filename
/REGDO	Registration Data Summary Output File	Path and Filename
/MYSQL	MySQL Connection Inputs File	Path and Filename
/TPFTO	Tab-Delimited Population by SUT/Fuel Type Output File	Path and Filename
/DSTYO	sourcetypeyear Output – Database Table	Database Name
/TSTYO	sourcetypeyear Output – Tab-Delimited File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Only if the utility factors the base population to a future value (i.e., future year analysis).

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (VehPopulationBuild), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 45 shows a sample jcf.

/JOB	VehPopulationBuild Harris
/STEP	Step1 VehPopulationBuild
/LST	C:\VehPopulationBuild\Output\VehPopulationBuild_Harris.lst
/CNTY	C:\VehPopulationBuild\Input\VehPopulationBuild_HarrisCounty.txt
/AGEI	C:\VehPopulationBuild\Input\RegData2009\31 Years\AGEPERKN.txt
/DSLJ	C:\VehPopulationBuild\Input\RegData2009\31 Years\DSLPERKN.txt
/GASI	C:\VehPopulationBuild\Input\RegData2009\31 Years\GASPERKN.txt
/DISTI	C:\VehPopulationBuild\Input\District_Houston.txt
/VMXI	C:\VehPopulationBuild\Input\SU_08RT.Lst
/MVSDB	movesdb20140731
/MYSQL	C:\VehPopulationBuild\Input\MySQL_ConnInputs_MOVES2014.txt
/REGDO	C:\VehPopulationBuild\Output\VehPopulationBuild_Harris_RegData.tab
/TPFTO	C:\VehPopulationBuild\Output\VehPopulationBuild_Harris_StFtPop.tab
/DSTYO	VehPopulationBuild_Harris
/TSTYO	C:\VehPopulationBuild\Output\VehPopulationBuild_Harris_sourcetypeyear.tab

**Figure 45. Sample VehPopulationBuild jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The VehPopulationBuild utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (VehPopulationBuild in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path, and filename of all input and output files (some input files also include the contents of the input file);

- The total registrations for the main registration data categories used in the calculations;
- The VMT mix summary;
- The SUT/fuel type combination population factors used in the calculations;
- The path and filename or database name of all output files;
- A population summary by SUT/fuel type combination;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### County Name Input File (/CNTY)

The /CNTY (required) identifier and its corresponding text string specify the path and filename of the county name input file, which lists the county name or state designation for calculating the population. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. If the population is based on a county, the input line consists of the identifier *COUNTY*, followed by the county name as it appears in the registration data input files (including the county number), and separated by any combination of tabs and/or spaces. If the state totals are the basis for calculating the population, the utility uses the text *GRAND TOTALS* instead of the identifier and county name. Figure 46 shows a sample county name input file.

<p>* NOTE: Comments can be included in jcf by placing the symbol * in the first column.          * NOTE: Must appear as in registration data, including county number.          * NOTE: If using GRAND TOTALS, the identifier COUNTY is not required.          * GRAND TOTALS          COUNTY 101 HARRIS</p>
--

**Figure 46. Sample County Name Input File.**

### Age Registrations Input File (/AGEI)

The /AGEI (required) identifier and its corresponding text string specify the path and filename of the age registrations input file. This input file is a text file and consists of vehicle registrations by model year for each county in Texas and a statewide total. For each block of data (either an individual county or statewide total), 32 lines of registration data (30 labeled model years, an OLDER model year, and a total) must be present, with each model year appearing on one line. The registration data vehicle types included in this input file are PASSENGER, MOTORCYCLES, TRUCKS <=6000, TRUCKS >6000 <=8500, TOTAL TRUCKS <=8500, GAS TRUCKS >8500 (not used), DIESEL TRUCKS >8500 (not used), TOTAL TRUCKS >8500 (not used), and TOTAL ALL TRUCKS (not used). Figure 47 shows a sample of an age registrations input file.

PROGRAM: AGEPERKD			TEXAS DEPARTMENT OF TRANSPORTATION					RUN DATE 11/13/09	
WRITTEN BY: JOEY OESTRICK			AGE COUNTS BY COUNTY USING GROSS WEIGHT					PAGE 1	
COUNTY 1 ANDERSON									
YEAR	PASSENGER	MOTOR-CYCLES	TRUCKS <=6000	TRUCKS >6000 <=8500	TOTAL TRUCKS <=8500	GAS TRUCKS >8500	DIESEL TRUCKS >8500	TOTAL TRUCKS >8500	TOTAL ALL TRUCKS
2009	457	33	97	145	242	5	30	35	277
2008	1,187	71	395	556	951	24	161	185	1,136
2007	1,413	107	485	600	1,085	14	115	129	1,214
.									
.									
1980	26	5	29	2	31	2	0	2	33
OLDER	243	18	248	23	271	12	1	13	284
TOTAL	18,640	952	9,774	4,338	14,112	212	899	1,111	15,223

**Figure 47. Sample Age Registrations Data Input File.**

### Diesel Heavy-Duty Registrations Input File (/DSLJ)

The /DSLJ (required) identifier and its corresponding text string specify the path and filename of the diesel heavy-duty registrations input file. This input file is a text file and consists of diesel heavy-duty vehicle registrations by model year for each county in Texas and a statewide total. For each block of data (either an individual county or statewide total), the user must include 32 lines of registration data (30 labeled model years, an OLDER model year, and a total), with each model year appearing on one line. The registration data vehicle types included in this input file are DIESEL > 8500, DIESEL > 10000, DIESEL > 14000, DIESEL > 16000, DIESEL > 19500, DIESEL > 26000, DIESEL > 33000, DIESEL > 60000, and DIESEL TOTALS (not used). Figure 48 shows a sample of a diesel heavy-duty registrations input file.

PROGRAM: DSLPERKD			TEXAS DEPARTMENT OF TRANSPORTATION							RUN DATE 11/13/09	
WRITTEN BY: JOEY OESTRICK			DIESEL TRUCK COUNTS BY COUNTY USING GROSS WEIGHT							PAGE 1	
COUNTY 1 ANDERSON											
YEAR	DIESEL > 8500	DIESEL > 10000	DIESEL > 14000	DIESEL > 16000	DIESEL > 19500	DIESEL > 26000	DIESEL > 33000	DIESEL > 60000	DIESEL TOTALS		
2009	25	4	0	0	0	0	1	0	30		
2008	134	18	2	3	2	1	1	0	161		
2007	83	21	1	4	1	1	4	0	115		
.											
.											
1980	0	0	0	0	0	0	0	0	0		
OLDER	0	0	0	0	0	1	0	0	1		
TOTAL	661	105	18	25	21	20	48	1	899		

**Figure 48. Sample Diesel Heavy-Duty Registrations Data Input File.**

### Gas Heavy-Duty Registrations Input File (/GASI)

The /GASI (required) identifier and its corresponding text string specify the path and filename of the gas heavy-duty registrations input file. This input file is a text file and consists of gas heavy-

duty vehicle registrations by model year for each county in Texas and a statewide total. For each block of data (either an individual county or statewide total), the user must include 32 lines of registration data (30 labeled model years, an OLDER model year, and a total), with each model year appearing on one line. The registration data vehicle types included in this input file are GAS > 8500, GAS > 10000, GAS > 14000, GAS > 16000, GAS > 19500, GAS > 26000, GAS > 33000, GAS > 60000, and GAS TOTALS (not used). Figure 49 shows a sample of a gas heavy-duty registrations input file.

PROGRAM: GASPERKD		TEXAS DEPARTMENT OF TRANSPORTATION								DATE 11/13/09
WRITTEN BY: JOEY OESTRICK		GAS TRUCK COUNTS BY COUNTY USING GROSS WEIGHT								PAGE 1
COUNTY	1 ANDERSON									
YEAR	GAS	GAS	GAS	GAS	GAS	GAS	GAS	GAS	GAS	GAS
MODEL	> 8500	> 10000	> 14000	> 16000	> 19500	> 26000	> 33000	> 60000	TOTALS	
2009	1	3	1	0	0	0	0	0	5	
2008	18	5	0	0	0	1	0	0	24	
2007	9	4	0	0	0	1	0	0	14	
.										
.										
.										
1980	2	0	0	0	0	0	0	0	2	
OLDER	3	3	1	1	3	0	1	0	12	
TOTAL	128	43	6	6	14	9	6	0	212	

**Figure 49. Sample of an Diesel Heavy-Duty Registrations Data Input File.**

### TxDOT District Name Input File (/DISTI)

The /DISTI (required) identifier and its corresponding text string specify the path and filename of the TxDOT district name input file, which lists the TxDOT district for extracting the VMT mix from the VMT mix input file. The TxDOT district in this input file must appear (spelled the same) as it appears in the VMT mix input file. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the data identifier (District), followed by the equals symbol (=) and the appropriate TxDOT district name, separated by any combination of tabs and/or spaces. Figure 50 shows a sample TxDOT district name input file.

* NOTE: Comments can be included in jcf by placing the symbol * in the first column.
* NOTE: Format is fixed with identifier district followed by the symbol = and the district
* as it appears in the VMT mix input file.
* NOTE: District can be any combination of capital or lowercase letters.
* NOTE: The user can separate all data by any combination of tabs and/or spaces.
* District = Texas
District = Houston

**Figure 50. Sample TxDOT District Name Input File.**

### **VMT Mix Input File (/VMXI)**

The /VMXI (required) identifier and its corresponding text string specify the path and filename of the VMT mix input file. This input file lists the all roadway type 24-hour VMT mix by TxDOT district (or state) for the desired MOVES SUT/fuel type combinations. Figure 51 shows a sample VMT mix input file.



The SAS System													09:38 Friday, January 8, 2010 53	
Obs	District	P_PV21D	P_PV21G	P_PT31D	P_PT31G	P_LCT32D	P_LCT32G	P_RFT51D	P_RFT51G	P_SUSH52D	P_SUSH52G	P_SULH53D		
	P_SULH53G													
1	Abilene	0.00041	0.41037	0.00537	0.16786	0.00398	0.05376	0.00005	0.00001	0.04540	0.01410	0.00121	0.00038	
2	Amarillo	0.00055	0.55034	0.00447	0.15507	0.00383	0.04935	0.00003	0.00002	0.03211	0.01483	0.00127	0.00059	
3	Atlanta	0.00056	0.56050	0.00409	0.13709	0.00329	0.04377	0.00004	0.00001	0.03663	0.01027	0.00071	0.00020	
.														
.														
25	Yoakum	0.00046	0.46201	0.00624	0.20190	0.00500	0.06439	0.00006	0.00002	0.05921	0.01443	0.00266	0.00065	
Obs	P_MH54D	P_MH54G	P_ICB41D	P_TB42D	P_TB42G	P_SB43D	P_SB43G	P_CSH61D	P_CSH61G	P_CLH62D	P_CLH62G	P_MC11G		
1	0.00005	0.00001	0.00036	0.00250	0.00000	0.00425	0.00004	0.08223	0.01474	0.19250	0.00000	0.00041		
2	0.00003	0.00002	0.00019	0.00131	0.00000	0.00222	0.00002	0.04704	0.01250	0.12367	0.00000	0.00055		
3	0.00004	0.00001	0.00015	0.00107	0.00000	0.00182	0.00002	0.04230	0.01207	0.14479	0.00000	0.00056		
.														
.														
.														

Figure 51. Sample VMT Mix Input File.

## MOVES Default Database (/MVSDB)

The /MVSDB identifier (optional, only if sourcetypeyear database table is output) and its corresponding text string specify the name of the database location of the MOVES default database tables. This database must be in the MySQL data folder. Only the name of the database is required.

## Population Factor Input File (/POPFI)

The /POPFI identifier and its corresponding text string specify the path and filename of the population factor input file, which lists the population factor applied to the calculated population prior to output. This is an optional input file and used only for adjusting the calculated population for the registration data year (i.e., a future year analysis). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the data identifier (PopFactor), followed by the equals symbol (=) and the population factor, separated by any combination of tabs and/or spaces. Figure 52 shows a sample population factor input file.

```
* NOTE: Comments can be included in jcf by placing the symbol * in the first column.  
* NOTE: Format is fixed with identifier PopFactor followed by the symbol = and the Year.  
* NOTE: PopFactor can be any combination of capital or lowercase letters.  
* NOTE: The user can separate all data by any combination of tabs and/or spaces.  
PopFactor = 1.38
```

**Figure 52. Sample Population Factor Input File.**

## Year ID Input File (/YRID)

The /YRID identifier and its corresponding text string specify the path and filename of the year ID input file, which lists the year for performing the calculations. This is an optional input file and used only if the population factor input file is in the jcf (i.e., a future year analysis). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the data identifier (Year), followed by the equals symbol (=) and the year, separated by any combination of tabs and/or spaces. Figure 53 shows a sample year ID input file.

```
* NOTE: Comments can be included in jcf by placing the symbol * in the first column.  
* NOTE: Format is fixed with identifier Year followed by the symbol = and the Year.  
* NOTE: Year can be any combination of capital or lowercase letters.  
* NOTE: The user can separate all data by any combination of tabs and/or spaces.  
Year = 2018
```

**Figure 53. Sample Year ID Input File.**

## MySQL Connection Inputs File (/MYSQL)

The /MYSQL identifier (optional, only if sourcetypeyear database table is output) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 54 shows a sample MySQL connections input file.

```
* Comments can be added by placing * in first column.
* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by
* any combination of tabs and/or spaces.
* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any
* combination of tabs and/or spaces.
* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any
* combination of tabs and/or spaces.
* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.
Port Number = 3307
Username = moves
Password = moves
```

**Figure 54. Sample MySQL Connections Input File.**

## Registration Data Summary Output File (/REGDO)

The /REGDO identifier and its corresponding text string specify the path and filename of the registration data summary output file. This output file is tab-delimited and lists a summary of the vehicle registrations from the age registrations input file, the diesel heavy-duty registrations input file, and the gas heavy-duty registrations input file used in calculating the vehicle population.

## Tab-Delimited Population by SUT/Fuel Type Output File (/TPFTO)

The /TPFTO identifier and its corresponding text string specify the path and filename of the tab-delimited population by SUT/fuel type output file. For the calculation year (either the latest year from the registration data inputs or the year ID input file, if used), this output file lists the calculated population for each SUT/fuel type combination in the VMT mix input file. This output file can be used with the EmsCalc utility to estimate emissions or with the OffNetActCalc utility to calculate the source hours parked (SHP) and starts off-network activity.

## sourcetypeyear Database Table Output (/DSTYO)

The /DSTYO identifier and its corresponding text string specify the name of the database where the sourcetypeyear database table is output. This output is optional and, if created, the database

will be in the MySQL data folder. Only the name of the database is required and the format of this database table is consistent with the MOVES default sourcetypeyear database table.

**sourcetypeyear Tab-Delimited Output (/TSTYO)**

The /TSTYO identifier and its corresponding text string specify the path and filename of the tab-delimited version of the sourcetypeyear database table output. This output file is optional and contains the same data as the sourcetypeyear database table output, except in tab-delimited form.

## OFFNETACTCALC: OFF-NETWORK ACTIVITY BUILDER

The OffNetActCalc utility creates the SHP, source hours idling (SHI), auxiliary power unit (APU) hours, and starts activity. The three files created by this utility are all tab-delimited and are in the proper format for use with the other utilities contained in MOVES2014aUtl.

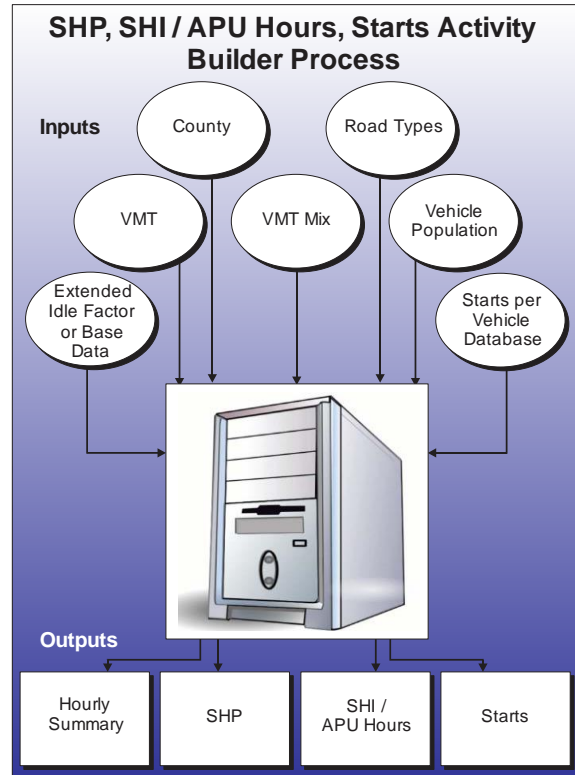
### OPERATION

The OffNetActCalc utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or OffNetActCalc cannot open any of these input files due to an incorrect path and/or filename or any other access error, then OffNetActCalc results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the county name input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). After locating this line, the utility ignores the first word of *County*, uses the number following the text *County* for the county number, and uses the remaining text following the county number for the county name. Since the last three characters of the county name are part of the county FIPS code, the utility also creates the full county FIPS code by appending the last three characters of the county name to the digits 48 (i.e., if the county name is “Anderson 001”, the full county FIPS code is 48001).

The next input file the utility processes is the analysis year VMT road type designations input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMT\_RDCODE*, *VMT\_ATCODE*, *VMTMIX\_RDCODE*,



*RATES\_RDCODE*, and *MOVES\_RDCODE* and the utility sets the location of the VMT road type code, VMT area type code, VMT mix road type, emissions rate road type code, and MOVES road type code data, respectively. For each remaining line, the utility saves the VMT mix road type code, rates road type code, and MOVES road type code in the analysis year portion of the VMT road type designations array based on the VMT road type and VMT area type.

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

The next input file the utility processes is the time period designation input file, which should only be included in the jcf if the four time-period VMT mix is used in the jcf. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *TIMEPER* and *TIMEPERIOD* and sets the location of the hour and time period data, respectively. For each remaining line, the utility saves the time period (must be AM Peak, Midday, PM Peak, or Overnight) in the time periods array based on the hour.

The next input file the utility processes is either the analysis year 24-hour VMT mix input file or the analysis year AM Peak VMT mix input file. If the time period designation input file was found in the jcf, then the utility processes the analysis year AM peak VMT mix file. Otherwise, the utility processes the analysis year 24-hour VMT mix input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMX\_RDCODE*, *MOVES\_STCODE*, *MOVES\_FTCODE*, and *VMTMIX* and sets the location of the VMT mix road type code, MOVES SUT code, MOVES fuel type code and VMT mix data, respectively. For each remaining line, the utility saves the VMT mix in the first section of the analysis year VMT mix array based on the VMT mix road type code and MOVES SUT code. The utility repeats this process until processing all of the data lines. The utility also calculates the total of the VMT mix for each SUT across all of the MOVES road types in the VMT mix input file (i.e., the all road types VMT mix by SUT) and saves this data in the VMT mix array as well.

If the time period designation input file was found in the jcf, the utility also processes the analysis year Midday, analysis year PM Peak, and analysis year Overnight VMT mix input files. Using the same process as the analysis year AM Peak VMT mix input file, the utility processes each of the remaining VMT mix input files, saving the Midday VMT mix data in section 2 of the analysis year VMT mix array, the PM Peak data in section 3 of the analysis year VMT mix array, and the Overnight VMT mix data in section 4 of the analysis year VMT mix array.

The next input file the utility processes is the vehicle population input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text

*MOVES\_STCODE*, *MOVES\_FTCODE*, and *Population* and sets the location of the MOVES source type code data, MOVES fuel type code, and population data, respectively. For each remaining line, the utility saves the population in the population array based on the MOVES source type code and MOVES fuel type code.

The next input file the utility processes is the analysis year VMT listing input file. For each line, the utility skips any blank lines and lines with an asterisk symbol (\*) in the first column. For all other lines, the utility saves the hourly VMT file (text following the hour number) in the analysis year portion of the VMT input list array based on the hour (number following the text “Hour”) for the respective hourly VMT file.

If the tab-delimited hourly summary output file was found in the jcf, the utility then processes the hourly summary header input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is saved as the output header.

If the hotelling factor input file was found in the jcf, the utility then processes the hotelling factor input file. The utility reads each line, skipping blank lines or lines that have the asterisk symbol (\*) in the first column (indicating a comment line), until it locates the text *ExtIdleFact* as the first word. Using this line, the utility then sets the hoteling factor as the number following the equal (=) sign.

If the base hoteling hours input file is found in the jcf, the utility then processes the base hoteling hours, the base VMT road type designations, base VMT mix, and base VMT listing input files. All of these files are required if the base hoteling hours input file is found in the jcf. For the base hoteling hours input file, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *FIPS* and *HOTELLINGHRS* (other data may be included following *HOTELLINGHRS* and will not affect the utility execution) and sets the location of the FIPS and hoteling hours data, respectively. The utility then reads each line until it locates the appropriate FIPS code and saves the hoteling hours as the base county hoteling hours.

For the base VMT road type designations input file, the utility follows the same processing procedures as the analysis year VMT road type designations input file with the VMT mix road type code, rates road type code, and MOVES road type code in the base year portion of the VMT road type designations array based on the VMT road type and VMT area type. For the base VMT mix input files, the utility follows the same process procedures as the analysis year VMT mix input files with the VMT mix saved in the base VMT mix array. For the base VMT listing input file, the utility follows the same processing procedures as the analysis year VMT listing input file with the hourly VMT files saved in the base year porting of the VMT input list array.

If the tab-delimited extended idle output file is found in the jcf, the utility then processes the age distribution, hoteling activity distribution, and relative mileage accumulation rates input files. For the age distribution input file, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *sourceTypeID*, *yearID*, *ageID*, and



*ageFraction* and sets the location of the source type ID, year ID, age ID, and age fraction data, respectively. The utility then reads each remaining line and if the source type ID is 62, the utility saves the age fraction data in the age distribution array based on the age ID.

For the hoteling activity distribution input file, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *beginModelYearID*, *endModelYearID*, *opModeID*, and *opModeFraction* and sets the location of the beginning model year, ending model year, operating mode ID, and operating mode fraction data, respectively. For each remaining line, the utility processes each line that has an operating mode ID of 200 or 201 (all other lines are skipped). For those lines with an operating mode of 200 or 201, the utility uses the input model year range (beginning model year to ending model year) to determine if the operating mode fraction is applicable to the analysis year (analysis year is input directly through the jcf). For each input year in the input model year range, the operating mode fraction is saved in the hoteling activity distribution array based on the operating mode (operating mode 200 data is saved in the SHI portion of the array and operating mode 201 data is saved in the APU portion of the array) if the input year is within 30 years of the analysis (i.e., analysis year minus input year is greater than or equal to 0 and less than or equal to 30).

For the relative mileage accumulation rates input file, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *sourceTypeID*, *ageID*, and *relativeMAR* and sets the location of the source type ID, age ID, and relative mileage accumulation rate data, respectively. The utility then reads each remaining line and if the source type ID is 62, the utility saves the relative mileage accumulation rate data in the relative mileage accumulation rate array based on the age ID.

If the tab-delimited starts output file was found in the jcf, the utility then processes the starts per vehicle input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, and *actpervehicle* and sets the location of the hour ID, source type ID, fuel type ID, and activity per vehicle data, respectively. The utility then reads each remaining line and saves the activity per vehicle data in the starts per vehicle array based on the hour ID, source type ID, and fuel type ID.

The utility uses the vehicle population data and the starts per vehicle data. For each hour, source type, and fuel type in the starts per vehicle array, the utility calculates the starts by multiplying the starts per vehicle by the vehicle population (based on the associated source type and fuel type).

For the analysis scenario first hourly VMT and speeds input, the utility applies the appropriate VMT mix (either the 24-hour VMT mix or the appropriate time period VMT mix as assigned by the user) to each link that has the desired county code; thus distributing the link VMT to each SUT/fuel type, which is added to the hourly SUT/fuel type VMT. The link VMT by SUT/fuel type is divided by the link speed to calculate the link VHT (or source hours operating [SHO]) by SUT/fuel type, which is added to the SUT fuel/type VHT. This calculation



process is repeated for each analysis scenario VMT and speeds input; therefore producing the analysis scenario hourly values for VMT by SUT/fuel type and for VHT by SUT/fuel type.

The analysis scenario hourly SUT/fuel type speed, total hours (or source hours), and SHP are then calculated. For each hour and SUT/fuel type, the hourly SUT/fuel type VMT is divided by the hourly SUT/fuel type VHT to calculate the hourly SUT/fuel type speed. The hourly SUT/fuel type total hours are set equal to the SUT/fuel type population. The hourly SUT/fuel type SHP is calculated by subtracting the hourly SUT/fuel type VHT (or SHO) from the hourly SUT/fuel type total hours. If the calculated SHP is negative (i.e., SHO is greater than the total hours), the SHP is set to 0.

If the hotelling factor input file was found in the jcf, the utility calculates the analysis scenario 24-hour hotelling hours as a direct function of SHO by multiplying the utility multiplies the CLhT\_Diesel analysis scenario 24-hour SHO by the hoteling factor, which represents the amount of extended idle time that must occur per SHO. If the base hoteling hours input file is found in the jcf, the utility calculates the analysis scenario 24-hour hoteling hours as a function of the base hotelling data. The utility calculates the base 24-hour CLhT\_Diesel VMT using the base VMT and speeds inputs and the base VMT mix with the same procedure used in the analysis scenario SHP calculations. The 24-hour analysis scenario CLhT\_Diesel VMT is then divided by the 24-hour base CLhT\_Diesel VMT to create a scaling factor, which is then applied to the base 24-hour hotelling hours to calculate the analysis scenario 24-hour hotelling hours.

The utility then calculates the analysis scenario hourly hotelling hours. The analysis scenario hourly CLhT\_Diesel SHO (from the SHP calculation process) is converted to hourly VHT fractions. The hourly hotelling fractions are calculated as the inverse of the hourly VHT fractions. The hourly hotelling fractions are then applied to the analysis scenario 24-hour hotelling hours to calculate the hourly hotelling hours. For each hour, the hourly hotelling hours are then compared to the hourly CLhT\_Diesel SHP. For those hours where the hotelling hours are greater than the SHP, hotelling hours are set to the SHP for that hour.

The utility then calculates the SHI fraction and the APU fraction using the source type age distribution (same distribution used in the MOVES runs), the relative mileage accumulation rates, and the hotelling activity distribution. Travel fractions for SUT 62 (CLhT) by ageID (0 through 30) are calculated by multiplying the age distribution by the appropriate relative mileage accumulation rate, which is then converted into a distribution by dividing the individual travel fraction (ageID 0 through 30) by the sum of the travel fractions. These travel fractions are then applied to the appropriate operating mode fractions from the hotelling activity distribution (operating mode 200) and summed to calculate the SHI fraction. Using a similar process, the APU fraction is calculated using the operating mode fractions for operating mode 201. For each hour the analysis scenario hotelling hours are multiplied by the SHI fraction to calculate the analysis scenario SHI activity and by the APU fraction to calculate the analysis scenario APU hours.

## JCF INPUT FILE

The jcf input file directs the OffNetActCalc utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (OffNetActCalc in this case), or the path and filename of the input or output file. Table 17 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 17. OffNetActCalc File Type Identifiers.**

<b>Identifier</b>	<b>Description</b>	<b>Text String Description</b>
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/CTY	County Name Input File	Path and Filename
/RDTD	Analysis VMT Road Type Designations Input File	Path and Filename
/VMX	Analysis VMT Mix Input File <sup>2</sup>	Path and Filename
/TMPD	Time Period Designation Input File <sup>3</sup>	Path and Filename
/VMXA, /VMXM, /VMXP, /VMXO	Analysis AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files, <sup>3</sup> respectively	Path and Filename
/POPI	Analysis Vehicle Population by Input File	Path and Filename
/VMT	Analysis VMT Listing Input File	Path and Filename
/HOTF	Extended Idle Factor Input File <sup>4</sup>	Path and Filename
/BHHR	Base Hotelling Hours Inputs File <sup>5</sup>	Path and Filename
/BVMT	Base VMT Listing Input File <sup>5</sup>	Path and Filename
/BRDTD	Base VMT Road Type Designations Input File	Path and Filename
/BVMX	Base VMT Mix Input File <sup>2, 5</sup>	Path and Filename
/BVMXA, /BVMXM, /BVMXP, /BVMXO	Base AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files, <sup>3, 5</sup> respectively	Path and Filename
/AGED	Age Distribution Input File <sup>6</sup>	Path and Filename
/YEAR	Year Input <sup>6</sup>	Value
/HADI	Hotelling Activity Distribution Input File <sup>6</sup>	Path and Filename
/RMAR	Relative Mileage Accumulation Rates Input File <sup>6</sup>	Path and Filename
/STRTI	Starts Per Vehicle Input File <sup>7</sup>	Path and Filename
/HRSH	Hourly Summary Header Input File	Path and Filename
/HRSO	Tab-Delimited Hourly Summary Output File	Path and Filename
/SHPO	Tab-Delimited SHP Output File	Path and Filename
/EXIO	Tab-Delimited SHI and APU Hours Output File	Path and Filename
/STRTO	Tab-Delimited Starts Output File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Only if the 24-hour VMT mix is used by the utility.

<sup>3</sup> Only if the time period VMT mix is used by the utility.

<sup>4</sup> Optional. Only used if the SHI output file (/EXIO) is to be written and based on the hoteling factor.

<sup>5</sup> Optional. Only used if the SHI output file (/EXIO) is to be written and based on the base hoteling hours data.

<sup>6</sup> Optional. Only used if the SHI output file (/EXIO) is to be written.

<sup>7</sup> Optional. Only used if the tab-delimited starts output file (/STRTO) is to be written.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (OffNetActCalc), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 55 shows a sample jcf.

```

/JOB      OffNetActCalc
/STEP     Step1 OffNetActCalc
/LST      C:\OffNetActCalc\Output\EIPaso_OffNetActCalc_2010swkd_BaseHot.lst
/CTY      C:\OffNetActCalc\Input\Counties\EIPaso.CTY
/RDTD     C:\OffNetActCalc\Input\ELPTDM_RdType_desig.txt
*/VMX     C:\OffNetActCalc\Input\vmx\D11_Dist_2010WK_VMTmix_24hr.vmx
* or
/TMPD     C:\OffNetActCalc\Input\HotelingFactor\TimePeriodDesignation.txt
/VMXA     C:\OffNetActCalc\Input\vmx\D11_Dist_2010WK_VMTmix_AMPK.vmx
/VMXM     C:\OffNetActCalc\Input\vmx\D11_Dist_2010WK_VMTmix_MDAY.vmx
/VMXP     C:\OffNetActCalc\Input\vmx\D11_Dist_2010WK_VMTmix_PMPK.vmx
/VMXO     C:\OffNetActCalc\Input\vmx\D11_Dist_2010WK_VMTmix_OVN.vmx
*
/POPI     C:\OffNetActCalc\Input\VehPop\elp_48141_VEHPOP_2010_StFtPop.tab
/VMT      C:\OffNetActCalc\Input\VTMlists\VTMLIST_2010SUMWKD.TXT
*/HOTF    C:\OffNetActCalc\Input\HotelingFactor\HotellingFact.txt
* or
*
/BHHR     C:\OffNetActCalc\Input\BaseData\BaseExtIdleHrs\2004Base_SummerWkd_HotellingHrs.txt
/BVMT     C:\OffNetActCalc\Input\BaseData\VMT\VTMLIST_2004Wk.TXT
/BRDTD    C:\OffNetActCalc\Input\ELPTDM_RdType_desig.txt
*/BVMX    C:\OffNetActCalc\Input\vmx\D11_Dist_2005WK_VMTmix_24hr.vmx
* or
/BVMXA    C:\OffNetActCalc\Input\vmx\D11_Dist_2005WK_VMTmix_AMPK.vmx
/BVMXM    C:\OffNetActCalc\Input\vmx\D11_Dist_2005WK_VMTmix_MDAY.vmx
/BVMXP    C:\OffNetActCalc\Input\vmx\D11_Dist_2005WK_VMTmix_PMPK.vmx
/BVMXO    C:\OffNetActCalc\Input\vmx\D11_Dist_2005WK_VMTmix_OVN.vmx
*
/AGED     C:\OffNetActCalc\Input\AgeDist_Test.tab
/YEAR     2010
/HADI     C:\OffNetActCalc\Input\HotellingActDist_Test.tab
/RMAR     C:\OffNetActCalc\Input\RelativeMAR.txt
/STRTI    C:\OffNetActCalc\Input\DefStartsPerVeh_Weekday_mvs10b.tab
/HRSH     C:\OffNetActCalc\Input\HrSummaryHdr\HRSUMMARYHDR_2010SUMWKD.TXT
/HRSO     C:\OffNetActCalc\Output\EIPaso_OffNetActCalc_2010swkd_HrSummary.tab
/SHPO     C:\OffNetActCalc\Output\EIPaso_OffNetActCalc_2010swkd_SHP.tab
/EXIO     C:\OffNetActCalc\Output\EIPaso_OffNetActCalc_2010swkd_ExtIdle.tab
/STRTO    C:\OffNetActCalc\Output\EIPaso_OffNetActCalc_2010swkd_Starts.tab

```

**Figure 55. Sample OffNetActCalc jcf.**

### **Job Identifier (/JOB)**

The /JOB (optional) identifier and its corresponding text string are comments used only as a description of the jcf. This utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP (required) identifier and its corresponding text string name the utility to run. The OffNetActCalc utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST (required) identifier and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input

- Date and time of the beginning of the utility execution;
- The utility (OffNetActCalc in this case) being executed;
- The utility compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of all input files (contents are included with some input files);
- An hourly summary of number of links from the hourly VMT input files, including the total number of links, the number of links with a VMT greater than 0, and the number of links with a VMT equal to 0;
- A summary of the VMT, VHT (SHO), speed (VMT/VHT), population, total hours, SHP, hoteling, SHI, APU hours, starts per vehicle, and starts by SUT/fuel type;
- The path and filename of all output files;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **County Name Input File (/CTY)**

The /CTY (required) identifier and its corresponding text string specify the path and filename of the county name input file. The county name input file specifies the county code within the hourly VMT and speeds files for generating the output. This input file consists of a single line containing the identifier text *COUNTY*, the county number, the county name and the county FIPS

code in a three-digit format (i.e., 000). The user can separate all of the data within the input line by a tab, space, or any combination of tabs and spaces. Figure 56 shows a sample county input file.

COUNTY 1 Upshur 459

**Figure 56. Sample County Input File.**

**Analysis VMT Road Type Designations Input File (/RDTD)**

The /RDTD (required) identifier and its corresponding text string specify the path and filename of the analysis VMT road type designations input file, which correlates each link road type/area type combination to a VMT mix road type, a rates road type, and a MOVES road type. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 18 shows the contents of the VMT road type designations input file header line.

**Table 18. VMT Road Type Designations Input File Header Line.**

Header Line Text	Description
VMT_RDcode	Link VMT and speeds road type code
VMT_ATcode	Link VMT and speeds area type code
VMTMIX_RDcode	VMT mix road type code
RATES_RDcode	Rates road type code
MOVES_RDcode	MOVES road type code
VMT_RDdesc	Link VMT and speeds road type code description <sup>1</sup>
VMT_ATdesc	Link VMT and speeds area type code description <sup>1</sup>
MOVES_RDdesc	MOVES road type code description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes data.

Following the header line, the user must list each link road type code/area type code combination (each on a separate line) with the combinations VMT mix road type code, rates road type code, and MOVES road type code, separated by any combination of tabs and/or spaces. Figure 57 shows a sample VMT road type/MOVES road type input file.

* NOTE: Lists the road type and area codes in the VMT input files and their MOVES designations.								
* NOTE: Descriptions (VMT_RDdesc, VMT_ATdesc, MOVES_RDdesc) are not used and are optional.								
VMT_RDcode	VMT_ATcode	VMTMIX_RDcode	RATES_RDcode	MOVES_RDcode	VMT_RDdesc	VMT_ATdesc	MOVES_RDdesc	
0	3	3	3	3	RUR IH FWY 110	RUR	Rural_Unrest	
2	3	2	2	2	RUR OTH PRIN ART 130	RUR	Rural_Rest	
3	3	2	2	2	RUR MINOR ART 150	RUR	Rural_Rest	

**Figure 57. Sample VMT Road Type Designations Input File.**

**Analysis 24-Hour VMT Mix Input File (/VMX)**

The /VMX identifier (optional, only if 24-hour VMT mix is to be used) and its corresponding text string specify the path and filename of the analysis 24-hour VMT mix input file that specifies the analysis 24-hour mix of vehicles by SUT/fuel type combination by road type, with codes 2 through 9 being the MOVES roadway types (see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). This input file is optional and should only be used if the user intends to use the 24-hour VMT mix in the calculations. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 19 shows the contents of the 24-hour VMT mix input file header line.

**Table 19. 24-Hour VMT Mix Input File Header Line.**

Header Line Text	Description
VMX_RDcode	VMT mix road type code
MOVES_STcode	MOVES SUT code
MOVES_FTcode	MOVES fuel type code
VMTmix	VMT mix value
VMX_RDdesc	VMT mix road type code description <sup>1</sup>
MOVES_STdesc	MOVES SUT code description <sup>1</sup>
MOVES_FTdesc	MOVES fuel type code description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes and VMT mix data.

Following the header line, the user must list each MOVES SUT/MOVES fuel type combination for each VMT mix road type (each on a separate line) with the combinations' associated VMT mix value, separated by any combination of tabs and/or spaces. The utility will calculate the emissions only for those MOVES SUT/MOVES fuel type combinations in this input file. Figure 58 shows a sample 24-Hour VMT mix input file.

* NOTE: Waco TxDOT District – 2000 Friday						
* NOTE: Descriptions (VMX_RDdesc, MOVES_STdesc, MOVES_FTdesc) are not used and are optional.						
VMX_RDcode	MOVES_STcode	MOVES_FTcode	VMTmix	VMX_RDdesc	MOVES_STdesc	MOVES_FTdesc
2	11	1	0.00055	Rural_Rest	MC	Gas
2	21	1	0.56562	Rural_Rest	PC	Gas
2	21	2	0.00170	Rural_Rest	PC	Diesel

**Figure 58. Sample 24-Hour VMT Mix Input File.**

### Time Period Designation Input File (/TMPD)

The /TMPD identifier (optional, only if time period VMT mix is to be used) and its corresponding text string specify the path and filename of the time period designation input file, which specifies which time period VMT mix (AM Peak, Midday, PM Peak, or Overnight) each hour will use in the calculations. This input file is optional and should only be used if the time period VMT mix input files are in the jcf. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 20 shows the contents of the time period designation input file header line.

**Table 20. Time Period Designation Input File Header Line.**

Header Line Text	Description
TIMEPER	Hour
TIMEPERIOD	Time Period

Following the header line, the user must list each hour (1, 2, 3, ..., 23, 24) with the hour's associated time period (AM Peak, Midday, PM Peak, Overnight), separated by any combination of tabs and/or spaces. Figure 59 shows a sample time period designation input file.



TIMEPER	TIMEPERIOD
01	Overnight
02	Overnight
03	Overnight
04	Overnight
05	Overnight
06	Overnight
07	AM Peak
08	AM Peak
09	AM Peak
10	Midday
11	Midday
12	Midday
13	Midday
14	Midday
15	Midday
16	Midday
17	PM Peak
18	PM Peak
19	PM Peak
20	Overnight
21	Overnight
22	Overnight
23	Overnight
24	Overnight

**Figure 59. Sample Time Period Designation Input File.**

**Analysis AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/VMXA, /VMXM, /VMXP, /VMXO)**

The /VMXA, /VMXM, /VMXP, and /VMXO identifiers (optional, only if time period VMT mix is to be used) and corresponding text string specify the path and filename of the analysis AM Peak, Midday, PM Peak, and Overnight VMT mix input files, respectively. These input files specify the time period mix of vehicles by SUT/fuel type combination for each of the four on-network MOVES roadway types (see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. The format of the time period VMT mix input files is the same as the 24-hour VMT mix input file (Table 19 shows the contents of the header line and Figure 58 shows a sample of the input file).

**Analysis Vehicle Population Input File (/POPI)**

The /POPI identifier (required) and its corresponding text string specify the path and filename of the analysis vehicle population input file, which is a tab-delimited file that specifies the vehicle population by SUT/fuel type combination. This input file is the same as the tab-delimited population by SUT/fuel type output file created by the VehPopulationBuild utility. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first

column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 21 shows the contents of the tab-delimited population by SUT/fuel type input file header line.

**Table 21. Tab-Delimited Population by SUT/Fuel Type Input File Header Line.**

Header Line Text	Description
yearID	Year <sup>1</sup>
MOVES_STcode	MOVES source type code
MOVES_FTcode	MOVES fuel type code
Population	Population value
MOVES_STdesc	MOVES source type code description <sup>2</sup>
MOVES_FTdesc	MOVES fuel type code description <sup>2</sup>

<sup>1</sup> Optional, and not used by the utility.

<sup>2</sup> Optional, but must appear after codes and population value.

Following the header line, the user must list each MOVES source type/MOVES fuel type combination (each on a separate line) with the combinations associated population value, separated by any combination of tabs and/or spaces. Figure 60 shows a sample tab-delimited population by SUT/fuel type input file.

* Comments can be added by placing a * in first column.						
yearID	MOVES_STcode	MOVES_FTcode	Population	MOVES_STdesc	MOVES_FTdesc	
2009	11	1	49862	MC	Gas	
2009	21	1	1789653	PC	Gas	
2009	21	2	1782	PC	Diesel	

**Figure 60. Sample Tab-Delimited Population by SUT/Fuel Type Input File.**

### Analysis VMT Listing Input File (/VMT)

The /VMT identifier (required) and its corresponding text string specify the path and filename of the analysis VMT listing input file. This input file consists of the identifier text *Hour*, the hour number (beginning with hour 1, which is 12:00-to-1:00 a.m.), and the path and filename for the respective hourly VMT and speeds file. The hourly VMT and speeds files in this input file can be the output from either the TRANSVMT or VirtualLinkVMT utilities. Figure 61 shows a sample VMT listing input file.

Hour	1	C:\VMT\Output\upshur_2005fr_summer.T01
Hour	2	C:\VMT\Output\upshur_2005fr_summer.T02
Hour	3	C:\VMT\Output\upshur_2005fr_summer.T03

**Figure 61. Sample VMT Listing Input File.**

### Hotelling Factor Input File (/HOTF)

The /HOTF identifier (optional, only used if the SHI and APU hours are calculated based on the hotelling factor) and its corresponding text string specify the path and filename of the extended idle factor input file, which lists the factor that is applied to the 24-hour SHO to estimate the 24-hour hotelling for MOVES source type 62 and fuel type 2 (used to calculate SHI and APU hours). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only input line, other than the optional comments, consists of the text *HotellingFact* = followed by the hotelling factor, separated by any combination of tabs and/or spaces. Figure 62 shows a hotelling factor input file.

<ul style="list-style-type: none"> <li>* Comments can be added by placing * in first column.</li> <li>* Format of input file is fixed, with " HotellingFact = " followed by the factor</li> <li>* that will be applied to the 24-hour SHO to calculate the 24-hour hotelling,</li> <li>* separated by any combination of tabs and/or spaces.</li> <li>* " HotellingFact" text can be any combination of capital or lowercase letters.</li> </ul> <p>HotellingFact = 0.59</p>
--

**Figure 62. Sample Hotelling Factor Input File.**

### Base Hotelling Hours Input File (/BHHR)

The /BHHR identifier (optional, only used if the SHI and APU hours are calculated based on the base hotelling hours) and its corresponding text string specify the path and filename of the base hotelling hours input file, which lists the base 24-hour hotelling by county. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 22 shows the contents of the base hotelling hours input file header line.

**Table 22. Base Hotelling Hours Input File Header Line.**

Header Line Text	Description
FIPS	County FIPS code
HOTELLINGHRS	Base 24-hour Hotelling Hours
COUNTY	County Name <sup>1</sup>

<sup>1</sup> Optional, but must appear after the codes data.

Following the header line, the user must list the desired county FIPS code with the county's associated base 24-hour hoteling hours, separated by any combination of tabs and/or spaces. If the county name is included in this input file, the county name must appear as the last column of data, with the base 24-hour hoteling hours and the county name by any combination of tabs and/or spaces. Multiple counties may be listed in this input file as long as the desired county is listed. Figure 63 shows a sample hoteling hours input file.

* Comments can be added by placing * in first column.		
* Data Header (FIPS, HOTELLINGHRS) is fixed with each field		
* separated by spaces, tabs, or combination of spaces and tabs.		
* COUNTY is optional and must appear as last column.		
FIPS	HOTELLINGHRS	COUNTY
48001	209.8006596	Anderson
48003	209.8006596	Andrews
48005	1141.315588	Angelina

**Figure 63. Sampe Hotelling Hours Input File**

### Base VMT Listing Input File (/BVMT)

The /BVMT identifier (optional, only used if the SHI and APU hours are calculated based on the base hoteling hours) and its corresponding text string specify the path and filename of the base VMT listing input file. This input file consists of the identifier text *Hour*, the hour number (beginning with hour 1, which is 12:00-to-1:00 a.m.), and the path and filename for the respective base hourly VMT and speeds file. All 24 hours must be listed in this input file. The base hourly VMT and speeds files in this input file can be the output from either the TRANSVMT or VirtualLinkVMT utilities. The format of this input file is the same as the analysis VMT listing input file (see Figure 61).

### Base VMT Road Type Designations Input File (/BRDTD)

The /BRDTD identifier (optional, only used if the SHI and APU hours are calculated based on the base hoteling hours) and its corresponding text string specify the path and filename of the base VMT road type designations input file that correlates each link road type/area type combination to a MOVES road type. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. The format of this input file is the same as the analysis VMT road type designations input file (see Table 18 and Figure 57).

### Base 24-Hour VMT Mix Input File (/BVMX)

The /BVMX identifier (optional, only used if the SHI and APU hours are calculated based on the base hoteling hours and if 24-hour VMT mix is to be used) and its corresponding text string specify the path and filename of the base 24-hour VMT mix input file that specifies the base 24-hour mix of vehicles by SUT/fuel type combination by road type, with codes 2 through 9 being the MOVES roadway types (see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). This input file is optional and should only be

used if the user intends to use the 24-hour VMT mix in the calculations. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. This format of this input file is identical to the analysis 24-hour VMT mix input file (see Table 19 and Figure 58).

**Base AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/BVMXA, /BVMXM, /BVMXP, /BVMXO)**

The /BVMXA, /BVMXM, /BVMXP, and /BVMXO identifiers (optional, only used if the SHI and APU hours are calculated based on the base hoteling hours and time period VMT mix is to be used) and corresponding text string specify the path and filename of the base AM Peak, Midday, PM Peak, and Overnight VMT mix input files, respectively. These input files specify the time period mix of vehicles by SUT/fuel type combination for each of the on-network MOVES roadway types (see Appendix A). Any additional road types must be included after the roadway-based MOVES roadway types (i.e., codes after 9). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. The format of these input files is the same as the 24-hour VMT mix input file (see Table 19 and Figure 58).

**Age Distribution Input File (/AGED)**

The /AGED identifier (optional, only used if the SHI and APU hours are calculated) and its corresponding text string specify the path and filename of the age distribution input file, which lists the age distribution (in MOVES format) for each SUT (only SUT 62 is required). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 23 shows the contents of the base hoteling hours input file header line.

**Table 23. Age Distribution Input File Header Line.**

<b>Header Line Text</b>	<b>Description</b>
sourceTypeID	sourceTypeID
yearID	yearID
ageID	ageID
ageFraction	Age Fraction

Following the header line, the user must list the sourceTypeID (only SUT 62 is required), yearID (one year only), ageID (0 through 30) and associated age fraction, separated by any combination of tabs and/or spaces. Only one year of data can be in this input file and only one

combination of sourceTypeID and ageID can appear per line. Figure 64 shows a sample age distribution input file.

sourceTypeID	yearID	ageID	ageFraction
62	2013	0	0.040322086
62	2013	1	0.050060289
62	2013	2	0.032688888
62	2013	3	0.018454558
...	...	...	...
62	2013	30	0.019772741

**Figure 64. Sample Age Distribution Input File.**

### Analysis Year Input (/YEAR)

The /YEAR identifier (optional, only used if the SHI and APU hours are calculated) and its corresponding numeric value specify the analysis year. No input file is required, only the analysis year. See the sample jcf (Figure 55) for an example of this input.

### Hotelling Activity Distribuiton Input File (/HADI)

The /HADI identifier (optional, only used if the SHI and APU hours are calculated) and its corresponding text string specify the path and filename of the hoteling activity distribution input file, which lists the hoteling activity distribution for each beginning model year and ending model year combination by operating mode (same as the MOVES format). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 24 shows the contents of the base hoteling hours input file header line.

**Table 24. Hotelling Activity Distribution Input File Header Line.**

Header Line Text	Description
beginModelYearID	Beginning Model Year
endModelYearID	Ending Model Year
opModeID	Operating Mode
opModeFraction	Operating Mode Fraction

Following the header line, the user must list the beginModelYearID, endModelYearID, opModeID (200 and 201 are required) and associated opModeFraction, separated by any combination of tabs and/or spaces. Only one combination of beginModelYearID, endModelYearID, and opModeID can appear per line. Figure 65 shows a hoteling activity distribution input file.

beginModelYearID	endModelYearID	opModeID	opModeFraction
1960	2009	200	1
2010	2050	200	0.7
1960	2009	201	0
2010	2050	201	0.3
1960	2009	203	0
2010	2050	203	0
1960	2009	204	0
2010	2050	204	0

**Figure 65. Sample Hotelling Activity Distribution Input File.**

### Relative Mileage Accumulation Rates Input File (/RMAR)

The /RMAR identifier (optional, only used if the SHI and APU hours are calculated) and its corresponding text string specify the path and filename of the relative mileage accumulation rates input file, which lists the relative mileage accumulation rates for each SUT and age (same as the MOVES format). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 25 shows the contents of the relative mileage accumulation rates input file header line.

**Table 25. Relative Mileage Accumulation Rates Input File Header Line.**

Header Line Text	Description
sourceTypeID	SUT ID
ageID	Age ID
relativeMAR	Relative Mileage Accumulation Rate

Following the header line, the user must list the sourceTypeID (only SUT 62 required), ageID (0 through 30), and associated relativeMAR, separated by any combination of tabs and/or spaces. Only one combination of sourceTypeID and ageID can appear per line. Figure 66 shows a sample relative mileage accumulation rates input file.

sourceTypeID	ageID	relativeMAR
62	0	1
62	1	1
62	2	1
...	...	...
62	29	0.081355831
62	30	0.057110861

**Figure 66. Sample Relative Mileage Accumulation Rates Input File.**

### Starts per Vehicle Input File (/STRTI)

The /STRTI identifier (optional, only used if the starts are calculated) and its corresponding text string specify the path and filename of the starts per vehicle input file, which lists the starts per vehicle for each SUT/fuel type by hour. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 26 shows the contents of the starts per vehicle input file header line.

**Table 26. Start per Vehicle Input File Header Line.**

Header Line Text	Description
hourID	Hour ID
sourceTypeID	SUT ID
fuelTypeID	Fuel Type ID
actpervehicle	Starts per Vehicle

Following the header line, the user must list the hourID, sourceTypeID, fuelTypeID, and associated starts per vehicle for each hour, SUT, and fuel type combination, separated by any combination of tabs and/or spaces. Only one combination of hourID, sourceTypeID, and fuelTypeID can appear per line. Figure 67 shows a sample starts per vehicle input file.

hourID	sourceTypeID	fuelTypeID	actpervehicle
1	11	1	0.001806685
2	11	1	0.002710027
3	11	1	0.000903341
4	11	1	0.001355013
5	11	1	0.000451672
...	...	...	...

**Figure 67. Sample Starts per Vehilce Input File.**

### Tab-Delimited Hourly Summary Header Input File (/HRSH)

The /HRSH identifier and its corresponding text string specify the path and filename of the tab-delimited hourly summary header input file, which lists the header that will be written to the tab-delimited hourly summary output file. This input file is optional and if this input file is not included in the jcf, then no header will be written to the tab-delimited hourly summary output file. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only input line, other than the optional comments, is the header.

### Tab-Delimited Hourly Summary Output File (/HRSO)

The /HRSO identifier and its corresponding text string specify the path and filename of the tab-delimited hourly summary output file. This output file is optional and lists the SUT/fuel type



VMT, VHT (SHO), speed (VMT/VHT), vehicle population, total hours, calculated SHP, final SHP, calculated hoteling hours, final hoteling hours, SHI, APU hours, starts per vehicle, and starts by hour (including a 24 hour total).

#### **Tab-Delimited SHP Output File (/SHPO)**

The /SHPO identifier and its corresponding text string specify the path and filename of the tab-delimited SHP output file. This output file is optional and lists the SUT, fuel type, SUT/fuel type description, and SHP by hour. This output file is in the proper format for use with the EmsCalc utility when estimating emissions based on the rates per activity emissions rates.

#### **Tab-Delimited SHI and APU Hours Output File (/EXIO)**

The /EXIO identifier and its corresponding text string specify the path and filename of the tab-delimited SHI and APU hours output file. This output file is optional and lists the SUT, fuel type, SUT/fuel type description, hoteling, SHI, and APU hours for each hour. This output file is in the proper format for use with the EmsCalc utility when estimating emissions based on the rates per activity emissions rates.

#### **Tab-Delimited Starts Output File (/STRTO)**

The /STRTO identifier and its corresponding text string specify the path and filename of the tab-delimited starts output file. This output file is optional and lists the SUT, fuel type, SUT/fuel type description, and starts by hour. This output file is in the proper format for use with the EmsCalc utility when estimating emissions based on the rates per activity emissions rates.



## OFFNETACTSUMMARY: OFF-NETWORK ACTIVITY SUMMARY

The OffNetActSummary utility creates tab-delimited summaries of the off-network activity (vehicle population, SHP, starts, and/or hoteling/SHI/APU hours). The user can specify multiple off-network activity input files for each off-network, which allows the user to create regional summaries. Four different tab-delimited output files can be created using this utility: 24-hour, hourly, hourly by SUT/fuel type, and SUT/fuel type.

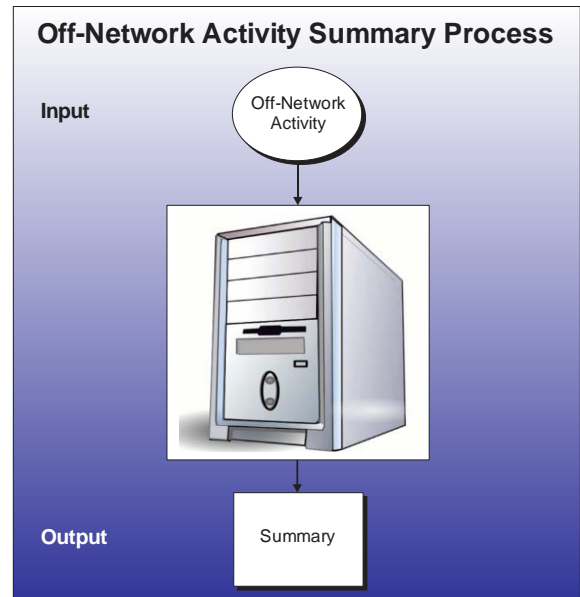
### OPERATION

The OffNetActSummary utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or OffNetActSummary cannot open any of these input files due to an incorrect path and/or filename or any other access error, then OffNetActSummary results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the header input file. If this input file is not found in the jcf, then the utility does not process this input and no header will be written to the tab-delimited off-network activity summary output file. Otherwise, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column. The utility saves this line as the header line.

The utility then processes the off-network category input files. For the vehicle population input file, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *County* and *TabDelimitedFile* and sets the location of the county identifier (typically the FIPS code) and the vehicle population tab-delimited file name for the associated



county, respectively. The utility then reads each remaining line and saves the county and associated vehicle population tab-delimited file name in the vehicle population files array. Using a similar process, the utility processes the SHP input file (saving the county and associated starts tab-delimited file name in the SHP files array), the starts input file (saving the county and associated starts tab-delimited file name in the starts files array), and the hoteling/SHI/APU hours input file (saving the county and associated hoteling/SHI/APU hours tab-delimited file name in the hoteling/SHI/APU hours files array).

After building a master county list, the utility then processes each individual vehicle population file in the vehicle population files array. For each vehicle population file, the utility reads each line, locating the first line that has the text *yearID* as the first word. This line is the header line consisting of the text *yearID*, *MOVES\_STcode*, *MOVES\_FTcode* and *Population* (other data may appear following *Population* that will not be used by the utility) and sets the location of the year, SUT code, fuel type code and the vehicle population, respectively. The utility then reads each remaining line and adds the vehicle population to the vehicle population portion of the 24-hour totals array by county; thus creating a county total vehicle population. The utility also adds the vehicle population to the vehicle population portion of the vehicle type 24-hour array by county, SUT, and fuel type and to the vehicle population portion of the hourly vehicle type array by county, hour (24-hour for vehicle population), SUT, and fuel type.

The utility then processes each individual SHP file in the SHP files array. For each SHP file, the utility reads each line, locating the first line that has the text *hourID* as the first word. This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, and *SHP* (other data may be included but will not be used by the utility) and sets the location of the hour, SUT code, fuel type code, and the SHP, respectively. The utility then reads each remaining line and adds the SHP to the SHP portion of the 24-hour totals array by county; thus creating a county total SHP. The utility also adds the SHP to the SHP portion of the vehicle type 24-hour array by county, SUT, and fuel type and to the SHP portion of the hourly vehicle type array by county, hour, SUT, and fuel type.

The utility then processes each individual starts file in the starts files array. For each starts file, the utility reads each line, locating the first line that has the text *hourID* as the first word. This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, and *Starts* (other data may be included but will not be used by the utility) and sets the location of the hour, SUT code, fuel type code, and the starts, respectively. The utility then reads each remaining line and adds the starts to the starts portion of the 24-hour totals array by county; thus creating a county total starts. The utility also adds the starts to the starts portion of the vehicle type 24-hour array by county, SUT, and fuel type and to the starts portion of the hourly vehicle type array by county, hour, SUT, and fuel type.

The utility then processes each individual hoteling/SHI/APU hours file in the hoteling/SHI/APU hours files array. For each hoteling/SHI/APU hours file, the utility reads each line, locating the first line that has the text *hourID* as the first word. This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, *Hotelling*, *ExtIdle*, and *APU* (other data may be included but will not be used by the utility) and sets the location of the hour, SUT code, fuel type code, hoteling hours, SHI, and the APU hours, respectively. The utility then reads

each remaining line and adds the hotelling to the hotelling portion of the 24-hour totals array, SHI to the SHI portion of the 24-hour totals array, and APU hours to the APU hours portion of the 24-hour totals array by county; thus creating a county total for hoteling, SHI, and APU hours. The utility also adds the hoteling, SHI, and APU hours to their respective portions of the vehicle type 24-hour array by county, SUT, and fuel type and to their respective portions of the hourly vehicle type array by county, hour, SUT, and fuel type.

## JCF INPUT FILE

The jcf input file directs the OffNetActSummary utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (OffNetActSummary in this case), or the path and filename of the input or output file. Table 27 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 27. OffNetActSummary File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/POPI	Vehicle Population Input File <sup>2</sup>	Path and Filename
/SHPI	SHP Input File <sup>2</sup>	Path and Filename
/SRTI	Starts Input File <sup>2</sup>	Path and Filename
/HEAI	Hotelling/SHI/APU Hours Input File <sup>2</sup>	Path and Filename
/HDRO	Header Input File <sup>2</sup>	Path and Filename
/TABO	24-Hour Summary Output File <sup>2</sup>	Path and Filename
/HTABO	Hourly Summary Output File <sup>2</sup>	Path and Filename
/VTABO	24-hour Vehicle Type Summary Output File <sup>2</sup>	Path and Filename
/ATABO	Hourly Vehicle Type Summary Output File <sup>2</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (OffNetActSummary), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 68 shows a sample jcf.

/JOB	OffNetActSummary
/STEP	Step1 OffNetActSummary
/LST	C:\OffNetActSummary\Output\OffNetActSummary_Test.lst
/TABO	C:\OffNetActSummary\Output\OffNetActSummary_Test.tab
/HTABO	C:\OffNetActSummary\Output\OffNetActSummary_Test_Hourly.tab
/VTABO	C:\OffNetActSummary\Output\OffNetActSummary_Test_VTYPE.tab
/ATABO	C:\OffNetActSummary\Output\OffNetActSummary_Test_HourlyVTYPE.tab
/HDRO	C:\OffNetActSummary\Input\OffNetActSummary_SHI_Test_hdr.txt
/POPI	C:\OffNetActSummary\Input\VehPop_TabDelimitedInputFiles.txt
/SHPI	C:\OffNetActSummary\Input\SHP_TabDelimitedInputFiles.txt
/SRTI	C:\OffNetActSummary\Input\Starts_TabDelimitedInputFiles.txt
/HEAI	C:\OffNetActSummary\Input\HotExIdleAPU_TabDelimitedInputFiles.txt

**Figure 68. Sample OffNetActSummary jcf.**

**Job Identifier (/JOB)**

The /JOB identifier and its corresponding text string are comments used only as a description of the jcf. This utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

**Step Identifier (/STEP)**

The /STEP identifier and its corresponding text string name the utility to run. The OffNetActSummary utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

**Listing File Output (/LST)**

The /LST identifier and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the inputs and outputs for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (OffNetActSummary in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of all input files;
- The path and filename of all output files;

- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### Vehicle Population Input File (/POPI)

The /POPI identifier (optional) and its corresponding text string specify the path and filename of the vehicle population input file, which specifies the tab-delimited vehicle population file (output from the VehPopulationBuild utility or equivalent format) for each county. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 28 shows the contents of the vehicle population input file header line.

**Table 28. Vehicle Population Input File Header Line.**

Header Line Text	Description
County	Hour ID
TabDelimitedFile	Path and Filename of Tab-Delimited File

Following the header line, the user must list the county (identifier, typically FIPS code) and the county's respective tab-delimited vehicle population file. Only one county can appear per line. Figure 69 shows a sample vehicle population input file.

* Comments can be placed at any line by putting a * in the first column	
* The County can be any text or number	
* The County and TabDelimitedFile can be separated by any combination of	
* spaces and/or tabs	
County	TabDelimitedFile
48001	C:\OffNetActSummary\Input\VehPop\48001_2018_StFtPop.tab
48003	C:\OffNetActSummary\Input\VehPop\48003_2018_StFtPop.tab

**Figure 69. Sample Vehicle Population Input File.**

### SHP Input File (/SHPI)

The /SHPI identifier (optional) and its corresponding text string specify the path and filename of the SHP input file, which specifies the tab-delimited SHP file (output from the OffNetActCalc utility or equivalent format) for each county. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 29 shows the contents of the SHP input file header line.

**Table 29. SHP Input File Header Line.**

<b>Header Line Text</b>	<b>Description</b>
County	Hour ID
TabDelimitedFile	Path and Filename of Tab-Delimited File

Following the header line, the user must list the county (identifier, typically FIPS code) and the county's respective tab-delimited SHP file. Only one county can appear per line. Figure 70 shows a sample SHP input file.

* Comments can be placed at any line by putting a * in the first column
* The County can be any text or number
* The County and TabDelimitedFile can be separated by any combination of spaces
*       and/or tabs
County    TabDelimitedFile
48001    C:\OffNetActSummary\Input\OffNetAct\48001_2018_OffNetActCalc_SHP.tab
48003    C:\OffNetActSummary\Input\OffNetAct\48003_2018_OffNetActCalc_SHP.tab

**Figure 70. Sample SHP Input File.**

### **Starts Input File (/SRTI)**

The /SRTI identifier (optional) and its corresponding text string specify the path and filename of the starts input file, which specifies the tab-delimited starts file (output from the OffNetActCalc utility or equivalent format) for each county. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 30 shows the contents of the starts input file header line.

**Table 30. Starts Input File Header Line.**

<b>Header Line Text</b>	<b>Description</b>
County	Hour ID
TabDelimitedFile	Path and Filename of Tab-Delimited File

Following the header line, the user must list the county (identifier, typically FIPS code) and the county's respective tab-delimited starts file. Only one county can appear per line. Figure 71 shows a sample starts input file.



```

* Comments can be placed at any line by putting a * in the first column
* The County can be any text or number
* The County and TabDelimitedFile can be separated by any combination of spaces
*   and/or tabs
County   TabDelimitedFile
48001   C:\OffNetActSummary\Input\OffNetAct\48001_2018_OffNetActCalc_SHP.tab
48003   C:\OffNetActSummary\Input\OffNetAct\48003_2018_OffNetActCalc_SHP.tab

```

**Figure 71. Sample Starts Input File.**

**Hotelling/SHI/APU Hours Input File (/HEAI)**

The /HEAI identifier (optional) and its corresponding text string specify the path and filename of the hotelling/SHI/APU hours input file, which specifies the tab-delimited hotelling/SHI/APU hours file (ouput from the OffNetActCalc utility or equivalent format) for each county. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 31 shows the contents of the starts input file header line.

**Table 31. Hotelling/SHI/APU Hours Input File Header Line.**

Header Line Text	Description
County	Hour ID
TabDelimitedFile	Path and Filename of Tab-Delimited File

Following the header line, the user must list the county (identifier, typically FIPS code) and the county’s respective tab-delimited hotelling/SHI/APU hours file. Only one county can appear per line. Figure 72 shows a sample hotelling/SHI/APU hours input file.

```

* Comments can be placed at any line by putting a * in the first column
* The County can be any text or number
* The County and TabDelimitedFile can be separated by any combination of spaces
*   and/or tabs
County   TabDelimitedFile
48001   C:\OffNetActSummary\Input\OffNetAct\48001_2018_OffNetActCalc_SHP.tab
48003   C:\OffNetActSummary\Input\OffNetAct\48003_2018_OffNetActCalc_SHP.tab

```

**Figure 72. Sample Hotelling/SHI/APU Hours Input File.**

**Header Input File (/HDRO)**

The /HDRO identifier (optional) and its corresponding text string specify the path and filename of the header input file, which specifies the header that will be used by the utility when writing the tab-delimited output file. This input file is optional. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input

file, other than any optional comments, must be the header that the user intends to appear in the tab-delimited output file. Figure 73 shows a sample header input file.

<pre>* The user can add comments by placing the symbol * in the first column * The utility recognizes the first line without a * in the first column as the header. ActivitySummary hgb0608_mvs10a SHP_Starts</pre>
---

**Figure 73. Sample Header Input File.**

### **24-Hour Summary Output File (/TABO)**

The /TABO identifier and its corresponding text string specify the path and filename of the 24-hour summary output file. This tab-delimited output file lists the county, and the 24-hour off-network activity. If more than one off-network activity type (i.e., SHP and starts) is found in the jcf, each off-network activity type will have a separate column of data.

### **Hourly Summary Output File (/HTABO)**

The /HTABO identifier and its corresponding text string specify the path and filename of the hourly summary output file. This tab-delimited output file lists the county and off-network data type activity by hour.

### **24-Hour Vehicle Type Summary Output File (/VTABO)**

The /VTABO identifier and its corresponding text string specify the path and filename of the 24-hour vehicle type summary output file. This tab-delimited output file lists the county and off-network data type activity by vehicle type (i.e., SUT/fuel type).

### **Hourly Vehicle Type Summary Output File (/ATABO)**

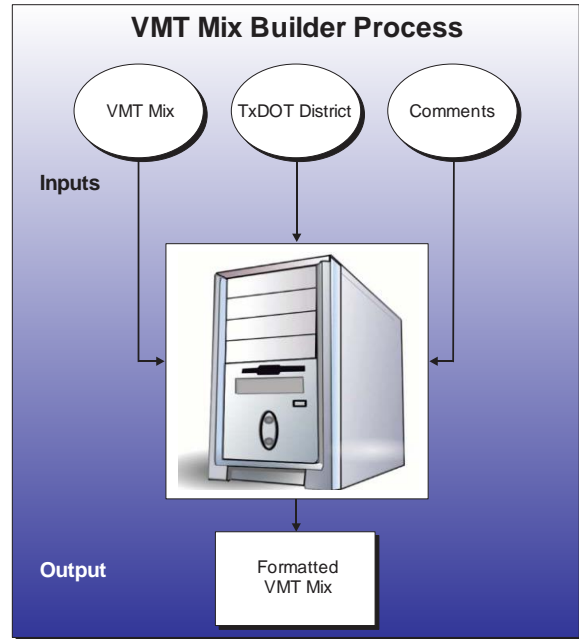
The /ATABO identifier and its corresponding text string specify the path and filename of the hourly vehicle type summary output file. This tab-delimited output file lists the county and off-network data type activity by vehicle type (i.e., SUT/fuel type) and hour.

## VMTMIXBUILD: VMT MIX BUILDER

The VMTmixBuild utility formats either the 24-hour or one of the time periods VMT mix for input to EmsCalc and the other utilities within MOVES2014aUtl. The user develops the VMT mix (from TxDOT districts) externally to this utility for the desired MOVES SUT/fuel type combinations and for each MOVES roadway type (additional roadway types may be included after the MOVES roadway types).

### OPERATION

The VMTmixBuild utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.



If any of the required input files are missing from the jcf, or TRANSVMT cannot open any of these input files due to an incorrect path and/or filename or any other access error, then TRANSVMT results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the TxDOT district name input file. The utility reads each line, checking the data in the first column. If the data in the first column is the asterisk symbol (\*), the utility recognizes the line as a comment and ignores the line. If the data in the first column is not the asterisk symbol (\*), the utility recognizes the line as a data line. For each remaining line in the input file, the utility locates the line with the text *DISTRICT* as the first word and saves the text following the “=” symbol as the district name. In this input file, the text *DISTRICT* is not case sensitive and may appear as any combination of uppercase and lowercase characters.

The next input file the utility processes is the comments input file. The utility reads each line of this input file and saves them in the comments array.

The next input file the utility processes is the VMT mix input file. The utility reads each line until locating the text *Obs* as the first word, which signifies the header line for the first block of data. Beginning with the fourth column of data, the utility uses the last three characters to set the SUT code (set as the first two characters) and the fuel type (set as the last character). The utility then sets the fuel type code based on the fuel type: G is fuel type one, D is fuel type two, C is fuel type three, and E is fuel type nine. For example, if the segment of the header line for the first block of data is the text *P\_PV21D*, then the SUT code is 21, the fuel type is D and the fuel type code is two.

The utility saves the SUT code and fuel type code for each column of data in the code segment of the VMT mix array. The utility then begins the process of reading the first block of data. The utility reads each line and compares the input district name (second column of data) to the desired district name. If these are the same, the utility saves the observation number (first column of data) to the VMT mix array based on the road type code (last character of the third column of data). The utility also saves the VMT mix data (starting with the fourth column of data) to VMT mix array based on the road type code.

The utility repeats this process until reaching either the end of the file or the utility locates the text *Obs* as the first word in the line, which signifies a new block of data. For a new block of data, the utility uses a process similar to the first block of data to read the VMT mix data. The utility sets the SUT code and fuel type code, beginning with the second column of data. The utility then reads each line and compares the input observation number (first column of data) to the observation numbers in the VMT mix array. If the observation numbers are the same, the utility saves the VMT mix data (starting with the second column of data) to the VMT mix array. The utility repeats this process until it finds a new block of data or reaches the end of the file.

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

Using the VMT mix array, the utility checks for any missing base MOVES road types (i.e., sum of VMT mix data across all SUT/fuel type combinations equals 0). This check is performed only for the base MOVES road types (ID 2 through 5). If a missing road type is found, the utility assigns the road type VMT mix data based on a substitute road type. Table 32 shows each road type and associated road type substitution.

**Table 32. VMT Mix Road Type Substitutions**

Road Type	Substitution Option 1 <sup>1</sup>	Substitution Option 2	Substitution Option 3
2 (Rural Restricted Access)	4 (Urban Restricted Access)	3 (Rural Unrestricted Access)	5 (Urban Unrestricted Access)
3	5	2	4
4	2	5	3
5	3	4	2

<sup>1</sup> Substitutions are performed in the order shown until a road type with VMT mix data is located.

The utility also checks the sum of the VMT mix data for each road type across all of the SUT/fuel type combinations. If this sum is not equal to the value of 1, then the utility adjusts the largest VMT mix data value for that road type so that the road type total VMT mix is equal to the value of 1. In the event that a road type total VMT mix does not sum to the value of 1, the utility writes a warning to the listing output file.

The utility then writes the data to the VMT mix output file. The utility writes each comment from the comments array to the VMT mix output file. The utility then writes the header to the VMT mix output file. For each road type in the VMT mix array, the utility writes the SUT code, fuel type code, VMT mix data (from the VMT mix array), the MOVES road type description (from the MOVES base data array), the MOVES SUT description (from the MOVES base data array), and the MOVES fuel type description (from the MOVES base data array) to the VMT mix output file for each SUT/fuel type combination in the VMT mix array.

### JCF INPUT FILE

The jcf input file directs the VMTmixBuild utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string that can either be a comment, the utility to run (VMTmixBuild in this case), or the path and filename of the input or output file. Table 33 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 33. VMTmixBuild File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/VMXI	VMT Mix Input File	Path and Filename
/DISTI	TxDOT District Name Input File	Path and Filename
/CMI	Comments Input File <sup>2</sup>	Path and Filename
/VMXO	VMT Mix Output File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (VMTmixBuild), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 74 shows a sample jcf.

/JOB	VMTmixBuild Bryan District
/STEP	Step1 VMTmixBuild
/LST	C:\VMTmixBuild\Output\BryanDist\VMTmixBuild_BryanDist.lst
/VMXI	C:\VMTmixBuild\Input\SU_08WX.Lst
/DISTI	C:\VMTmixBuild\Input\District_Bryan.txt
/CMI	C:\VMTmixBuild\Input\Comments_BryanDist.txt
/VMXO	C:\VMTmixBuild\Output\BryanDist\VMTmixBuild_BryanDist.txt

**Figure 74. Sample VMTmixBuild jcf.**

### Job Identifier (/JOB)

The /JOB identifier and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### Step Identifier (/STEP)

The /STEP identifier and its corresponding text string name the utility to run. The VMTmixBuild utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (VMTmixBuild in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path, filename, and contents of the TxDOT district name input file;
- The path and filename of the comments input file;
- The path and filename of the VMT mix file;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **VMT Mix Input File (/VMXI)**

The /VMXI identifier and its corresponding text string specify the path and filename of the VMT mix input file. This input file lists the VMT mix by TxDOT district for the desired MOVES SUT/fuel type combinations by roadway type. The four base MOVES roadway types are listed as RT2, RT3, RT4, and RT5. Any additional roadway types must be listed after those roadway types (i.e., RT6). Figure 75 shows a sample of a VMT mix input file.

The SAS System													11:25 Friday, January 8, 2010 1	
Obs	District	RT	P_PV21D	P_PV21G	P_PT31D	P_PT31G	P_LCT32D	P_LCT32G	P_RFT51D	P_RFT51G	P_SUSH52D	P_SUSH52G	P_SULH53D	
1	Abilene	RT2	0.00039	0.38584	0.00448	0.14009	0.00333	0.04487	0.00004	0.00001	0.03760	0.01168	0.00100	
2	Abilene	RT3	0.00046	0.46329	0.00694	0.21698	0.00515	0.06949	0.00006	0.00002	0.05790	0.01798	0.00155	
3	Abilene	RT4	0.00036	0.36368	0.00454	0.14200	0.00337	0.04548	0.00004	0.00001	0.04141	0.01286	0.00111	
4	Amarillo	RT2	0.00052	0.51953	0.00349	0.12132	0.00300	0.03861	0.00003	0.00001	0.02866	0.01324	0.00113	
5	Amarillo	RT3	0.00046	0.46005	0.00567	0.19692	0.00486	0.06267	0.00005	0.00002	0.04466	0.02063	0.00176	
6	Amarillo	RT4	0.00066	0.65812	0.00397	0.13771	0.00340	0.04383	0.00002	0.00001	0.02237	0.01033	0.00088	
7	Atlanta	RT2	0.00048	0.47669	0.00340	0.11391	0.00274	0.03637	0.00003	0.00001	0.03345	0.00938	0.00065	
8	Atlanta	RT3	0.00058	0.57535	0.00477	0.15965	0.00384	0.05097	0.00005	0.00001	0.04502	0.01262	0.00087	
9	Atlanta	RT4	0.00056	0.55885	0.00372	0.12448	0.00299	0.03974	0.00003	0.00001	0.03187	0.00894	0.00062	
10	Atlanta	RT5	0.00071	0.70703	0.00418	0.14003	0.00337	0.04471	0.00003	0.00001	0.02541	0.00713	0.00049	
11	Austin	RT2	0.00067	0.66960	0.00409	0.12785	0.00308	0.04090	0.00003	0.00001	0.02643	0.00997	0.00297	
12	Austin	RT3	0.00063	0.62948	0.00602	0.18805	0.00453	0.06016	0.00004	0.00001	0.03565	0.01345	0.00400	
13	Austin	RT4	0.00072	0.71910	0.00417	0.13034	0.00314	0.04170	0.00003	0.00001	0.02412	0.00910	0.00271	
14	Austin	RT5	0.00083	0.83204	0.00339	0.10601	0.00255	0.03391	0.00001	0.00000	0.00977	0.00369	0.00110	

**Figure 75. Sample VMT Mix Input File.**



### **TxDOT District Name Input File (/DISTI)**

The /DISTI identifier and its corresponding text string specify the path and filename of the TxDOT district name input file that lists the TxDOT district for formatting the VMT mix. The TxDOT district in this input file must appear (exact spelling) as it does in the VMT mix input file. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the data identifier (District), followed by the equals symbol (=) and the appropriate TxDOT district name, separated by any combination of tabs and/or spaces. Figure 76 shows a sample TxDOT district name input file.

```
* NOTE: Comments can be included in jcf by placing the symbol * in the first column.
* NOTE: Format is fixed with identifier text District followed by the symbol = and the
*       district as it appears in the VMT mix input file.
* NOTE: District can be any combination of capital or lowercase letters.
* NOTE: All data can be separated by any combination of tabs and/or spaces.
District = Bryan
```

**Figure 76. Sample TxDOT District Name Input File.**

### **Comments Input File (/CMI)**

The /CMI identifier and its corresponding text string specify the path and filename of the comments input file that lists any comments added to the beginning of the VMT mix output file. This input file is optional and all lines in this input file are comments added to the VMT mix input file. Since the user inserts these lines as comments in the VMT mix output file, each line of this input file must begin with the asterisk symbol (\*). Figure 77 shows a sample comments input file.

```
* NOTE: Bryan TxDOT District
* NOTE: Descriptions (VMX_RDdesc, MOVES_STdesc, MOVES_FTdesc) are not used and are optional.
```

**Figure 77. Sample Comments Input File.**

### **VMT Mix Output File (/VMXO)**

The /VMXO identifier and its corresponding text string specify the path and filename of the VMT mix output file that lists the VMT mix for each MOVES SUT/fuel type combination by roadway type. Prior to the VMT mix data, the utility writes any comments from the comments input file. The utility writes a header line (consisting of the text *VMX\_RDcode*, *MOVES\_STcode*, *MOVES\_FTcode*, *VMTmix*, *VMX\_RDdesc*, *MOVES\_STdesc*, and *MOVES\_FTdesc*) prior to the VMT mix data. Figure 78 shows a sample of a VMT mix output file.

\* NOTE: Bryan TxDOT District

\* NOTE: Descriptions (MOVES\_RDdesc, MOVES\_STdesc, MOVES\_FTdesc) are not used and are optional.

VMX_RDcode	MOVES_STcode	MOVES_FTcode	VMTmix	VMX_RDdesc	MOVES_STdesc	MOVES_FTdesc
2	11	1	0.00049	Rural_Rest	MC	Gas
2	21	1	0.48940	Rural_Rest	PC	Gas
2	21	2	0.00049	Rural_Rest	PC	Diesel
2	31	1	0.12970	Rural_Rest	PT	Gas
2	31	2	0.00415	Rural_Rest	PT	Diesel

**Figure 78. Sample VMT Mix Output File.**

## **EMISSIONS RATES UTILITIES**

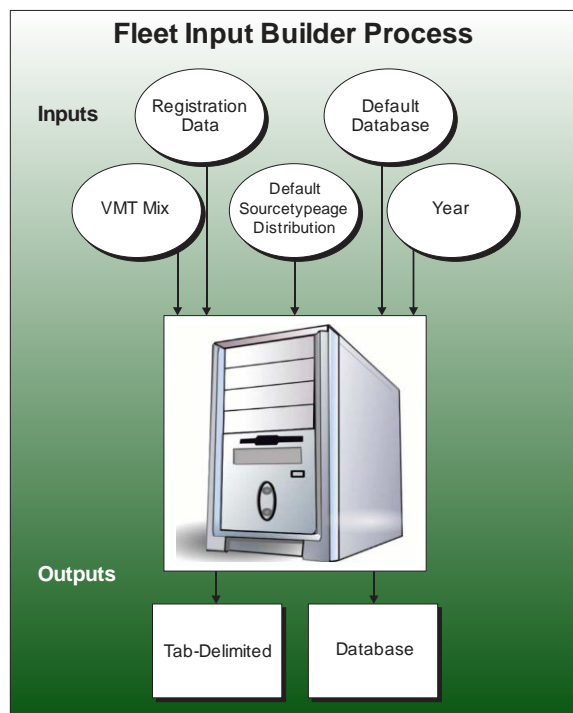


## MOVESFLEETINPUTBUILD: MOVES FLEET INPUT BUILDER

The MOVESfleetInputBuild utility develops the sourcetypeagedistribution and fuelengfraction database tables in a format consistent with MOVES inputs. Tab-delimited formats of these database tables are available for output as well. Vehicle registration data supplied by the TxDMV, the MOVES defaults, the default age distributions, and a VMT mix (only used to determine the SUTs and fuel types) are the basis for these outputs.

### OPERATION

The MOVESfleetInputBuild utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.



If any of the required input files are missing from the jcf, or MOVESfleetInputBuild cannot open any of these input files due to an incorrect path and/or filename or any other access error, then MOVESfleetInputBuild results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file processed by the utility is the sourcetypeagedistribution data sources input file. If the either the database table or tab-delimited format of the sourcetypeagedistribution output is in the jcf, this input file is required. The utility reads each line of this input file and checks the first column of data. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *SUT* and *DATASOURCE* and sets the location of the SUT and the SUT data sources file for the associated SUT, respectively. The utility then reads each remaining line and saves the SUT data sources file in the SUT data sources part of the SUT data sources input array for the associated SUT.

For each SUT, the utility then processes the SUT data sources file from the SUT data sources part of the SUT data sources input array. The utility reads each line of the SUT data sources file and checks the first column of data. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). If the first word in this line is the text *GRAND*, *COUNTY*, or *MOVES* (capitalization does not matter), then the utility recognizes this line as the data line and sets the temporary county name as the data line.

The utility saves this temporary county name in the county name part of the SUT data sources input array. The utility repeats this process for each line in the SUT data sources file (i.e., the user can list multiple counties for a SUT, in which the registration data is aggregated) and for each SUT. After processing each SUT, the utility then creates a list of the unique counties (including grand totals and MOVES default, if input) in the county name part of the SUT data sources input array and saves this list in the county calculation array.

The next input file the utility processes is the fuelengfraction data sources input file. If either the database table or tab-delimited format of the fuelengfraction output is in the jcf, this input file is required. The utility reads each line of this input file and checks the first column of data. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *SUT* and *DATASOURCE* and sets the location of the SUT and the SUT data sources file for the associated SUT, respectively. The utility then reads each remaining line and saves the SUT data sources file in the SUT data sources part of the fuel engine fractions SUT data sources input array for the associated SUT.

For each SUT, the utility then processes the SUT data sources file from the SUT data sources part of the fuel engine fractions SUT data sources input array. The utility reads each line of the SUT data sources file and checks the first column of data. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). If the first word in this line is the text *GRAND*, *COUNTY*, or *MOVES* (capitalization does not matter), then the utility recognizes this line as the data line and sets the temporary county name as the data line. The utility saves this temporary county name in the county name part of the fuel engine fractions SUT data sources input array.

This utility repeats this process for each line in the SUT data sources file (i.e., the user can list multiple counties for a SUT, in which the registration data is aggregated) and for each SUT. After processing each SUT, the utility then checks the county calculation array for each unique county in the county name part of the fuel engine fractions SUT data sources input array. If the utility does not locate the county in the county calculation array, then the utility adds the county to the county calculation array.

The utility then processes the default fuel engine fractions input file. If either the database table or tab-delimited format of the fuelengfraction output is in the jcf, this input file is required. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *sourceTypeModelYearID*, *fuelTypeID*, *engTechID*, and *fuelEngFraction* and sets the

location of the source type/model year ID, fuel type ID, engine technology ID, and the fuel engine fraction, respectively. The utility then reads each remaining line and saves the fuel engine fraction in the MOVES fuel engine fractions array by SUT (first two digits of source type/model year ID data), year (last four digits of source type/model year ID data), fuel type, and engine technology.

The utility then begins the county-level calculation process. For each county in the county calculation array where the county is not MOVES DEFAULT (capitalization does not matter), the utility sets the temporary county name as the county name from the county calculation array. The utility then begins the process of reading the age registrations, diesel heavy-duty registrations, and gas heavy-duty registrations input files. For the age registrations input file, the utility reads each line until locating the text *COUNTY* or *GRAND* as the first word in the line. After locating this text, the utility then compares the registrations county name to the temporary county name from the county calculation array. If these county names do not match, then the utility repeats this process until locating the desired county name.

After locating the desired county name, the utility reads the each line until locating the text *MODEL* as the first word in the line, which signifies the beginning of the data, and the utility skips the next line (part of the data header). The utility then reads the next 30 lines and saves the registration data for data columns 2 through 6 (Passenger, Motorcycles, Trucks <=6000, Trucks >6000 <=8500, Total Trucks <=8500 age registrations vehicle categories) in the age registration data array (by vehicle type and year). The utility also saves the first data column of the first line as the temporary output year.

The utility then reads the next line, which is the *OLDER* year model in the age registrations input file, and processes this input line in a similar manner as the first 30 data lines, except that the utility saves the registration data in year 31 of the age registration data array. The utility then processes the next line, which is the *TOTAL* year model in the age registrations input file, in the same manner as the *OLDER* year model except that the utility saves the registration data in year 32 of the age registration data array.

For the diesel heavy-duty registrations input file, the utility reads each line until locating the text *COUNTY* or *GRAND* as the first word in the line. After locating this text, the utility then compares the registrations county name to the temporary county name from the county calculation array. If these county names do not match, then the utility repeats this process until locating the desired county name.

After locating the desired county name, the utility reads the each line until locating the text *MODEL* as the first word in the line, which signifies the beginning of the data, and the utility skips the next line (part of the data header). The utility then reads the next 30 lines and saves the registration data for data columns 2 through 9 (Diesel > 8500, Diesel > 10000, Diesel > 14000, Diesel > 16000, Diesel > 19500, Diesel > 26000, Diesel > 33000, Diesel > 60000 diesel heavy-duty registrations vehicle categories) in the diesel portion of the diesel/gas registration data array (by diesel/gas, vehicle type and year).

The utility then reads the next line, which is the OLDER year model in the diesel heavy-duty registrations input file, and processes this input line in a similar manner as the first 30 data lines, except that the utility saves the registration data in year 31 of the diesel portion of the diesel/gas registration data array. The utility processes the next line, which is the TOTAL year model in the age registrations input file, in the same manner as the OLDER year model except that the utility saves the registration data in year 32 of the diesel portion of the diesel/gas registration data array.

For the gas heavy-duty registrations input file, the utility reads each line until locating the text *COUNTY* or *GRAND* as the first word in the line. After locating this text, the utility then compares the registrations county name to the temporary county name from the county calculation array. If these county names do not match, then the utility repeats this process until locating the desired county name.

After locating the desired county name, the utility reads the each line until locating the text *MODEL* as the first word in the line, which signifies the beginning of the data, and the utility skips the next line (part of the data header). The utility then reads the next 30 lines and saves the registration data for data columns 2 through 9 (Gas > 8500, Gas > 10000, Gas > 14000, Gas > 16000, Gas > 19500, Gas > 26000, Gas > 33000, Gas > 60000 gas heavy-duty registrations vehicle categories) in the gas portion of the diesel/gas registration data array (by diesel/gas, vehicle type and year).

The utility then reads the next line, which is the OLDER year model in the gas heavy-duty registrations input file, and processes this input line in a similar manner as the first 30 data lines, except that the utility saves the registration data in year 31 of the gas portion of the diesel/gas registration data array. The utility processes the next line, which is the TOTAL year model in the age registrations input file, in the same manner as the OLDER year model except that the utility saves the registration data in year 32 of the gas portion of the diesel/gas registration data array.

The utility then creates the registration data category array (by registration data category and year, including total). The utility combines the age registration data array and the diesel/gas registration data array to form the registration category data array using the combinations in Table 34.



**Table 34. Registration Categories.**

Registration Category	Vehicle Category	Data Location
1	Passenger	Age registration data array
2	Motorcycles	
3	Total Trucks <=8500	
4	Diesel > 8500, Diesel > 10000, Diesel > 14000, Diesel > 16000	Diesel/gas registration data array
5	Diesel > 19500, Diesel > 26000, Diesel > 33000, Diesel > 60000	
6	Gas > 8500, Gas > 10000, Gas > 14000, Gas > 16000	
7	Gas > 19500, Gas > 26000, Gas > 33000, Gas > 60000	

The utility then begins the process of adding the registration data (from the registration category data array) to the SUT age distribution array (by SUT and year). For each SUT, the utility checks the county name part of the SUT data sources input array for the temporary county name (from the county calculation array). If the utility does not locate the temporary county name for the SUT, then the utility skips this SUT. Otherwise, the utility adds the registration data to the SUT age distribution array for the SUT. The utility assigns each SUT the registrations from one or more of the registration categories in the registration category data array. Table 35 shows the SUTs and their associated registration category in the registration category data array.

**Table 35. SUT/Registration Category Correlation For SUT/Age Distribution Array.**

SUT	Registration Category
11	2
21	1
31, 32	3
52, 53	4 + 6
61, 62	5 + 7

The utility then begins the process of adding the registration data (from the registration category data array) to the SUT fuel engine fractions array (by SUT, year, and diesel/gas). For SUTs 52 and 53, the utility adds the registrations from registration category 4 to the diesel part of the SUT fuel engine fractions array and adds the registrations from registration category 6 to the gas part of the SUT fuel engine fractions array. For SUT 61, the utility adds the registrations from registration category 5 to the diesel part of the SUT fuel engine fractions array and adds the registrations from registration category 7 to the gas part of the SUT fuel engine fractions array.

For SUT 62, the utility adds the registration data from registration categories 5 and 7 to the diesel part of the SUT fuel engine fractions array and the gas part of the SUT fuel engine fractions array is set to 0 (i.e., all vehicles in SUT 62 are assumed to be diesel). This ends the county level calculation process, which the utility repeats for each county in the county calculation array.

The utility then processes the default *sourcetypeagedistribution* input file. If either the database table or tab-delimited format of the *sourcetypeagedistribution* output is in the jcf, this input file is required. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *YEARID*, *SOURCETYPEID*, *AGEID*, or *AGEFRACTION* (capitalization does not matter) and sets the location of the year ID, SUT ID, age ID, and age fraction data, respectively. The utility then reads each remaining line and compares the year from the input line to the temporary output year (from the age registrations input file). If these years are not the same, then the utility skips this input line. Otherwise, the utility saves the age fractions data to the MOVES default age distribution array (by SUT and age) based on the SUT ID and age ID data.

The next input file the utility processes is the VMT mix input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMX\_RDCODE*, *MOVES\_STCODE*, *MOVES\_FTCODE*, and *VMTMIX* and sets the location of the VMT mix road type code, MOVES SUT code, MOVES fuel type code, and VMT mix data, respectively. The utility then reads each remaining line and if the VMT mix value is greater than 0 for a SUT/fuel type, then the utility sets the SUT/fuel type array to *YES* for that SUT/fuel type (i.e., the SUT/fuel type is desired in the output). If the VMT mix value is equal to 0, then the utility sets the SUT/fuel type array to *NO* for that SUT/fuel type.

If the database format of the *sourcetypeagedistribution* output or *fuelengfraction* output is found in the jcf, the utility then processes the MySQL connection inputs file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

Using MySQL, the utility then checks that the user-supplied MOVES default database does exist. If this database does not exist, the utility results in an error and the current step execution is terminated.

If the text for either the database table or tab-delimited format of the *sourcetypeagedistribution* output is in the jcf, the utility then converts the registration data in the SUT age distribution array to distributions. For each SUT, the utility checks the county name part of the SUT data sources input array for the text *MOVES DEFAULT* (capitalization does not matter). If the utility locates the text *MOVES DEFAULT*, then the utility copies the distribution

from the MOVES default age distribution array to the SUT age distribution array for each of the 31 years. Otherwise, the utility replaces the yearly registration values in the SUT age distribution array with the yearly registration values divided by the total SUT registrations. The utility repeats this process for each of the 31 years in the SUT age distribution array and for each SUT.

If the text for either the database table or tab-delimited format of the fuelengfraction output is in the jcf, the utility then begins the fuel engine fractions calculation process. The first step of this process is to adjust the MOVES fuel engine fractions array to reflect diesel and gas vehicles only. For each SUT and year, the utility calculates the total of the fuel engine fractions for all fuel types other than diesel and gas (across all engine technologies). If this sum is equal to 0, then no adjustment is necessary and this SUT and year is skipped. Otherwise, the utility makes the adjustment for each engine technology by calculating the sum of the fuel engine fractions for the diesel and gas fuel types, replacing the diesel fractions with the ratio of the diesel fractions and the sum of the diesel and gas fuel types, replacing the gas fractions with the ratio of the gas fractions and the sum of the diesel and gas fuel types, and setting all other fuel types to 0.

The utility then calculates the fuel engine fractions for all of the SUTs in the SUT fuel engine fractions array by dividing the yearly diesel registrations by the sum of the yearly diesel and gas registrations and saves these fractions in the diesel part of the SUT fuel engine fractions array. The utility then adjusts the SUT fuel engine fractions array to reflect the SUT/fuel type array. For each SUT in the SUT fuel engine fractions array, the utility checks each fuel type in the SUT/fuel type array for that SUT. If the text "NO" is found in the SUT/fuel type array for any of the fuel types, then the utility adjusts the SUT fuel engine fractions array so that the fuel engine fractions with the *YES* designation in the SUT/fuel type array have a value of 1 for all years and the fuel engine fractions with the *NO* designation in the SUT/fuel type array have a value of 0 for all years. Then, for the SUTs in the SUT fuel engine fractions array, the utility replaces the fuel engine fractions in the MOVES fuel engine fractions array with the appropriate fuel engine fractions from the SUT fuel engine fractions array.

For years in the MOVES fuel engine fractions array that are greater than the temporary output year, the utility uses the fuel engine fractions from the first year in the SUT fuel engine fractions array. The utility replaces the next 31 years (beginning with the temporary output year) with the values from the SUT fuel engine fractions array. The utility replaces the remaining years in the MOVES fuel engine fractions array with the last year of fuel engine fractions in the SUT fuel engine fractions array.

The utility then processes the year ID input file. The utility reads each line until locating the text *YEAR* (capitalization does not matter) as the first word in the line. After locating this text in the first segment, the utility then sets the year output as segment three.

The utility then writes the sourcetypeagedistribution database table and tab-delimited output file. The utility opens the output file and writes the header line, which consists of the text *sourceTypeID*, *yearID*, *ageID*, and *ageFraction*, with each set of text separated by a tab. For each SUT, the utility writes the SUT ID, the year output (either the temporary output year from the registration data or the year from the year ID input file), the age ID, and the age fractions (from the SUT age distribution array ) for each of the 31 years to the output file. If the text for

the *sourcetypeagedistribution* database table output is in the jcf, then the utility uses the appropriate MySQL scripts to create a database table similar to the MOVES default *sourcetypeagedistribution* database table. Using MySQL scripts, the utility adds the data to the database table in the same manner as writing the tab-delimited output.

The utility then writes the *fuelengfraction* database table and tab-delimited output file. The utility opens the output file and writes the header line, which consists of the text *sourceTypeID*, *modelYearID*, *fuelTypeID*, *engTechID*, and *fuelEngFraction*, with each set of text separated by a tab. For each SUT, model year, fuel type, and engine technology with values in the MOVES fuel engine fractions array, the utility writes the SUT ID, the model year ID, the fuel type ID, engine technology ID, and fuel engine fractions (from the MOVES fuel engine fractions array) to this output file. If the text for the *fuelengfraction* database table output is in the jcf, then the utility uses the appropriate MySQL scripts to create a database table that contains the same data as the tab-delimited output. Using MySQL scripts, the utility adds the data to the database table in the same manner as writing the tab-delimited output.

## JCF INPUT FILE

The jcf input file directs the MOVESfleetInputBuild utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (MOVESfleetInputBuild in this case), or the path and filename of the input or output file. Table 36 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 36. MOVESfleetInputBuild File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/STAI	sourcetypeagedistribution Data Sources Input File	Path and Filename
/FEFI	fuelengfraction Data Sources Input File	Path and Filename
/AGEI	Age Registrations Input File	Path and Filename
/DSLJ	Diesel Heavy-Duty Registrations Input File	Path and Filename
/GASI	Gasoline Heavy-Duty Registrations Input File	Path and Filename
/MVSDB	MOVES Default Database	Database Name
/DFEFI	Default Fuel Engine Fractions Input File	Path and Filename
/DFSTA	Default sourcetypeagedistribution Input File	Path and Filename
/YRID	Year ID Input File <sup>2</sup>	Path and Filename
/VMX	VMT Mix Input File	Path and Filename
/MYSQL	MySQL Connection Inputs File	Path and Filename
/DSTAO	sourcetypeagedistribution Output – Database Table	Database Name
/TSTAO	sourcetypeagedistribution Output – Tab-Delimited File	Path and Filename
/DFEFO	fuelengfraction Output – Database Table	Database Name
/TFEFO	fuelengfraction Output – Tab-Delimited File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Only if the output year is a year other than the registration data year (i.e., future year analysis).

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (MOVESfleetInputBuild), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 79 shows a sample jcf.

```

/JOB      MOVESfleetInputBuild Harris
/STEP     Step1 MOVESfleetInputBuild
/LST      C:\MOVESfleetInputBuild\Output\Harris\MOVESfleetInputBuild_Harris.lst
/STAI     C:\MOVESfleetInputBuild\Input\SUT_DataSources.txt
/FEFI     C:\MOVESfleetInputBuild\Input\SUT_FEF_DataSources.txt
/AGEI     C:\MOVESfleetInputBuild\Input\RegData2009\31 Years\AGEPERKN.txt
/DSL1     C:\MOVESfleetInputBuild\Input\RegData2009\31 Years\DSLPERKN.txt
/GASI     C:\MOVESfleetInputBuild\Input\RegData2009\31 Years\GASPERKN.txt
/MVSDB    MOVESDB20091221
/DFEFI    C:\MOVESfleetInputBuild\Input\movesdb20141021_fuelengfraction.tab
/DFSTA    C:\MOVESfleetInputBuild\Input\MOVES2010DefaultAgeDistributions20100209.tab
/YRID     C:\MOVESfleetInputBuild\Input\yearID_2018.txt
/VMX      C:\MOVESfleetInputBuild\Input\MOVESvmtMixBuild_HoustonDist.txt
/MYSQL    C:\MOVESfleetInputBuild\Input\MySQL_ConnInputs_MOVES2014.txt
/TSTAO    C:\MOVESfleetInputBuild\Output\Harris\MOVESfleetInputBuild_Harris_sourcetypeagedist.tab
/DSTAO    MOVESfleetInputBuild_Harris
/TFEFO    C:\MOVESfleetInputBuild\Output\Harris\MOVESfleetInputBuild_Harris_fuelengfraction.tab

```

**Figure 79. Sample MOVESfleetInputBuild jcf.**

**Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

**Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The MOVESfleetInputBuild utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

**Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (MOVESfleetInputBuild in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;

- The path, file name, and contents of the sourcetypeagedistribution data sources input file;
- The path, file name, and contents of the fuelengfraction data sources input file;
- The name of the MOVES default database;
- The path and filename of the age registrations input file;
- The path and filename of the diesel heavy-duty registrations input file;
- The path and filename of the gas heavy-duty registrations input file;
- The path and filename of the default sourcetypeagedistribution input file;
- The path, filename, and contents of the year ID input file (if input);
- The path and filename or database name of all output files;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

#### **sourcetypeagedistribution Data Sources Input File (/STAI)**

The /STAI identifier (optional, only used if the sourcetypeagedistribution output is to be created) and its corresponding text string specify the path and filename of the sourcetypeagedistribution data sources input file, which lists each SUT (by MOVES SUT code) and a path and file name of a text file that lists the data sources for the respective SUT. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first data line, other than comments, must be the header line, which consists of the text *SUT* and *DataSource*, separated by any combination of tabs and/or spaces. Following the header line, the user must list the MOVES SUT code, followed by the path and file name to the SUT data sources file (separated by any combination of tabs and/or spaces) for each SUT included in the sourcetypeagedistribution output. Figure 80 shows a sample sourcetypeagedistribution data sources input file.



```

* NOTE: Comments can be included in jcf by placing the symbol * in the first column.
* NOTE: User must list each SUT with the corresponding data source path and file name.
* NOTE: First line must be header line. Header line is not case sensitive but must spell each header
* line as shown.
SUT    DataSource
11     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_HarrisCounty.txt
21     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_HarrisCounty.txt
31     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_HarrisCounty.txt
32     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_HarrisCounty.txt
52     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_HGB_DataSource.txt
53     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_State_DataSource.txt
51     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_MOVESdefault_DataSource.txt
54     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_MOVESdefault_DataSource.txt
41     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_MOVESdefault_DataSource.txt
42     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_MOVESdefault_DataSource.txt
43     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_MOVESdefault_DataSource.txt
61     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_HGB_DataSource.txt
62     C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_State_DataSource.txt

```

**Figure 80. Sample sourcetypeagedistribution Data Sources Input File.**

The SUT data sources file lists the data sources for the specific SUT and has a set format as well. This file can specify that the SUT uses either the MOVES default SUT age distribution, uses the registration data for the entire state to build the SUT age distribution, or uses the county level (can be more than one county) registration data. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. If the population is based on a county, the input line consists of the identifier *COUNTY*, followed by the county name as it appears in the registration data input files (including the county number), and separated by any combination of tabs and/or spaces. If the user wishes to use more than one county, then the user must list each county. If the SUT is to use the registration data for the entire state, use the text *GRAND TOTALS* instead of the identifier and county name. If the SUT is to use the MOVES default SUT age distribution, use the text *MOVES Default* instead of the identifier and county name. Figure 81 shows a sample SUT data source file.



```

* NOTE: Comments can be included in jcf by placing the symbol * in the first column.
* NOTE: Must appear as in registration data, including county number.
* NOTE: If using state level aggregation, then use GRAND TOTALS.
* NOTE: If using region level aggregation, user must list each county.
* NOTE: If using MOVES defaults, use the text MOVES Defaults for county name.
* GRAND TOTALS
* MOVES Default
COUNTY 101 HARRIS
COUNTY 20 BRAZORIA
COUNTY 36 CHAMBERS
COUNTY 84 GALVESTON
COUNTY 79 FORT BEND
COUNTY 237 WALLER
COUNTY 170 MONTGOMERY
COUNTY 146 LIBERTY

```

**Figure 81. Sample SUT Data Source File.**

### **fuelengfraction Data Sources Input File (/FEFI)**

The /FEFI identifier (optional, only used if the fuelengfraction output is to be created) and its corresponding text string specify the path and filename of the fuelengfraction data sources input file, which lists SUT 52, 53, and 61 and a path and file name of a text file that lists the data sources for the respective SUT. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first data line, other than comments, must be the header line, which consists of the text *SUT* and *DataSource*, separated by any combination of tabs and/or spaces. Following the header line, the user must list the MOVES SUT code, followed by the path and file name of the SUT data sources file (separated by any combination of tabs and/or spaces) for each SUT. Figure 82 shows a sample fuelengfraction data sources input file.

```

* NOTE: Comments can be included in jcf by placing the symbol * in the first column.
* NOTE: User must list each SUT with the corresponding data source path and file name.
* NOTE: First line must be header line. The header line is not case sensitive but the user must spell the
* header line as shown.
* NOTE: Currently SUT 11, 21, 31, 32, 51, 54, 41, 42, and 43 are not valid (data does not exist to
* perform calculation).
* NOTE: Currently SUT 62 considered all diesel.
SUT  DataSource
52  C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_State_DataSource.txt
53  C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_State_DataSource.txt
61  C:\MOVESfleetInputBuild\Input\MOVESfleetInputBuild_State_DataSource.txt

```

**Figure 82. Sample fuelengfraction Data Sources Input File.**

The SUT data sources file lists the data sources for the specific SUT and has a set format as well. This file can specify that the SUT use either the MOVES default SUT age distribution data, the registration data for the entire state to build the SUT age distribution data, or the county-level (can be more than one county) registration data. The format of the SUT data sources files used with the fuelengfraction data sources input file is the same as used with the sourcetypeagedistribution data sources input file (see Figure 80).

### Age Registrations Input File (/AGEI)

The /AGEI identifier (required) and its corresponding text string specify the path and filename of the age registrations input file. This input file is a text file and consists of vehicle registrations by model year for each county in Texas and a statewide total. For each block of data (either an individual county or statewide total), 32 lines of registration data (30 labeled model years, an OLDER model year, and a total) must be present, with each model year appearing on one line. The registration data vehicle types included in this input file are PASSENGER, MOTORCYCLES, TRUCKS <=6000, TRUCKS >6000 <=8500, TOTAL TRUCKS <=8500, GAS TRUCKS >8500 (not used), DIESEL TRUCKS >8500 (not used), TOTAL TRUCKS >8500 (not used), and TOTAL ALL TRUCKS (not used). Figure 83 shows a sample of an age registrations input file.

PROGRAM: DSLPERKD		TEXAS DEPARTMENT OF TRANSPORTATION							RUN DATE 11/13/09	
WRITTEN BY: JOEY OESTRICK		DIESEL TRUCK COUNTS BY COUNTY USING GROSS WEIGHT							PAGE 1	
COUNTY	1 ANDERSON									
YEAR	DIESEL	DIESEL	DIESEL	DIESEL	DIESEL	DIESEL	DIESEL	DIESEL	DIESEL	DIESEL
MODEL	> 8500	> 10000	> 14000	> 16000	> 19500	> 26000	> 33000	> 60000	TOTALS	
2009	25	4	0	0	0	0	1	0	30	
2008	134	18	2	3	2	1	1	0	161	
2007	83	21	1	4	1	1	4	0	115	
.										
.										
1980	0	0	0	0	0	0	0	0	0	
OLDER	0	0	0	0	0	1	0	0	1	
TOTAL	661	105	18	25	21	20	48	1	899	

Figure 83. Sample Age Registrations Data Input File.

### Diesel Heavy-Duty Registrations Input File (/DSLH)

The /DSLH identifier (required) and its corresponding text string specify the path and filename of the diesel heavy-duty registrations input file. This input file is a text file and consists of diesel heavy-duty vehicle registrations by model year for each county in Texas and a statewide total. For each block of data (either an individual county or statewide total), the user must include 32 lines of registration data (30 labeled model years, an OLDER model year, and a total), with each model year appearing on one line. The registration data vehicle types included in this input file are DIESEL > 8500, DIESEL > 10000, DIESEL > 14000, DIESEL > 16000, DIESEL > 19500, DIESEL > 26000, DIESEL > 33000, DIESEL > 60000, and DIESEL TOTALS (not used). Figure 84 shows a sample of a diesel heavy-duty registrations input file.

PROGRAM: AGEPERKD WRITTEN BY: JOEY OESTRICK			TEXAS DEPARTMENT OF TRANSPORTATION AGE COUNTS BY COUNTY USING GROSS WEIGHT					RUN DATE 11/13/09 PAGE 1	
COUNTY 1 ANDERSON									
YEAR MODEL	PASSENGER	MOTOR- CYCLES	TRUCKS <=6000	TRUCKS >6000 <=8500	TOTAL TRUCKS <=8500	GAS TRUCKS >8500	DIESEL TRUCKS >8500	TOTAL TRUCKS >8500	TOTAL ALL TRUCKS
2009	457	33	97	145	242	5	30	35	277
2008	1,187	71	395	556	951	24	161	185	1,136
2007	1,413	107	485	600	1,085	14	115	129	1,214
.									
.									
1980	26	5	29	2	31	2	0	2	33
OLDER	243	18	248	23	271	12	1	13	284
TOTAL	18,640	952	9,774	4,338	14,112	212	899	1,111	15,223

**Figure 84. Sample Diesel Heavy-Duty Registrations Data Input File.**

### Gas Heavy-Duty Registrations Input File (/GASI)

The /GASI identifier (required) and its corresponding text string specify the path and filename of the gas heavy-duty registrations input file. This input file is a text file and consists of gas heavy-duty vehicle registrations by model year for each county in Texas and a statewide total. For each block of data (either an individual county or statewide total), the user must include 32 lines of registration data (30 labeled model years, an OLDER model year, and a total), with each model year appearing on one line. The registration data vehicle types included in this input file are GAS > 8500, GAS > 10000, GAS > 14000, GAS > 16000, GAS > 19500, GAS > 26000, GAS > 33000, GAS > 60000, and GAS TOTALS (not used). Figure 85 shows a sample of a gas heavy-duty registrations input file.

PROGRAM: GASPERKD WRITTEN BY: JOEY OESTRICK			TEXAS DEPARTMENT OF TRANSPORTATION GAS TRUCK COUNTS BY COUNTY USING GROSS WEIGHT							RUN DATE 11/13/09 PAGE 1
COUNTY 1 ANDERSON										
YEAR MODEL	GAS > 8500	GAS > 10000	GAS > 14000	GAS > 16000	GAS > 19500	GAS > 26000	GAS > 33000	GAS > 60000	GAS TOTALS	
2009	1	3	1	0	0	0	0	0	5	
2008	18	5	0	0	0	1	0	0	24	
2007	9	4	0	0	0	1	0	0	14	
.										
.										
1980	2	0	0	0	0	0	0	0	2	
OLDER	3	3	1	1	3	0	1	0	12	
TOTAL	128	43	6	6	14	9	6	0	212	

**Figure 85. Sample of an Gas Heavy-Duty Registrations Data Input File.**

### MOVES Default Database (/MVSDB)

The /MVSDB (required) identifier and its corresponding text string specify the name of the database containing the MOVES default database tables. This database must be in the MySQL data folder. Only the name of the database is required.

### Default Fuel Engine Fractions Input File (/DFEFI)

The /DFEFI identifier (optional, only used if the fuelengfraction output is to be created) and its corresponding text string specify the path and filename of the default fuel engine fractions input file. This input file is a text file and consists of the fuel engine fractions by source type/model year, fuel type, engine and technology. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column, which signals the utility to ignore the line. The first line of this input file is the header line, which consists of the text *sourceTypeModelYearID*, *fuelTypeID*, *engTechID*, and *fuelengfraction* separated by any combination of tabs/and or spaces. The capitalization of the text in the header line does not matter. The lines following the header line should be the fuel engine fractions data, with each data column separated by any combination of tabs and/or spaces. Figure 86 shows a sample default fuel engine fractions input file.

sourceTypeModelYearID	fuelTypeID	engTechID	fuelengfraction
111960	1	1	1
111961	1	1	1
111962	1	1	1
111963	1	1	1
...	...	...	...

**Figure 86. Sample Default Fuel Engine Fractions Input File.**

### Default sourcetypeagedistribution Input File (/DFSTA)

The /DFSTA identifier (optional, only used if the sourcetypeagedistribution output is to be created) and its corresponding text string specify the path and filename of the default sourcetypeagedistribution input file. This input file is a text file and consists of the age fractions data for various years by SUT ID and age ID, which the utility uses as the MOVES default sourcetypeagedistribution data. The first line of this input file is the header line, which consists of the text *yearid*, *sourcetypeid*, *ageid*, and *ageFraction* separated by any combination of tabs/and or spaces. The capitalization of the text in the header line does not matter. The lines following the header line should be the age fractions data, with each data column separated by any combination of tabs and/or spaces.

For a given year and SUT, age fractions data for all 31 years (age ID 0 through 30) must appear in this input file (each age ID on a separate line) for the utility to operate properly. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column, which signals the utility to ignore the line. Figure 87 shows a sample of a default sourcetypeagedistribution input file.

yearid	sourcetypeid	ageid	ageFraction
1999	11	0	0.1118
1999	11	1	0.0993
1999	11	2	0.0950
1999	11	3	0.0833
1999	11	4	0.0627
1999	11	5	0.0722
1999	11	6	0.0588
1999	11	7	0.0492
1999	11	8	0.0390
1999	11	9	0.0316

**Figure 87. Sample of a Default sourcetypeagedistribution Input File.**

### Year ID Input File (/YRID)

The /YRID identifier (optional, only used if the output year is different from the registration data year) and its corresponding text string specify the path and filename of the year ID input file, which specifies the year for performing the calculations. Use this input file in the jcf only if replacing the year from the registration data (i.e., future year analysis). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only input line, other than the optional comments, consists of the text *yearID* = followed by the year, separated by any combination of tabs and/or spaces. Figure 88 shows a sample year ID input file.

* Comments can be added by placing the symbol * in first column.
* Format of input file is fixed, with yearID = followed by MOVES yearID code, separated by any
* combination of tabs and/or spaces.
* yearID text can be any combination of capital or lowercase letters.
yearID = 2005

**Figure 88. Sample Year ID Input File.**

### VMT Mix Input File (/VMX)

The /VMX identifier (required) and its corresponding text string specify the path and filename of the VMT mix input file that specifies the mix of vehicles by SUT/fuel type combination for each of the on-network MOVES roadway types (see Appendix A) and any additional road types specified by the user. Since the VMT mix input file is only used to determine the required SUT/fuel type combinations, either the 24-hour or one of the time period VMT mixes can be used. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 37 shows the contents of the VMT mix input file header line.

**Table 37. VMT Mix Input File Header Line.**

Header Line Text	Description
VMX_RDcode	VMT mix road type code
MOVES_STcode	MOVES SUT code
MOVES_FTcode	MOVES fuel type code
VMTmix	VMT mix value
VMX_RDdesc	VMT mix road type code description <sup>1</sup>
MOVES_STdesc	MOVES SUT code description <sup>1</sup>
MOVES_FTdesc	MOVES fuel type code description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes and VMT mix data.

Following the header line, the user must list each MOVES SUT/MOVES fuel type combination for each road type (each on a separate line) with the combinations' associated VMT mix value, separated by any combination of tabs and/or spaces. The utility will calculate values only for those MOVES SUT/MOVES fuel type combinations in this input file. Figure 89 shows a sample VMT mix input file.

* NOTE: Waco TxDOT District – 2000 Friday						
* NOTE: Descriptions (VMX_RDdesc, MOVES_STdesc, MOVES_FTdesc) are not used and are optional.						
VMX_RDcode	MOVES_STcode	MOVES_FTcode	VMTmix	VMX_RDdesc	MOVES_STdesc	MOVES_FTdesc
2	11	1	0.00055	Rural_Rest	MC	Gas
2	21	1	0.56562	Rural_Rest	PC	Gas
2	21	2	0.00170	Rural_Rest	PC	Diesel

**Figure 89. Sample VMT Mix Input File.**

### MySQL Connection Inputs File (/MYSQL)

The /MYSQL identifier (required) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 90 shows a sample MySQL connections input file.

```

* Comments can be added by placing * in first column.
* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by
* any combination of tabs and/or spaces.
* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any
* combination of tabs and/or spaces.
* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any
* combination of tabs and/or spaces.
* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.
Port Number = 3307
Username = moves
Password = moves

```

**Figure 90. Sample MySQL Connections Input File.**

### **sourcetypegedistribution Database Table Output (/DSTAO)**

The /DSTAO identifier and its corresponding text string specify the name of the database where the sourcetypegedistribution database table is output. This output is optional and, if created, the database will be in the MySQL data folder. Only the name of the database is required and the format of this database table is identical to the MOVES default sourcetypegedistribution database table.

### **sourcetypegedistribution Tab-Delimited Output (/TSTAO)**

The /TSTAO identifier and its corresponding text string specify the path and filename of the tab-delimited version of the sourcetypegedistribution database table output. This output file is optional and contains the same data as the sourcetypegedistribution database table output, except in tab-delimited form.

### **fuelengfraction Database Table Output (/DFEFO)**

The /DFEFO identifier and its corresponding text string specify the name of the database where the fuelengfraction database table is output. This output is optional and, if created, the database will be in the MySQL data folder. Only the name of the database is required and the format of this database table is identical to the MOVES default fuelengfraction database table.

### **fuelengfraction Tab-Delimited Output (/TFEFO)**

The /TFEFO identifier and its corresponding text string specify the path and filename of the tab-delimited version of the fuelengfraction database table output. This output file is optional and contains the same data as the fuelengfraction database table output, except in tab-delimited form.

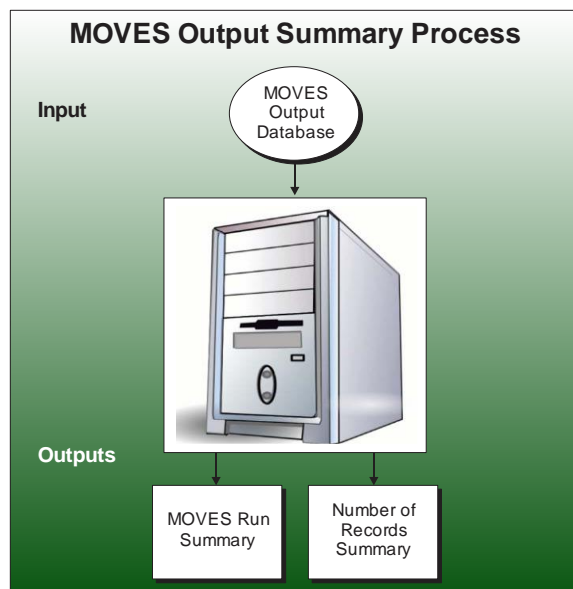




## MOVESOUTSUMMARY: MOVES OUTPUT SUMMARY

The MOVESoutSummary utility creates summary report files for one or more user-specified MOVES runs. The utility has the option of creating a MOVES run summary tab-delimited output file, which lists pertinent information regarding the user-specified MOVES runs including the database name, run spec file name, run spec file date and time, the MOVES run date and time, the duration of the MOVES run, the default database used, the master version date, the master computer ID, the domain county name, and the domain database name.

The utility also has the option of creating the pollutant/process/SUT tab-delimited output file, which lists the number of “null” records, number of not “null” records, the total number of records, minimum value, and maximum value by database table (rateperdistance, rateperprofile, ratepervehicle, rateperstart, and rateperhour for a MOVES rates mode run or movesoutput for a MOVES inventory mode run), pollutant, process, and SUT for each of the user-specified MOVES runs. The utility also has the option of creating the pollutant/process/SUT/fuel type/road type tab-delimited output file, which lists the number of “null” records, number of not “null” records, the total number of records, minimum value, and maximum value by database table (rateperdistance, rateperprofile, ratepervehicle, rateperstart, and rateperhour for a MOVES rates mode run or movesoutput for a MOVES inventory mode run), pollutant, process, SUT, fuel type, and road type for each of the user-specified MOVES runs.



### OPERATION

The MOVESoutSummary utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or MOVESoutSummary cannot open any of these input files due to an incorrect path and/or filename or any other access error, then MOVESoutSummary results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data

headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the MySQL connection inputs file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

The utility then processes the database listing input file. The utility reads each line, checking the data in the first column. The utility reads each line, checking the data in the first column. If the utility finds an asterisk symbol (\*) in the first column, the line is skipped. If the data in the first column is not the asterisk symbol (\*), the utility recognizes the line as a data line and saves the data line in the database array, which stores a list of each database to process.

The utility then attempts to open the three optional output files (MOVES run summary output file, pollutant/process/SUT tab-delimited output file, and pollutant/process/SUT/fuel type/road type tab-delimited output file). If the text for the MOVES run summary output file is not in the jcf, then the utility will not write this output file. Otherwise, the utility opens the MOVES run summary output file and writes the header line to the output file, which consists of the text *Database Name*, *runSpecFileName*, *runSpecFileDateTime*, *runDateTime*, *minutesDuration*, *defaultDatabaseUsed*, *masterVersionDate*, *masterComputerID*, *domainCountyName*, and *domainDatabaseName*, with each set of text separated by a tab.

If the text for the pollutant/process/SUT tab-delimited output file is not the jcf, then the utility will not write this output file. Otherwise, the utility opens the pollutant/process/SUT tab-delimited output file and writes the header line to the output file, which consists of the text *Database Name*, *Table Name*, *Output Type*, *Pollutant*, *Process*, *Source Type*, *Null Records*, *Not Null Records*, *Total Records*, *Min Value*, and *Max Value*, with each set of text separated by a tab.

If the text for the pollutant/process/SUT/fuel type/road type tab-delimited output file is not in the jcf, then the utility will not write this output file. Otherwise, the utility opens the pollutant/process/SUT/fuel type/road type tab-delimited output file and writes the header line to the output file, which consists of the text *Database Name*, *Table Name*, *Output Type*, *Pollutant*, *Process*, *Source Type*, *Fuel Type*, *Road Type*, *Null Records*, *Not Null Records*, *Total Records*, *Min Value*, and *Max Value*, with each set of text separated by a tab.

The utility then processes each database listed in the database array. The utility sets the temporary database name as first database listed in the database array. The utility uses a MySQL scripts to extract a list of the databases contained in the user’s MySQL data folder and checks this list for a valid database name (i.e., the temporary database name). If the temporary database name is not in the list, then the utility writes an error to the listing file and does not process this database. Otherwise, the utility uses MySQL scripts to extract any errors in the moveserror

database table in the temporary database and writes these errors to the listing output file. The utility also uses MySQL script to determine the type of MOVES run (rates mode or inventory mode) from the movesrun table. The utility then uses MySQL scripts to extract the run spec file name, run spec file date and time, the MOVES run data and time, the duration of the MOVES run, the default database used, the master version date, the master computer ID, the domain county name, and the domain database name from the movesrun database table in the temporary database and writes this data, along with the database name, to the MOVES run summary output file.

The utility then processes the necessary database tables. If the database was found to be from a rates mode run, the utility processes the rateperdistance, rateperprofile, ratepervehicle, rateperstart, and rateperhour database tables. If the database was found to be from an inventory mode run, the utility processes the movesoutput database table. Using MySQL script, the utility extracts the number of records in the database table. If the number of records is 0, then the database table is skipped and no output will be written for that database table. Otherwise, the utility processes the database table to produce the necessary output.

If the pollutant/process/SUT tab-delimited output file was found in the jcf, the utility then begins the procedure for extracting the data for this output file. Using MySQL scripts, the utility extracts the number of “null” records, number of not “null” records, total number of records, minimum value, and maximum value from the database table in the temporary database by pollutant, process, and SUT and writes the data to the pollutant/process/SUT tab-delimited output file. The utility also saves the total number of records in the total number of records array by database and database table.

If the pollutant/process/SUT/fuel type/road type tab-delimited output file was found in the jcf, utility then begins the procedure for extracting the data for this output file. Using MySQL scripts, the utility extracts the number of “null” records, number of not “null” records, total number of records, minimum value, and maximum value from the database table in the temporary database by pollutant, process, SUT, fuel type, and road type (rateperdistance and movesoutput database tables only) and writes the data to the pollutant/process/SUT/fuel type/road type tab-delimited output file.

After processing all of the necessary database tables, the utility repeats the entire procedure (set the temporary database, check for a valid database name, extraction of MOVES errors, extraction of MOVES run summary output file data, extraction of the pollutant/process/SUT tab-delimited output file data, and extraction of the pollutant/process/SUT/fuel type/road type tab-delimited output file data) for each remaining database in the database array. After processing all of the databases, the utility writes a summary of the total number of records by database table for each database in the database array to the listing output file.

## **JCF INPUT FILE**

The jcf input file directs the MOVESoutSummary utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (MOVESoutSummary in this case), or the path and filename of the input or output file. Table 38 shows the file type identifiers. The utility does not recognize

any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 38. MOVESoutSummary File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/MYSQL	MySQL Connection Inputs File	Path and Filename
/DBASE	Database Listing Input File	Path and Filename
/MVRO	MOVES Run Summary Output File <sup>2</sup>	Path and Filename
/PPSO	Pollutant/Process/SUT Number of Records Output File <sup>2</sup>	Path and Filename
/PPSFRO	Pollutant/Process/SUT/Fuel Type/Road Type Number of Records Output File <sup>2</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (MOVESoutSummary), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 91 shows a sample jcf.

/JOB	MOVESoutSummary
/STEP	Step1 MOVESoutSummary
/LST	C:\MOVESoutSummary\Output\MOVESoutSummary.lst
/MYSQL	C:\MOVESoutSummary\Input\MySQL_ConnInputs_MOVES2014.txt
/DBASE	C:\MOVESoutSummary\Input\DatabaseList.txt
/MVRO	C:\MOVESoutSummary\Output\MOVESoutSummary_MOVESrun.tab
/PPSO	C:\MOVESoutSummary\Output\MOVESoutSummary_PolProcSt.tab
/PPSFRO	C:\MOVESoutSummary\Output\MOVESoutSummary_PolProcStFrRDt.tab

**Figure 91. Sample MOVESoutSummary jcf.**

### Job Identifier (/JOB)

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The MOVESoutSummary utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (MOVESoutSummary in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of the database listing input file;
- The path and file name of all output files;
- Each database and any MOVES errors found in the moveserror database table;
- Total number of records summary for each emissions rate database table by database;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **MySQL Connection Inputs File (/MYSQL)**

The /MYSQL identifier (required) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 92 shows a sample MySQL connection inputs file.

```

* Comments can be added by placing * in first column.
* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by
* any combination of tabs and/or spaces.
* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any
* combination of tabs and/or spaces.
* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any
* combination of tabs and/or spaces.
* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.
Port Number = 3307
Username = moves
Password = moves

```

**Figure 92. Sample MySQL Connection Inputs File.**

**Database Listing Input File (/DBASE)**

The /DBASE identifier (required) and its corresponding text string specify the path and filename of the database listing input file, which specifies each database in the summaries. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The utility ignores all comment lines. Other than comment lines, all other lines should list a database name. This database should be in the user’s MySQL data folder. Figure 93 shows a sample database listing input file.

```

* NOTE: The user can add comments by placing the symbol * in the first column.
* NOTE: The utility reads each line not recognized as a comment as a database
* to check.
* NOTE: The user can list only one database per line.
* NOTE: The utility ignores blank lines.
bigratessamplerun_out
harr_natdef2_out

```

**Figure 93. Sample Database Listing Input File.**

**MOVES Run Summary Output File (/MVRO)**

The /MVRO identifier (optional) and its corresponding text string specify the path and filename of the MOVES run summary output file, which is optional. This tab-delimited output file lists the database name, run spec file name, run spec file date and time, the MOVES run data and time, the duration of the MOVES run, the default database used, the master version date, the master computer ID, the domain county name, and the domain database name for each of the databases specified in the database listing input file.

**Pollutant/Process/SUT Number of Records Output File (/PPSO)**

The /PPSO identifier (optional) and its corresponding text string specify the path and filename of the pollutant/process/SUT number of records output file, which is optional. This tab-delimited output file lists the number of “null” records, number of not “null” records, and the total number

of records by database table (rateperdistance, rateperprofile, and ratepervehicle), pollutant, process, and SUT for each of the databases specified in the database listing input file.

**Pollutant/Process/SUT/Fuel Type/Road Type Number of Records Output File (/PPSFRO)**

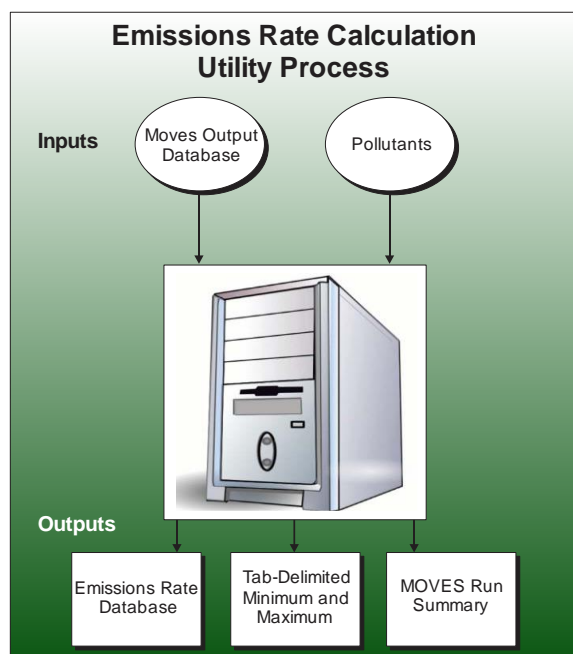
The /PPSFRO identifier (optional) and its corresponding text string specify the path and filename of the pollutant/process/SUT/fuel type/road type number of records output file, which is optional. This tab-delimited output file lists the number of “null” records, number of not “null” records, and the total number of records by database table (rateperdistance, rateperprofile, and ratepervehicle), pollutant, process, SUT, fuel type, and road type for each of the databases specified in the database listing input file. Since the rateperprofile and ratepervehicle database tables do not contain road type information, the utility uses the MOVES road type code 0 for reporting purposes.





## RATESCALC: MOVES EMISSIONS RATES CALCULATION UTILITY

The RatesCalc utility assembles emissions rates in terms of rate-per-activity (from MOVES rateperidistance, rateperhour, and rateperstart database tables and calculated rate-per-SHP emissions rates). The user has the option of creating any combination of these emissions rate tables with each run of the RatesCalc utility. The SHP-based rate-per-activity emissions rates are calculated using the MOVES run CDB and the MOVES default database. For each user-supplied pollutant (excluding fuel consumption or total energy), the emissions rates are assembled for each pollutant and process combination (excluding refueling emissions processes) in the MOVES emissions rate tables. The utility also uses the movesrun database table, along with a pollutant energy or mass lookup table (mass, TEQ, or gmole), to determine the units of the emissions rates, which are added to the emissions rate tables. This allows the user to specify any of the units available in MOVES for the MOVES emissions rate run. Table 39 shows the MOVES2014a emissions process and corresponding RatesCalc input/output database tables.



**Table 39. MOVES Emissions Process and Corresponding RatesCalc Input/Output Database Tables.**

<b>MOVES2014a Emissions Process</b>	<b>Input Database Table</b>	<b>Output Database Table</b>	<b>Emissions Rate Units</b>
Running Exhaust	rateperdistance	ttirateperdistance	Rate/Mile
Crankcase Running Exhaust	rateperdistance	ttirateperdistance	Rate/Mile
Start Exhaust	rateperstart	ttirateperstart	Rate/Start
Crankcase Start Exhaust	rateperstart	ttirateperstart	Rate/Start
Extended Idle Exhaust	rateperhour	ttirateperhour	Rate/Extended Idle Hour
Crankcase Extended Idle Exhaust	rateperhour	ttirateperhour	Rate/Extended Idle Hour
Auxiliary Power Exhaust	rateperhour	ttirateperhour	Rate/APU Hour
Evaporative Permeation	rateperdistance Calculated	ttirateperdistance ttiratepershp	Rate/Mile Rate/SHP
Evaporative Fuel Vapor Venting	rateperdistance Calculated	ttirateperdistance ttiratepershp	Rate/Mile Rate/SHP
Evaporative Fuel Leaks	rateperdistance Calculated	ttirateperdistance ttiratepershp	Rate/Mile Rate/SHP
Brake Wear	rateperdistance	ttirateperdistance	Rate/Mile
Tire Wear	rateperdistance	ttirateperdistance	Rate/Mile

**OPERATION**

The RatesCalc utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until locating the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or RatesCalc cannot open any of these input files due to an incorrect path and/or filename or any other access error, then RatesCalc results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file processed by the utility is the MySQL connection inputs file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

The next input file processed by the utility is the pollutants input file. The utility reads the each line of the pollutants input file and checks the first column of data. If the first column is the asterisk symbol (\*) the utility recognizes the line as a comment and ignores the line. Otherwise, the utility saves the pollutant ID in the pollutants array, provided that the pollutant ID is not related to fuel consumption or total energy (pollutant ID values 91, 92, 93, and 99). The utility repeats this process until processing the entire pollutants input file.

The utility then performs checks on the MOVES rate output database. Using MySQL script, the utility checks that the MOVES rate output database exists in the MySQL data folder. If this database does not exist, the utility writes an error to the listing output file and terminates the current step execution. Otherwise, the utility uses MySQL script to check that the following tables exist: *rateperdistance*, *rateperstart*, *rateperprofile*, *ratepervehicle*, *rateperhour*, *movesrun*, and *moveserror*. If the *movesrun* or *moveserror* tables do not exist, the utility writes an error to the listing output file and terminates the current step execution. If the any of the *rateperdistance*, *rateperstart*, *rateperprofile*, *ratepervehicle*, or *rateperhour* tables do not exist and their respective output is to be created (as specified by the associated output flag in the *jcf*), the utility writes an error to the listing output file and terminates the current step execution.

The utility then processes the *moveserror* table. Using MySQL script, the utility extracts the number of records in the table. If any records are found, the utility extracts the data from this table and writes the data to the listing output file.

The utility then checks that the emissions rate tables (*rateperdistance*, *rateperstart*, *rateperprofile*, *ratepervehicle*, and *rateperhour*) contain data. For each emissions rate table where output is to be created, the utility extracts the number of records in the table using MySQL script. If no records are found in the table, the utility writes a warning to the listing output file.

The utility then checks that the emissions rate tables (*rateperdistance*, *rateperstart*, *rateperprofile*, *ratepervehicle*, and *rateperhour*) for a single MOVESscenarioID. For each

emissions rate table where output is to be created, the utility extracts the number of distinct MOVESscenarioID values in the table using MySQL script. If more than one MOVESscenarioID value is found in the table, the utility writes an error to the listing output file and terminates the current step execution. Using a similar process, the utility checks the emissions rate tables for only one MOVESrunID, yearID, dayID, and monthID (rateperdistance, rateperstart, ratepervehicle, and rateperhour tables only) value.

The utility then checks the emissions rate tables (rateperdistance, rateperstart, rateperprofile, ratepervehicle, and rateperhour) for negative values. For each emissions rate table where output is to be created, the utility extracts the number of records with negative emissions rate values in the table using MySQL script. If any negative values are found in the table, the utility writes an error to the listing output file and terminates the current step execution.

The utility then checks the emissions rate tables (rateperdistance, rateperstart, rateperprofile, ratepervehicle, and rateperhour) for the user-supplied pollutants. For each pollutant, the utility extracts the number of records containing the pollutant from each of the emissions rate tables (rateperdistance, rateperstart, rateperprofile, ratepervehicle, and rateperhour) using MySQL script. If no records are found in a table for the pollutant, a warning is written to the listing output file.

The utility then sets the mass units, day ID, month ID, year ID, and county ID that will be used in the output. Using MySQL script, the utility extracts the mass units from the massUnits data in the movesrun table. The utility also extracts the day ID, month ID, year ID, and county ID values from the emissions rate tables using MySQL script. If the utility is not able to extract any of this data, the utility writes an error to the listing output file and terminates the current step execution.

If the evaporative rate calculation output flag is found in the jcf and set to *YES*, the utility then processes the SHP calculation database tables input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *Table* and *Database* and the utility sets the location of the table name and database name, respectively. For each remaining line, the utility saves the database name in the database tables input array based on the table name. If any of the required tables are missing, the utility writes an error to the listing output file and terminates the current step execution. For each required table, the utility also checks that the table's specified database exists and that the table is in the database using MySQL script. If either the database is not found or the table is not in the database, the utility writes an error to the listing output file and terminates the current step execution.

If the rateperdistance output flag is found in the jcf and set to *YES*, the utility creates a copy of the rateperdistance emissions rates in the specified output database with the table name *ttirateperdistance*. Only the emissions rates for those user-specified pollutants are copied to the *ttirateperdistance* table. The utility also adds a "Units\_Per\_Activity" field to the *ttirateperdistance* table and fills that field based on the pollutants energy or mass designation (mass, TEQ, or gmole as determined by the MOVES defaults). For those pollutants designated as mass, the mass units are added to the "Units\_Per\_Activity" field. For those pollutants designated

as gmole, the mass units from the movesrun table, along with the text “-mole” (i.e., pound-mole or gram-mole) are added to the “Units\_Per\_Activity” field. For those pollutants designated as TEQ, the mass units, along with text “-TEQ” (i.e., pounds-TEQ or grams-TEQ) are added to the “Units\_Per\_Activity” field. No unit conversions are performed in this utility. If the rateperstart or rateperhour output flags are found in the jcf set to *YES*, the emissions rate tables are processed in a similar manner to produce the ttirateperstart and ttirateperhour emissions rate tables.

If the evaporative rate calculation output flag is found in the jcf and set to *YES*, the utility then calculates the evaporative emissions rates in terms of rate-per-SHP. The utility uses the county database (CDB) from the MOVES run and the MOVES default database to replicate the MOVES vehicle population and SHP calculation process. Using the emissions rates from the rateperprofile and ratepervehicle emissions rate tables, the utility calculates the rate-per-SHP emissions rates by multiplying the emissions rate by the appropriate vehicle population and dividing by the appropriate SHP value. These rate-per-SHP emissions rates are then saved in the ttiratepershp emissions rate table. Similar to the previous RatesCalc emissions rate tables, the “Units\_Per\_Activity” field is added to the ttiratepershp table and filled based on the pollutants energy or mass designation.

The utility then writes the MOVES run summary. Using MySQL code, the utility extracts the runSpecFileName, runSpecFileDateTime, runDateTime, minutesDuration, defaultDatabaseUsed, masterVersionDate, masterComputerID, domainCountyName, and domainDatabaseName from the movesrun table and writes this data (along with the input database name) to the movesrun summary output file.

The utility then creates the RatesCalcrun table, which contains information regarding the RatesCalc step execution. Using MySQL script, the utility creates the table in the RatesCalc output database and writes the run date/time, county, year ID, month ID, day ID, MOVES rates output database, the user-specified output tables, the user-specified pollutant ID values, the mass units, and the file name of the listing output file to the table.

If the evaporative rate calculation output flag is found in the jcf and set to *YES* and the activity summary output file is found in the jcf, the utility then assembles the activity summary output. Using MySQL script, the utility extracts the summary data (minimum activity value, maximum activity value, and number of records) by activityTypeID, sourceTypeID, and fuelTypeID from the temporary table created in the evaporative emissions rates calculation process and writes the summary data to the activity summary output file.

If the emissions rates summary output file is found in the jcf, the utility then assembles the emissions rates summary data. For each output emissions rate table (ttirateperdistance, ttirateperstart, and ttirateperhour as specified in the jcf by their respective output flag; ttiratepershp as specified by the evaporative rate calculation output flag), the utility uses MySQL script to extract the summary data (minimum rate value, maximum rate value, and number of records) by pollutantID, processID, sourceTypeID, and fuelTypeID and writes the summary data to the emissions rates summary output file.

## JCF INPUT FILE

The jcf input file directs the RatesCalc utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (RatesCalc in this case), or the path and filename of the input or output file.

Table 40 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 40. RatesCalc File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/OUTDB	Output Database Name	Database Name
/MVRO	Tab-Delimited MOVES Run Summary Output File <sup>2</sup>	Path and Filename
/MMRRO	Tab-Delimited Emission Rates Summary Output File <sup>2</sup>	Path and Filename
/MMRAO	Tab-Delimited Activity Summary Output File <sup>3</sup>	Path and Filename
/MYSQL	MySQL Connection Inputs File	Path and Filename
/POLL	Pollutants Input File	Path and Filename
/MRODB	MOVES Rate Output Database Name	Database Name
/RPDOF	rateperdistance Output Flag	Text (YES or NO)
/RPSOF	rateperstart Output Flag	Text (YES or NO)
/RPHOF	rateperhour Output Flag	Text (YES or NO)
/ERCOF	Evaporative Rate Calculation Output Flag	Text (YES or NO)
/DBTIN	SHP Calculation Database Tables Input File <sup>4</sup>	Path and Filename
/KCDBF	SHP Calculation Output Database Flag <sup>4</sup>	Text (YES or NO)

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional.

<sup>3</sup> Optional. Should only be included if evaporative rate calculation output flag (/ERCOF) is set to "YES".

<sup>4</sup> Required only if evaporative rate calculation output flag (/ERCOF) is set to "YES".

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (RatesCalc), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 94 shows a sample jcf.

/JOB	RatesCalc mvs14_aus1040_2020swkd_48453_er_out
/STEP	Step1 RatesCalc
/LST	C:\RatesCalc\Output\RatesCalc_Test.lst
/OUTDB	mvs14_aus1040_2020swkd_48453_RatesCalc
/MVRO	C:\RatesCalc\Output\RatesCalc_Test_MOVESrunSummary.tab
/MMRRO	C:\RatesCalc\Output\RatesCalc_Test_RatesSummary.tab
/MMRAO	C:\RatesCalc\Output\RatesCalc_Test_ActivitySummary.tab
/MYSQL	C:\RatesCalc\Input\MySQL_ConnInputs_MOVES2014.txt
/POLL	C:\RatesCalc\Input\pollutants.txt
/MRODB	mvs14_aus1040_2020swkd_48453_er_out
/RPDOF	Yes
/RPSOF	Yes
/RPHOF	Yes
/ERCOF	Yes
/DBTIN	C:\RatesCalc\Input\SHPCalc_DatabaseTableInputs.txt
/KCDBF	Yes

**Figure 94. Sample RatesCalc jcf.**

### Job Identifier (/JOB)

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. This utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### Step Identifier (/STEP)

The /STEP identifier (required) and its corresponding text string name the utility to run. The RatesCalc utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### Listing File Output (/LST)

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (RatesCalc in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path, filename, and contents of the MySQL connection inputs file;



- The path, filename, and contents of the pollutants input file;
- The MOVES rates output database name;
- The path, filename, and contents of the SHP calculation database table input file (if input);
- The path and filename of all output files;
- Activity minimum value, maximum value, and number of records summary;
- Emission rates minimum value, maximum value, and number of records summary;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### MySQL Connection Inputs File (/MYSQL)

The /MYSQL identifier (required) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 95 shows a sample MySQL connection inputs file.

<p>* Comments can be added by placing * in first column.  * Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by any combination of tabs and/or spaces.  * Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any combination of tabs and/or spaces.  * If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.  * Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any combination of tabs and/or spaces.  * If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.  * "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.  Port Number = 3307  Username = moves  Password = moves</p>
---

**Figure 95. Sample MySQL Connection Inputs File.**

### Pollutants Input File (/POLL)

The /POLL identifier (required) and its corresponding text string specify the path and filename of the pollutants input file, which specifies the ID codes for the desired pollutants for which the emissions rates will be calculated. The user can add comments at any point in this input file by



placing the asterisk symbol (\*) in the first column. The utility reads any other line in this input file that is not a comment line as an input line. Each input line consists of the MOVES pollutant ID code. Figure 96 shows a sample pollutants input file.

```
* NOTE: Comments can be included in jcf by placing the symbol * in the first column.  
* NOTE: Lists the pollutants (by MOVES pollutantID) to write to the emissions  
* rates output files and listing file.  
* pollutantID  
2  
3  
87
```

**Figure 96. Sample Pollutants Input File.**

### **MOVES Rate Output Database Name (/MRODB)**

The /MRODB identifier (required) and its corresponding text string specify the MOVES rate output database name. This database must be in the MySQL data folder. Only the database name is required. The input database must contain the following MOVES output database tables: rateperdistance (only if the rateperdistance output flag is “Yes”), rateperstart (only if the rateperstart output flag is “Yes”), rateperprofile (only if the evaporative rate calculation output flag is “Yes”), ratepervehicle (only if the evaporative rate calculation output flag is “Yes”), rateperhour (only if the rateperhour output flag is “Yes”), movesrun, and moveserror.

### **rateperdistance Output Flag (/RPDOF)**

The /RPDOF identifier (required) and its corresponding text string specify the rateperdistance output flag, which signifies whether or not the utility creates the ttirateperdistance database table. The only valid input following this identifier is the text *YES* (ttirateperdistance database table is to be created) or the text *NO* (ttirateperdistance database table is not to be created). If this flag is set to the text *YES*, then the rateperdistance table is required in the MOVES rate output database.

### **rateperstart Output Flag (/RPSOF)**

The /RPSOF identifier (required) and its corresponding text string specify the rateperstart output flag, which signifies whether or not the utility creates the ttirateperstart database table. The only valid input following this identifier is the text *YES* (ttirateperstart database table is to be created) or the text *NO* (ttirateperstart database table is not to be created). If this flag is set to the text *YES*, then the rateperstart table is required in the MOVES rate output database.

### **rateperhour Output Flag (/RPHOF)**

The /RPHOF identifier (required) and its corresponding text string specify the rateperhour output flag, which signifies whether or not the utility creates the ttirateperhour database table. The only valid input following this identifier is the text *YES* (ttirateperhour database table is to be created) or the text *NO* (ttirateperhour database table is not to be created). If this flag is set to the text *YES*, then the rateperhour table is required in the MOVES rate output database.

### **Evaporative Rate Calculation Output Flag (/ERCOF)**

The /ERCOF identifier (required) and its corresponding text string specify the evaporative rate calculation output flag, which signifies whether or not the utility calculates the off-network evaporative emissions rates in terms of rate-per-SHP. The only valid input following this identifier is the text *YES* (off-network evaporative rates are calculated) or the text *NO* (off-network evaporative rates are not calculated).

### **SHP Calculation Database Tables Input File (/DBTIN)**

The /DBTIN identifier (optional, only used if the evaporative rate calculation output flag is set to the text *YES*) and its corresponding text string specify the path and filename of the SHP calculation database tables input file, which specifies the database for each of the required tables needed to calculate the off-network evaporative emissions rates in terms of rate-per-SHP. The following database tables are required in this input file:

- sourcetypeyear;
- sourcetypeagedistribution;
- sourceusetype;
- sourcetypeage;
- hpmsvtypeyear, hpmsvtypeday, sourcetypeyearvmt, or sourcetypeedayvmt;
- roadtypedistribution;
- roadtype;
- monthvmtfraction;
- dayvmtfraction;
- hourvmtfraction;
- monthofanyyear;
- avg speeddistribution;
- avg speedbin;
- sourcetypehour;
- dayofanyweek;
- zone; and
- avft.

The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces.

**Table 41. SHP Calculation Database Tables Input File Header Line.**

Header Line Text	Description
Table	Database Table Name
Database	Database Name

Following the header line, the user must list each required (each on a separate line) with the table's associate database, separated by any combination of tabs and/or spaces. Figure 97 shows a sample SHP calculation database tables input file.

Table	Database
sourcetypeyear	mvs14_aus1040_2020s_48453_er_in
sourcetypeagedistribution	mvs14_aus1040_2020s_48453_er_in
sourceusetype	movesdb20141021
sourcetypeage	movesdb20141021
hpmsvtypeyear	mvs14_aus1040_2020s_48453_er_in
roadtypedistribution	mvs14_aus1040_2020s_48453_er_in
roadtype	mvs14_aus1040_2020s_48453_er_in
monthvmfraction	mvs14_aus1040_2020s_48453_er_in
dayvmfraction	mvs14_aus1040_2020s_48453_er_in
hourvmfraction	mvs14_aus1040_2020s_48453_er_in
monthofanyyear	movesdb20141021
avgspeeddistribution	mvs14_aus1040_2020s_48453_er_in
avgspeedbin	movesdb20141021
sourcetypehour	movesdb20141021
dayofanyweek	movesdb20141021
zone	mvs14_aus1040_2020s_48453_er_in
avft	mvs14_aus1040_2020s_48453_er_in

**Figure 97. Sample SHP Calculation Database Tables Input File**

### **SHP Calculation Output Database Flag (/KCDBF)**

The /KCDBF identifier (optional, only used if the evaporative rate calculation output flag is set to the text *YES*) and its corresponding text string specify the SHP calculation output database flag, which signifies whether or not to keep the database tables used in the evaporative rate calculation. The only valid input following this identifier is the text *YES* (database tables will not be deleted) or the text *NO* (database tables will be deleted).

### **Output Database Name (/OUTDB)**

The /OUTDB identifier (required) and its corresponding text string specify the output database name. Only the database name is required. This database will be created in the MySQL data folder and the output database tables will be located in this output database. The emissions rate database tables in this output database are very similar to the MOVES emissions rate databases, with an added field to include the units for the emission rates.

### **Tab-Delimited MOVES Run Summary Output File (/MVRO)**

The /MVRO identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited MOVES run summary output file. This output file is optional and is only created if the movesrun database table is found in the input database. This output file includes the database name, run spec file name, run spec file date/time, the MOVES run date/time, duration minutes of the run, default database used, master version date, master computer ID, domain county name, and domain database name from the movesrun database table in the input database.

### **Tab-Delimited Emission Rates Summary Output File (/MMRRO)**

The /MMRRO identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited emissions rates summary output file. This output file lists the emissions rate summary (minimum rate value, maximum rate value, and number of records) by pollutantID, processID, sourceTypeID, and fuelTypeID.

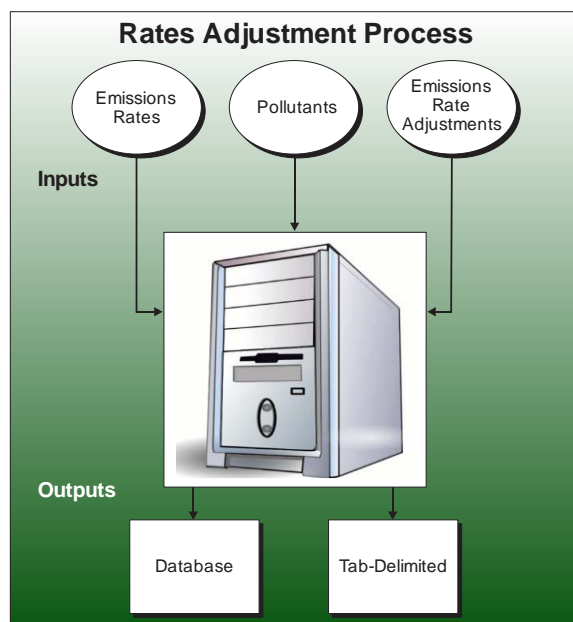
### **Tab-Delimited Activity Summary Output File (/MMRAO)**

The /MMRAO identifier (optional, can only be used if the evaporative rate calculation output flag is set to the text *YES*) and its corresponding text string specify the path and filename of the tab-delimited activity summary output file. This output file lists the activity summary (minimum activity value, maximum activity value, and number of records) by activityTypeID, sourceTypeID, and fuelTypeID.

## RATESADJ: EMISSIONS RATES ADJUSTMENT

The RatesAdj utility applies emissions rate adjustments to an emissions rate database table produced by the RatesCalc utility (ttirateperdistance, ttirateperstart, ttirateperhour, or ttiratepershp) or by this utility to produce a new emissions rate database table in the same format as the input emissions rate database table. The emissions rate adjustments can be linear adjustments that are applied to all emissions rates or can be applied by SUT, fuel type, pollutant, and process (adjustments may also include roadway type, average speed bin, and hour).

The user has the option of selecting which pollutants will be in the new emissions rate database table, along with the output units of the emissions rates. This allows the user to perform any unit conversions between mass units (i.e., pounds to grams or pound-mole to gram-mole or pounds-TEQ to grams-TEQ) without providing any additional adjustment factors. Unit conversions between unit types (i.e., gram-moles to grams or grams-TEQ to grams) are not performed internally by the utility. These types of conversions must be made using the emissions rate adjustment factors. The utility also has the option for combining multiple emissions rate database tables into one new emissions rate database table, if the input emissions rate database tables are in the same format.



### OPERATION

The RatesAdj utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until finding the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until finding the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or RatesAdj cannot open any of these input files due to an incorrect path and/or filename or any other access error, then RatesAdj results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file processed by the utility is the MySQL connection inputs file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data

after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

The next input file processed by the utility is the emissions rate database/table input file. The utility reads each line of this input file and checks the first column of data. If the first column is the asterisk symbol (\*), the utility recognizes the line as a comment and ignores the line. Otherwise, the utility sets the temporary database/table number (the first word excluding the text *DB*) and the temporary database/table name. For example, if the input line is:

```
DB1 harr_out.ttirateperdistance,
```

then the temporary database/table number is 1 and the temporary database/table name is harr\_out.ttirateperdistance. The utility saves the temporary database/table name in the input database information array based on the temporary database/table number. The utility repeats this process until processing the entire emissions rate database/table input file.

The utility then checks that each database/table in the input database information array exists. For each database/table, the utility sets the database name as the text to the left of the period (using the example above, the database name is harr\_out) and the table name as the text to the right of the period (using the example above, the table name is ttirateperdistance). Using MySQL script, the utility checks that the database exists in the MySQL data folder. If the database does not exist, the utility writes an error to the listing file and terminates the current step execution. Otherwise, the utility checks that the table exists in the database using MySQL script. If the table does not exist, the utility writes an error to the listing file and terminates the current step execution. The database name and table name are also saved in the input database information array.

The utility then assembles the database information (database name, table name, number of null records, number of not null records, and total number of records) for each database/table in the input database information array. For each database/table, the utility extracts the number of “null” records, number of not “null” records, and total number of records from the database/table using MySQL scripts and writes this data, along with the database name and table name, to the listing output file.

The utility then checks that database/table in the input database information array for negative emissions rates. For each database/table, the utility uses MySQL script to extract the number of records with negative emissions rate values. If any negative values are found, the utility writes an error to the listing output file and terminates the current step execution.

The utility then determines the contents of the database tables in the input database information array. For the first database table in the input database information array, the utility uses MySQL scripts to extract the distinct hour ID values from the database table and adds these values to the hour ID array. For each remaining database table in the input database information array, the utility uses MySQL scripts to extract the hour ID data and compares that data to the hour ID array. If the utility does not locate the ID value in the hour ID array, then the utility adds that ID value to the hour ID array, thus creating a list of the hour ID values from all of the database tables in the input database information array. The set of unique hour ID values are also written to the listing output file. The utility uses a similar procedure to build the road type ID (ttirateperdistance table only), pollutant/process, average speed bin (ttirateperdistance table only), and SUT/fuel type arrays. The utility then writes these ID values to the listing output file.

The next input file processed by the utility is the pollutants input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *pollutantID* and *OutUnitsPerActivity* and sets the location of the pollutant ID and the output units for the pollutant, respectively. The utility then reads each remaining line and saves the pollutant ID and the output units for the pollutant in the pollutant/units array. If total energy consumption (pollutantID 91) or any of the energy consumption pollutants (pollutantID 92, 93, 99) are found while processing this input file, the utility skips the pollutant (i.e., is not processed by the utility) and writes a warning to the listing output file.

If the emissions rate adjustments input file was not found in the jcf, then no emissions rate adjustments will be applied by the utility. If this input file was found in the jcf, the utility opens this input file and reads each line until locating the *SINGLE* or *PARTS* (capitalization does not matter) as the first word in the line. If the utility locates the text *SINGLE*, then the utility sets the adjustment type to one-factor. If the utility locates the text *PARTS*, then the utility sets the adjustment type to parts. For each remaining line, the utility determines the input by the text in the first word of the line. If the first word in the line is the text *LINEAR*, the linear adjustment is saved as the numeric value following the text. Otherwise, the utility then sets the temporary database/table number (the first word excluding the text *DB*) and the temporary adjustment (either numeric value for linear adjustment or path and file name for parts adjustment). For example, if the input line is:

DB1 C:\RatesAdj\Input\RateAdj\_harr\_rpd.txt,

then the temporary database/table number is 1 and the temporary adjustment is:

C:\RatesAdj\Input\RateAdj\_harr\_rpd.txt.



The temporary database/table number is very important. This temporary database/table number refers to the same database/table number in the emissions rate database/table input file (i.e., database DB1 in the emissions rate database/table input file uses database DB1 in the emissions rate adjustments input file). The utility saves the temporary adjustment in the input database information array.

If the adjustment type is linear, then the utility is finished processing the emissions rate adjustments. If the adjustment type is parts, then the utility processes each adjustment file in the database information array. For each adjustment file, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *SOURCETYPEID*, *FUELTYPEID*, *ROADTYPEID*, *POLLUTANTID*, *PROCESSID*, *AVGSPEEDBINID*, *HOURLID*, and *ADJUSTMENT* and sets the location of the SUT ID, fuel type ID, road type ID, pollutant ID, process ID, average speed bin ID, hour ID, and the emissions rate adjustment, respectively. If the utility does not locate the text *HOURLID* in the header line, then the emissions rate adjustment is for all hours. If the utility cannot locate the text *ROADTYPEID* in the header line, then the emissions rate adjustment is for all road types. If the utility cannot locate the text *AVGSPEEDBINID* in the header line, then the emissions rate adjustment is for all average speed bins. All of the other data (*SOURCETYPEID*, *FUELTYPEID*, *POLLUTANTID*, *PROCESSID* and *ADJUSTMENT*) is required and must be in the header line or the utility results in an error condition and the utility terminates the current step execution. The utility then reads each remaining line in the input file and save the emissions rate adjustment in the data in the adjustments array based on the database/table number and the data locations (as determined by the header line). The utility repeats this process for each database database/table in the input database information array.

For the first input emissions rate database table, the utility extracts the emissions rates for the specified pollutants from the input database emissions rate table, applies any emissions rate adjustments (either the linear adjustment factor to all emission rates or by the individual parts in the adjustments array) , applies any unit conversion adjustments that may be necessary, and saves these adjusted emissions rates. If more than one emissions rate database table is input, then the utility performs a similar calculation process to the first input emissions rate database table for each input emissions rate database table. If pollutants are found in more than one input emissions rate database table, the adjusted emissions rates are summed to produce one emissions rate.

After processing all of the input emissions rate database tables, the utility creates a new emissions rate database table in the same format as the first input emissions rate database table and writes the adjusted emissions rates to this new emissions rate database table. Using MySQL code, the utility also creates a minimum and maximum emissions rate summary for each input emissions rate table and the output emissions rate table by pollutant, process, and source type/fuel type, which is written to the emissions rate Min/Max summary output file.

## **JCF INPUT FILE**

The jcf input file directs the RatesAdj utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the



utility to run (RatesAdj in this case), or the path and filename of the input or output file. Table 42 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 42. RatesAdj File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/MYSQL	MySQL Connection Inputs File	Path and Filename
/DBTIN	Emissions Rate Database/Table Input File	Path and Filename
/POLL	Pollutants Input File	Path and Filename
/DBTA	Emissions Rate Adjustments Input File <sup>2</sup>	Path and Filename
/DRATO	Emissions Rate Output Database	Database Name
/RATMM	Emissions Rate Min/Max Summary Output File <sup>3</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional. Only used if the utility applies emissions rate adjustments to the emissions rates.

<sup>3</sup> Optional.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (RatesAdj), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 98 shows a sample jcf.

/JOB	RatesAdj ttirpd
/STEP	Step1 RatesAdj
/LST	C:\RatesAdj\Output\RatesAdj_ttirpd.lst
/MYSQL	C:\RatesAdj\Input\MySQL_ConnInputs_MOVES2014.txt
/DBTIN	C:\RatesAdj\Input\RatesAdj_DBTIN_ttirpd.txt
/POLL	C:\RatesAdj\Input\pollutants.txt
/DBTA	C:\RatesAdj\Input\RatesAdj_AdjInputFile_Parts.txt
/DRATO	RatesAdj_Output
/RATMM	C:\RatesAdj\Output\RatesAdj_ttirpd_MinMax.tab

**Figure 98. Sample RatesAdj jcf.**

### Job Identifier (/JOB)

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The RatesAdj utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (RatesAdj in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of the emissions rate database/table input file;
- The database information (including database name, table name, number of records summary) for each database in the emissions rate database tables input file;
- Emissions rate database hours, database road types, database pollutants, database processes, database speed bins, database SUTs, and database fuel types;
- The path, filename, and contents of the pollutants input file;
- The path, filename, and contents of the emissions rate adjustments input file;
- The path and filename of each database emissions rate adjustments input file in the emissions rate adjustments input file (if necessary);
- Minimum and maximum emissions rates;
- The path and filename or database/table name of all output files;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

## MySQL Connection Inputs File (/MYSQL)

The /MYSQL identifier (required) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 99 shows a sample MySQL connections input file.

```
* Comments can be added by placing * in first column.
* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by
* any combination of tabs and/or spaces.
* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any
* combination of tabs and/or spaces.
* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any
* combination of tabs and/or spaces.
* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.
Port Number = 3307
Username = moves
Password = moves
```

**Figure 99. Sample MySQL Connections Input File.**

## Emissions Rate Database/Table Input File (/DBTIN)

The /DBTIN identifier (required) and its corresponding text string specify the path and filename of the emissions rate database/table input file, which specifies each emissions rate database table to process and designates each database table with a database/table number. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The utility reads any other line in this input that is not a comment line as an input line. Each input line consists of the database/table identifier (the text *DB*, followed by a number) and the appropriate database and table name (with the database and table separated by the period symbol [.]). The database/table identifier is very important, as it is used to identify correlations between other input files and the designated database/table. The database table in this input file must be in the user’s MySQL data folder. The tables used in this input file (i.e., the portion to the right of period) can be any of the database tables produced by RatesCalc (ttirateperdistance, ttirateperstart, ttirateperhour, or ttiratepershp). Figure 100 shows a sample emissions rate database/table input file.

```

* Database information input file
* NOTE: A * in first column indicates line is a comment line and will not be used by utility.
* Each database.table that should be processed in one run should be listed here (max. of 12).
* Each database.table will be combined to form one output file.
* Database Identifier should be listed first, followed by the database.table, separated by any
* combination of tabs and/or spaces.
* All database identifiers must use the same table.
* DatabaseIdentifier Database.table
DB1 071_sumwkd_er_defaults_out.ttirateperdistance
DB2 071_sumwkd_er_30avgspeed_out.ttirateperdistance

```

**Figure 100. Sample Emissions Rate Database/Table Input File.**

### Pollutants Input File (/POLL)

The /POLL identifier (required) and its corresponding text string specify the path and filename of the pollutants input file, which specifies the MOVES pollutant codes for all pollutants to process from the input emissions rates database tables and their associated units. Any units conversion between mass types (i.e., pounds to grams or pound-moles to gram-moles or pounds-TEQ to grams-TEQ) are performed by the utility. Although unit conversions between unit types (i.e., gram-mole to grams or grams-TEQ to grams) may be listed in this input file, the actual units conversion is not performed by the utility and must be made using the emissions rate adjustment factors. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 43 shows the contents of the pollutants input file header line.

**Table 43. Pollutants Input File Header Line.**

Header Line Text	Description
pollutantID	MOVES pollutantID
OutUnitsPerActivity	Output Units/Activity

Following the header line, the user must list each pollutantID for each pollutant that will be in the emission rate output, followed by the pollutant’s desired units. Only one pollutantID can appear per line. Figure 101 shows a sample rateperdistance pollutants input file.

```

* NOTE: Comments can be included in jcf by placing * in the first column.
* NOTE: Lists the pollutants (by MOVES pollutantID) that will be written to the emission rates output
*       files and listing file.
* NOTE: Also lists the output units for each pollutantID (grams, pounds, tons, grams-TEQ,
*       pounds-TEQ, tons-TEQ, gram-moles, pound-moles, ton-moles)
* NOTE: Unit conversions between mass types (pounds-TEQ to grams-TEQ or pound-moles to
*       gram-moles or pounds to grams) are performed by the utility.
* NOTE: Conversions between unit types (gram-mole to grams or grams-TEQ to grams) are not
*       performed by the utility. These conversions must be made using the emission rate adjustment
*       factors.
pollutantID      OutUnitsPerActivity
2                grams
3                grams
87               grams

```

**Figure 101. Sample Pollutants Input File.**

### **Emissions Rate Adjustments Input File (/DBTA)**

The /DBTA identifier (optional, only used if the emissions rates adjustments are to be applied) and its corresponding text string specify the path and filename of the emissions rate adjustments input file, which specifies the emissions rate adjustments (either by a single factor or by parts) for each database table. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The utility reads any other line in this input file that is not a comment line as an input line. The first input line (other than any optional comments) should be either the text *SINGLE FACTOR* or *PARTS*. If the user directs the utility to adjust the emissions rates by a single factor (i.e., one factor applied to all emissions rates), then use the text *SINGLE FACTOR*. If the user directs the utility to adjust the emissions rates by the parts of the emissions rates (i.e., SUT or fuel type), then use the text *PARTS*.

The next input line should be the text *LINEAR*, followed by the linear adjustment factor, separated by any combination of tabs and/or spaces. This utility adds this linear adjustment factor to the adjusted emissions rates prior to output. The remaining input lines consist of the database/table identifier (the text *DB*, followed by a number) and the appropriate adjustment (a number for the *SINGLE FACTOR* option or the path and filename of the database emissions rate adjustments input file for the *PARTS* option). The database/table identifier is very important.

The database/table identifier in this input file should correspond to the database/table identifier used in the emissions rate database/table input file (i.e., DB1 in the emissions rate database/table input file uses DB1 in the emissions rate adjustments input file). Figure 102 shows a sample emissions rate adjustments input file using the *SINGLE FACTOR* option. Figure 103 shows a sample emissions rate adjustments input file using the *PARTS* option.

```

* Rate adjustment input file
* NOTE: The symbol * in first column indicates the line is a comment line and will not be used by utility.
* Each database listed in the Input Database information input file should be listed here.
* The database identifier (DB1, DB2, ...) should be the same as those listed in the Input Database
* information input file.
* The first line should be the text Single Factor or Parts to indicate the type of input file.
* For the single factor input file, the adjustment for DB1 here will be applied to DB1 in the Input
* Database information input file.
* For the parts input file, the file listed here for DB1 contains the factors that will be applied to
* DB1 in the Input Database information input file.
SINGLE FACTOR
LINEAR  0
DB1     0.48
DB2     0.52

```

**Figure 102. Sample Emissions Rate Adjustments Input File (SINGLE FACTOR Option).**

```

* Rate adjustment input file
* NOTE: The symbol * in first column indicates the line is a comment line and will not be used by
* utility.
* Each database listed in the Input Database information input file should be listed here.
* The database identifier (DB1, DB2, ...) should be the same as those listed in the Input Database
* information input file.
* The first line should be the text Single Factor or Parts to indicate the type of input file.
* For the single factor input file, the adjustment for DB1 here will be applied to DB1 in the Input
* Database information input file.
* For the parts input file, the file listed here for DB1 contains the factors that will be applied to
* DB1 in the Input Database information input file.
PARTS
LINEAR  0
DB1     C:\MOVESratesAdj\Input\RateAdj_DB1.txt
DB2     C:\MOVESratesAdj\Input\RateAdj_DB2.txt

```

**Figure 103. Sample Emissions Rate Adjustments Input File (PARTS Option).**

The database emissions rate adjustments input file for the PARTS option specifies the emissions rate adjustment by SUT, fuel type, road type (optional), pollutant, process, average speed bin (optional), and hour (optional). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The utility reads any other line in this input file that is not a comment line as an input line.

The first input line (other than optional comments) in this input file should be the header, which consists of the text *SOURCETYPEID*, *FUELTYPEID*, *ROADTYPEID*, *POLLUTANTID*, *PROCESSID*, *AVGSPEEDBINID*, *HOURLID*, and *ADJUSTMENT*, with each set of text separated by any combination of tabs and/or spaces. Since road type is optional, the user can omit that text. However, the utility will apply the emissions rate adjustments to all road types if that text is not in the header line. Similarly, the utility will apply the same emissions rate adjustment to all hours if the *HOURLID* text is not in the header line. The same principle also applies to the average speed bins. The lines following the header line should be the emissions rate adjustment data, with the appropriate ID codes and adjustment (as specified in the header line), separated by any

combination of tabs and/or spaces. Figure 104 shows a sample database emissions rate adjustments input file.

pollutantID	processID	sourceTypeID	fuelTypeID	ADJUSTMENT
1	2	53	2	0.96
1	2	61	2	0.96
1	2	62	2	0.96
1	16	53	2	0.96

**Figure 104. Sample Database Emissions Rate Adjustments Input File.**

### **Emissions Rate Output Database (/DRATO)**

The /DRATO identifier (required) and its corresponding text string specify the name of the database table for outputting the adjusted emissions rates. This database, if created, is in the MySQL data folder. Only the name of the database is required (the table name for the emissions rate output is set by the utility based on the input emissions rate database/table name). This database will contain the adjusted emissions rates in a database table with the same format as the input emissions rate database/table.

### **Emissions Rate Min/Max Summary Output File (/RATMM)**

The /RATMM identifier (optional) and its corresponding text string specify the path and filename of the emissions rate min/max summary output file. This tab-delimited file list minimum and maximum emissions rate values by pollutant, process, and SUT/fuel type for each input database/table and the adjusted output emissions rates.





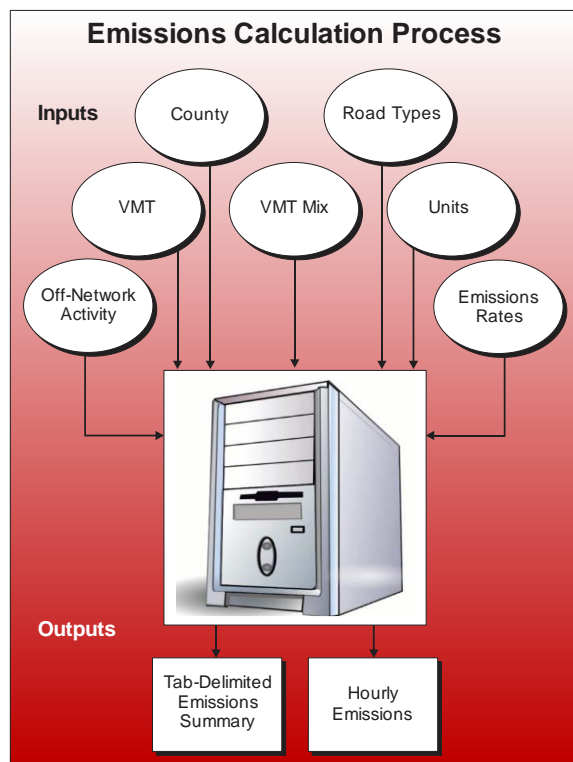
## **EMISSIONS UTILITIES**



## EMSCALC: MOVES-BASED EMISSIONS CALCULATION

The EmsCalc utility estimates on-road mobile hourly link emissions for one user-specified county using the emissions factors (either the RatesCalc utility, or the Ratesadj utility), either a 24-hour or the four time-period VMT mix, the off-network activity (SHP, starts, SHI, and APU hours), and the hourly link VMT and speeds activity estimates (either from the TRANSVMT or VirtualLinkVMT utility). Only on-road, mobile-related emissions (as determined by the pollutant and process ID, see Appendix A) are calculated using EmsCalc.

This utility produces a tab-delimited output file listing the VMT, VHT, calculated speed (VMT/VHT), off-network activity, and emissions in the user-specified units by roadway type and SUT/fuel type combination for each hour of the day. The utility outputs a 24-hour summary as well. The utility also has the option of writing the link-level emissions by SUT/fuel type combination. The utility also has the option of creating a tab-delimited file that contains the activity and emissions output by source classification code (SCC).



### OPERATION

The EmsCalc utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until finding the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until finding the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or EmsCalc cannot open any of these input files due to an incorrect path and/or filename or any other access error, then EmsCalc results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the header input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column. The utility saves this line as the output header, which the utility uses in the tab-delimited output file.

The next input file the utility processes is the county name input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). The utility ignores the first set of text (the text *COUNTY*), uses the second set of text (a numeric value) for the county number, and uses the sets of text for the county name, with a space separating each remaining set of text. The utility then writes the county name and county number to the listing output file.

The next input file the utility processes is the VMT listing input file. The utility reads each line of the file, checking the data in the first column. If the data in the first column is the asterisk symbol (\*), then the utility recognizes the line as a comment and ignores the line. Otherwise, the utility recognizes the line as a data line. For each data line in the input file, the utility ignores the first set of text (the text *Hour*) and saves the hourly VMT file (the third set of text) in part 1 of the hourly inputs array based on the file's respective hour (the second set of text).

The next input file the utility processed is the VMT road type codes input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMT\_RDCODE* and *VMT\_RDdesc* (capitalization does not matter) and sets the location of the VMT road type code and VMT road type description data, respectively. The utility then reads each remaining line and saves the VMT road type description in the VMT road type description array based on the VMT road type code. The utility also adds the "Off-Network" and "TOTALS" to the VMT road type description array as VMT road type code 41 and 42, respectively.

The next input file the utility processes is the VMT road type designations input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMT\_RDCODE*, *VMT\_ATCODE*, *VMTMIX\_RDCODE*, *RATES\_RDCODE*, and *MOVES\_RDCODE* (capitalization does not matter) and sets the location of the VMT road type code, VMT area type code, VMT mix road type, emissions rate road type code, and MOVES road type code data, respectively. The utility then reads each remaining line and saves the VMT mix road type code, rates road type code, and MOVES road type code in the VMT road type designations array based on the VMT road type and VMT area type.

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

If the time period VMT mix is found in the jcf, the utility then processes the time period designation input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *TIMEPER* and *TIMEPERIOD* (capitalization does not matter) and sets the location of the hour and time period data, respectively. The utility then reads each remaining line and saves the time period (must be AM Peak, Midday, PM Peak, or Overnight) in the time periods array based on the hour.

The next input file the utility processes is either the 24-hour VMT mix input file or the AM Peak VMT mix input file. If the time period designation input file is found in the jcf, then the utility processes the AM peak VMT mix file. Otherwise, the utility processes the 24-hour VMT mix input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMX\_RDCODE*, *MOVES\_STCODE*, *MOVES\_FTCODE*, and *VMTMIX* and sets the location of the VMT mix road type code, MOVES SUT code, MOVES fuel type code, and VMT mix data, respectively. For each remaining line, the utility saves the VMT mix in the first section of the VMT mix array based on the VMT mix road type code and MOVES SUT code. The utility repeats this process until processing all of the data lines. The utility also calculates the total of the VMT mix for each SUT across all of the MOVES road types in the VMT mix input file (i.e., the all road types VMT mix by SUT) and saves this data in the VMT mix array as well.

If the time period designation input file is found in the jcf, the utility also processes the Midday, PM Peak, and Overnight VMT mix input files. Using the same process as the AM Peak VMT mix input file, the utility processes each of the remaining VMT mix input files, saving the Midday VMT mix data in section 2 of the VMT mix array, the PM Peak data in section 3 of the mix array, and the Overnight VMT mix data in section 4 of the VMT mix array.

If the tireperhour emissions rates are found in the jcf, the utility then processes the hourly tab-delimited SHP by SUT/fuel type input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, and *SHP* (capitalization does not matter) and sets the location of the hour, MOVES SUT ID, MOVES fuel type ID, and SHP data, respectively. For each remaining line, the utility saves the SHP in the SHP array based on the hour, MOVES SUT and MOVES fuel type codes.

If the tireperhour emissions rates are found in the jcf, the utility then processes the hourly extended Idle and APU activity input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, *ExtIdle*, and *APU* (capitalization does not matter) and sets the location of the hour, MOVES SUT ID, MOVES fuel type ID, SHI, and APU hours data, respectively. For each remaining line, the utility saves the SHI in the SHI portion of the extended idle array and the APU hours in the APU portion of the extended idle array based on the hour, MOVES SUT, and MOVES fuel type codes.

If the *ttirateperstart* emissions rates are found in the *jcf*, the utility then processes the hourly starts activity input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, and *Starts* (capitalization does not matter) and sets the location of the hour, MOVES SUT ID, MOVES fuel type ID, and starts data, respectively. For each remaining line, the utility saves the starts in the starts array based on the hour, MOVES SUT and MOVES fuel type codes.

The next input file the utility processes is the pollutant/tab-delimited output units input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *pollutantID*, *processID*, and *TabOutput\_Units* (capitalization does not matter) and sets the location of the pollutantID, processID, and tab-delimited output units data, respectively. For each remaining line, the utility saves the tab-delimited output units in the units part of the output units array based on the pollutantID and processID. While processing this input file, the utility checks the pollutantID for the total energy consumption or fuel consumption pollutants (pollutantID 91, 92, 93, 99) and the processID for the refueling emissions processes (processID 18, 19, 99). If any of these are found, the utility writes a warning to the listing output file and skips the line (i.e., no data is saved for the pollutantID and processID values).

The utility then processes the MySQL connection inputs file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

If the *ttirateperdistance* emissions rates are not found in the *jcf*, the utility does not process these rates and does not calculate any emissions related to these rates. Otherwise, the utility uses MySQL script to extract the emissions rates from the user-specified database (with the table name set as *ttirateperdistance*) for the user-specified pollutants and processes (from the pollutant/tab-delimited output units input file). The utility then saves these emissions rates in the on-network emissions rates array based on the hour, road type, pollutant, process, average speed bin, SUT, and fuel type.

If the *ttiratepershp* emissions rates are not found in the *jcf*, the utility does not process these rates and does not calculate any emissions related to these rates. Otherwise, the utility uses MySQL script to extract the emissions rates from the user-specified database (with the table name set as *ttiratepershp*) for the user-specified pollutants and processes (from the pollutant/tab-delimited output units input file). The utility then saves these emissions rates in the *ratepershp* emissions rates array based on the hour, pollutant, process, SUT, and fuel type.

If the *ttirateperstart* emissions rates are not found in the *jcf*, the utility does not process these rates and does not calculate any emissions related to these rates. Otherwise, the utility uses

MySQL script to extract the emissions rates from the user-specified database (with the table name set as *ttirateperstart*) for the user-specified pollutants and processes (from the pollutant/tab-delimited output units input file). The utility then saves these emissions rates in the *rateperstart* emissions rates array based on the hour, pollutant, process, SUT, and fuel type.

If the *ttirateperhour* emissions rates are not found in the *jcf*, the utility does not process these rates and does not calculate any emissions related to these rates. Otherwise, the utility uses MySQL script to extract the emissions rates from the user-specified database (with the table name set as *ttirateperhour*) for the user-specified pollutants and processes (from the pollutant/tab-delimited output units input file). The utility then saves these emissions rates in the *rateperhour* emissions rates array based on the hour, pollutant, process, SUT, and fuel type.

The utility then processes the SCC input file. If this input file is not found in the *jcf*, then the utility does not process this input file and output by SCC will not be created. Otherwise, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *SCC*, *fuelTypeID*, *sourceTypeID*, *roadTypeID*, and *processID* (capitalization does not matter) and sets the location of the SCC, fuel type ID, source type ID, road type ID, and process ID data, respectively. For each remaining line, the utility saves the SCC in the SCC array based on the road type ID, source type ID, fuel type ID, and process ID.

The utility then processes the SCC pollutants input file. This input file is required if the SCC output is in the *jcf* (results in an error condition if this input file is not found in the *jcf* and the SCC output is to be written). Otherwise, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *pollutantID*, *SCCpollutant*, and *OutputUnits* (capitalization does not matter) and sets the location of the pollutantID, SCC pollutant, and output units data, respectively. For each remaining line, the utility saves the data in the SCC pollutants array.

The utility then processes the hourly link output files listing input file. If this input file is not found in the *jcf*, then the utility does not process this input file and the link-level output will not be created. Otherwise, the utility reads each line, checking the data in the first column. If the data in the first column is the asterisk symbol (\*), the utility recognizes the line as a comment and ignores the line. If the data in the first column is not the asterisk symbol (\*), the utility recognizes the line as a data line. For each data line in the input file, the utility saves the hourly link emissions output file (segment three of the line) in part 2 of the hourly inputs array based on the file's respective hour (segment two of the line). The utility then checks each hour of the hourly inputs array for an hourly VMT file and an hourly link emissions output file. If an hour has an hourly VMT file and not an hourly link emissions output file or no hourly VMT file and an hourly link emissions output file, then the utility writes a warning to the listing output file.

The utility then begins the emissions calculations process. For the first hour, which is 12:00 a.m. to 1:00 a.m. (Hour 1), the utility calculates the VMT-based emissions. The utility checks part 1 of the hourly inputs array for the path and file name of the hourly VMT input file. If the utility does not find a path and file name, then the utility skips this hour and does not perform



any VMT-based emissions calculations for this hour. Otherwise, the utility sets the temporary VMT input file as part 1 of the hourly inputs array. If the utility locates the hourly link output files listing input file in the jcf and a path and file name are in part 2 of the hourly inputs array, then the utility sets the temporary link emissions output file as part 2 of the hourly inputs array. Otherwise, the utility will not output the link emissions.

The utility then processes the hourly VMT input file. The utility reads the first line and sets the temporary county number (segment three of the line). If the temporary county number does not match the county number from the county name input file, then the utility skips this line. Otherwise, the utility checks the VMT (segment seven of the line). If the VMT is equal to 0, then the utility skips this line. Otherwise, the utility saves the link's ANODE (segment one of the line), BNODE (segment two of the line), road type (segment four of the line), speed (segment six of the line), and VMT in the hourly link data array. The utility also saves the link's VMT mix road type and rates road type (determined using the road type designation data and the link's road type and area type). The utility repeats this process for each line in the VMT input file.

The utility then begins the link-level calculations. For the first link in the hourly link data array, the utility calculates the link VHT (VMT divided by speed). Then for each SUT/fuel type combination in the VMT mix, the utility applies the VMT mix from the VMT mix array (either the 24-hour VMT mix or the appropriate time period VMT mix as assigned by the time period designation array) to the VMT and calculated link VHT. The utility saves this SUT VMT and VHT in the VMT, VHT, and speed array (based on the hour, functional class, and SUT/fuel type combination). The utility then calculates the emissions for each pollutant and process in the on-network emissions rates array. If the link speed is less than 2.5 mph, then the utility uses the 2.5 mph emissions rates for the appropriate road type (determined using the link's rates road type code from the hourly link data array). If the link speed is greater than 75 mph, then the utility uses the 75 mph emissions rates for the appropriate rates road type. Otherwise, the utility interpolates the emissions rates using the emissions rates for the appropriate rates road type for speeds bounding the link speed. For example, if the link speed is 37 mph, then the utility uses the 35 mph emissions rates and the 40 mph emissions rates to calculate the interpolated emissions factor for the link speed of 37 mph. Following is the equation for the interpolated emissions factor ( $EF_{Interp}$ ):

$$EF_{Interp} = EF_{LowSpeed} - FAC_{Interp} H (EF_{LowSpeed} - EF_{HighSpeed})$$

Where:

$EF_{LowSpeed}$  = emissions factor (EF) corresponding to the speed below the average link speed;

$EF_{HighSpeed}$  = EF corresponding to the speed above the average link speed; and

$$FAC_{Interp} = \left( \frac{1}{Speed_{link}} - \frac{1}{Speed_{low}} \right) \bigg/ \left( \frac{1}{Speed_{high}} - \frac{1}{Speed_{low}} \right)$$

The utility calculates the emissions (VMT multiplied by the emissions rate), applies any necessary units conversions to convert the emissions to grams (or gram-moles or grams-TEQ, depending upon the units included with the emissions rates) and saves the emissions in the link



emissions array (by pollutant, process, and SUT/fuel type combination). The utility also saves the emissions in the emissions array (by hour, link road type, pollutant, process, and SUT/fuel type combination). If the SCC output is to be created, the utility determines the appropriate SCC pollutant (using the MOVES pollutantID and the data in the SCC pollutants array) and saves the emissions in the SCC output array by SCC pollutant, SCC road type, SUT/fuel type, and process. The utility repeats this process for each SUT/fuel type combination in the VMT mix. The utility then sets the temporary ANODE and BNODE for the first link in the hourly link data, and then compares this to the next link's ANODE and BNODE. If both ANODE values are equal and both BNODE values are equal, then the utility recognizes these as repeat links. In this case, the utility repeats the SUT/fuel type combination emissions calculations and adds the link emissions to the link emissions array.

If the link emissions are to be output, then the utility writes the link emissions to the temporary link emissions output file. For the VMT-based based emissions, the link emissions consist of the link ANODE, link BNODE, the link roadway classification code, the MOVES road type code, the MOVES pollutant code, the MOVES process code, the link emissions in grams for each SUT/fuel type combination, and the emissions units (grams, gram-moles, or grams-TEQ, depending upon the emissions rate units). The utility writes these by pollutant and process (including a process of 00 for the composite emissions) for each link. The utility repeats this process (calculations by SUT/fuel type combination in the VMT mix and link output, if necessary) for each link in the hourly link data array.

After processing each link in the hourly link data array, the utility then adds the hourly VMT and VHT to the 24-hour part of the VMT, VHT, and speed array. The utility also calculates the speed for the calculation hour (by road type and SUT/fuel type combination) by dividing the VMT by the VHT (from the VMT, VHT, and speed array). The utility saves these speeds in the speed part of the VMT, VHT, and speed array (by hour, road type, and SUT/fuel type combination).

The utility then begins the process of calculating the hourly off-network emissions. For each off-network emissions rate category (ttiratepershp, ttirateperstart, ttirateperhour), the utility only calculates the emissions if the emission rates database is found in the jcf. For each SUT/fuel type combination, the utility calculates the emissions for each pollutant and process in the respective emissions rates array by multiplying the emissions rate by the appropriate activity (see Table 44). The utility applies any necessary units conversions to convert the emissions to grams (or gram-moles or grams-TEQ, depending upon the units included with the emissions rates) and saves the emissions in the link emissions array (by pollutant, process, and SUT/fuel type combination). The utility also saves the emissions in the emissions array (by hour, road type 41 for off-network emissions, pollutant, process, and SUT/fuel type combination). If the SCC output is to be created, the utility determines the appropriate SCC pollutant (using the MOVES pollutantID and the data in the SCC pollutants array) and saves the emissions in the SCC output array by SCC road type, SUT/fuel type, and process. The utility repeats this process for each SUT/fuel type combination.

**Table 44. MOVES2014a Emissions Process and Corresponding Activity.**

<b>MOVES2014a Emissions Process</b>	<b>Input Database Table</b>	<b>Emissions Rate Units</b>	<b>Activity</b>
Running Exhaust	ttirateperdistance	Rate/Mile	VMT
Crankcase Running Exhaust	ttirateperdistance	Rate/Mile	VMT
Start Exhaust	ttirateperstart	Rate/Start	Starts
Crankcase Start Exhaust	ttirateperstart	Rate/Start	Starts
Extended Idle Exhaust	ttirateperhour	Rate/Extended Idle Hour	SHI
Crankcase Extended Idle Exhaust	ttirateperhour	Rate/Extended Idle Hour	SHI
Auxiliary Power Exhaust	ttirateperhour	Rate/APU Hour	APU Hours
Evaporative Permeation	ttirateperdistance ttiratepershp	Rate/Mile Rate/SHP	VMT SHP
Evaporative Fuel Vapor Venting	ttirateperdistance ttiratepershp	Rate/Mile Rate/SHP	VMT SHP
Evaporative Fuel Leaks	ttirateperdistance ttiratepershp	Rate/Mile Rate/SHP	VMT SHP
Brake Wear	ttirateperdistance	Rate/Mile	VMT
Tire Wear	ttirateperdistance	Rate/Mile	VMT

If the link emissions are to be output, then the utility writes the hourly off-network link emissions to the link emissions output file. For the off-network emissions, the utility writes the link emissions for the link ANODE (automatically set to 99999 by the utility), link BNODE (automatically set to 99999 by the utility), the link roadway classification code (automatically set to 99 by the utility), the MOVES road type code (automatically set to 01 by the utility), the MOVES pollutant code, the MOVES process code (including a process of 00 for the composite emissions), and the link emissions in grams, gram-moles, or grams-TEQ (depending upon the units from the emission rates) for each SUT/fuel type combination.

After the utility calculates the off-network emissions, the utility then calculates composite emissions, applies the appropriate unit conversion, calculates the totals road type, and adds the hourly emissions to the 24-hour emissions. For each road type, the utility calculates the pollutant composite emissions across all processes in the emissions array by SUT/fuel type combination and saves the pollutant composite emissions in the emissions array. Then for each road type, pollutant, and process, the utility applies the unit conversion factor (based on the pollutant, process, and the unit conversion part of the output units array) to the emissions in the emissions array by SUT/fuel type combination. The utility also adds these emissions to the totals road type part of the emissions array and the 24-hour emissions part of the emissions array. The utility repeats this entire emissions calculations process (calculation of all emissions) until processing

all 24 hours. The utility then calculates the 24-hour speed by functional class and SUT/fuel type combination by dividing the 24-hour VMT by the 24-hour VHT, which the utility saves in the speed part of the VMT, VHT, and speed array.

The utility then writes the tab-delimited output. The utility writes the output header, the county name, and the data header line (consisting of the text HOUR, ACTIVITY/EMISSIONS, ROADWAY TYPE, and the SUT/fuel type combination abbreviations) to the output file, with each occurring on a separate line (3 different lines). For the first hour, the utility writes the hour (hour 1 is 0:00 AM TO 0:59 AM, hour 2 is 1:00 AM TO 1:59 AM, etc.), the activity/emissions data text (*VEHICLE MILES OF TRAVEL* in this case), the roadway type, and the VMT data by SUT/fuel type combination (including the SUT/fuel type combination totals) for each roadway type (from the VMT, VHT, and speed array) to the tab-delimited output file without roadway type per line. Using a similar process, the utility writes the VHT (using the data output header *VEHICLE HOURS* and the VHT data from the VMT, VHT, and speed array), the speed (using the data output header *SPEED [VMT/VHT]*), and the speed data from the VMT, VHT, and speed array) to the tab-delimited output file. The utility also writes the off-network activity (vehicle population or SHP, starts, SHI, and APU hours) to the tab-delimited output file under the off-network roadway type.

For the first pollutant in the emissions array, the utility then writes the composite emissions (using the same format as the activity with the activity/emissions data consisting of the user-specified units, the text *OF*, the pollutant abbreviation, and the text *EMISSIONS*, with each set of text separated by a space) and the MOVES process-based emissions (using the same format as the activity with the activity/emissions data consisting of the user-specified units, the text *OF*, the pollutant abbreviation, the process abbreviation, and the text *EMISSIONS*, with each set of text separated by a space) to the tab-delimited output. The utility repeats this output for each pollutant in the emissions array. The utility repeats this output process for each hour of the day and for the 24-hour totals. The utility also writes a summary of the hourly total VMT, VHT, speed, off-network activity and emissions (from the emissions array) to the listing output file. The utility writes the 24-hour totals to the listing output file as well.

If the SCC output file is found in the jcf, the utility then writes the SCC output. The utility writes the SCC code, associate activity type, associated activity, and the emissions for each SCC pollutant to this output file for each SCC code that has emissions values.

## JCF INPUT FILE

The jcf input file directs the EmsCalc utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (EmsCalc in this case), or the path and filename of the input or output file.

Table 45 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and these do not affect the execution of the utility.

**Table 45. EmsCalc File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/THDR	Header Input File	Path and Filename
/CTY	County Name Input File	Path and Filename
/VMT	VMT Listing Input File	Path and Filename
/VMTRD	VMT Road Type Codes Input File	Path and Filename
/RDTD	VMT Road Type Designations Input File	Path and Filename
/POLL	Pollutants/Tab-Delimited Output Units Input File	Path and Filename
/VMX	VMT Mix Input File <sup>2</sup>	Path and Filename
/TMPD	Time Period Designation Input File <sup>3</sup>	Path and Filename
/VMXA, /VMXM, /VMXP, /VMXO	AM Peak, Midday, PM Peak, Overnight VMT Mix Input Files, <sup>3</sup> respectively	Path and Filename
/RDIST	ttirateperdistance Emissions Rates Database	Database Name
/RSTRT, /RHOOR, /RSHP	ttirateperstart, ttirateperhour, and ttiratepershp Emissions Rates Database, <sup>4</sup> respectively	Database Name
/SHPI	SHP by SUT/Fuel Type Input File <sup>4</sup>	Path and Filename
/STRTI	Starts by SUT/Fuel Type Input File <sup>4</sup>	Path and Filename
/EIAI	SHI and APU Hours by SUT/Fuel Type Input File <sup>4</sup>	Path and Filename
/MYSQL	MySQL Connection Inputs File	Path and Filename
/SCCI	SCC Input File <sup>5</sup>	Path and Filename
/SCCP	SCC Pollutants Input File <sup>5</sup>	Path and Filename
/TOUT	Tab-Delimited Output File	Path and Filename
/LNKO	Hourly Link Output Files Listing Input File <sup>5</sup>	Path and Filename
/SCCO	Activity and Emissions by SCC Output File <sup>5</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Only if the 24-hour VMT mix is used by the utility.

<sup>3</sup> Only if the time-period VMT mix is used by the utility.

<sup>4</sup> Optional, should only be used if off-network emissions are calculated using the rate-per-activity emissions rates output from the RatesCalc or RatesAdj utilities.

<sup>5</sup> Optional.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (EmsCalc), with each being separated by a space, a tab, or any combination of tabs and spaces. Figure 105 shows a sample jcf.

```

/JOB      AUS_48491_EmsCalc
/STEP     Step1 EmsCalc
/LST      C:\EmsCalc\Output\AUS_48491_EmsCalc.lst
/TOUT     C:\EmsCalc\Output\AUS_48491_EmsCalc.TAB
/LNKO     C:\EmsCalc\Input\LinkFiles_List.txt
/SCCO     C:\EmsCalc\Output\AUS_48491_EmsCalc_SCCoutput.TAB
/THDR     C:\EmsCalc\Input\AUS_EmsCalc_TabHeader.txt
/CTY      C:\EmsCalc\Input\WILLIAMSON.CTY
/VMT      C:\EmsCalc\Input\AUS_VMT_LINKINPUTFILESLIST.TXT
/VMTRD    C:\EmsCalc\Input\Roads\AUSTDM_RdTypeCodes.txt
/RDTD     C:\EmsCalc\Input\Roads\AUSTDM_RdType_desig.txt
/POLL     C:\EmsCalc\Input\EmsCalc_Poll_TabOutUnits.txt
/VMXA     C:\EmsCalc\Input\vmx\D04_Dist_2015WK_VMTmix_AM.vmx
/VMXM     C:\EmsCalc\Input\vmx\D04_Dist_2015WK_VMTmix_MD.vmx
/VMXP     C:\EmsCalc\Input\vmx\D04_Dist_2015WK_VMTmix_PM.vmx
/VMXO     C:\EmsCalc\Input\vmx\D04_Dist_2015WK_VMTmix_ON.vmx
/TMPD     C:\EmsCalc\Input\vmx\TimePeriodDesignation.txt
/MYSQL    C:\EmsCalc\Input\MySQL_ConnInputs_MOVES2014.txt
/RDIST    mvs14_aus48491_swkd_er_outratesadj
/RSTRT    mvs14_aus48491_swkd_er_outratesadj
/RHOUR    mvs14_aus48491_swkd_er_outratesadj
/RSHP     mvs14_aus48491_swkd_er_outratesadj
/SHPI     C:\EmsCalc\Input\OffNetAct\AUS_48491_OffNetActCalc_2014sumwkd_SHP.tab
/EIAI     C:\EmsCalc\Input\OffNetAct\AUS_48491_OffNetActCalc_2014sumwkd_ExtIdle.tab
/STRTI    C:\EmsCalc\Input\OffNetAct\AUS_48491_OffNetActCalc_2014sumwkd_Starts.tab
/SCCI     C:\EmsCalc\Input\MOVES2014_SCCcodes_onroadmobile.tab

```

**Figure 105. Sample EmsCalc jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The EmsCalc utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

## Listing File Output (/LST)

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (EmsCalc in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of all input files (some input files include the contents as well);
- The database name for all input emissions rate databases;
- A summary of the contents of all input emissions rate databases;
- The path and filename of all output output files;
- The begin time, the hourly link emissions output file (if input), the hourly VMT input file, and the end time for each hour in which calculations are performed;
- A summary of the link emissions output data locations in the hourly link emissions output file (only if link emissions are written);
- An hourly summary of number of links from the hourly VMT input files, including the total number of links, the number of links with a VMT greater than 0, and the number of links with a VMT equal to 0;
- An hourly summary (including the 24-hour total) of the VMT, VHT, speed, and emissions;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

## Header Input File (/THDR)

The /THDR (required) identifier and its corresponding text string specify the path and filename of the header input file, which specifies the header that the utility will use when writing the tab-delimited output file. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header that the user intends to appear in the tab-delimited output file. Figure 106 shows a sample header input file.

```
* The user can add comments can be added by placing a * in the first column.  
* The non-comment line will appear in the tab-delimited output file.  
EmsCalc Emissions Run
```

**Figure 106. Sample Header Input File.**

### County Name Input File (/CTY)

The /CTY (required) identifier and its corresponding text string specify the path and filename of the county name input file. The county name input file specifies the county code within the hourly VMT and speeds files for generating the output. This input file consists of a single line containing the identifier text *COUNTY*, the county number, the county name and the county FIPS code in a three-digit format (i.e., 000). The user can separate all of the data within the input line by a tab, space, or any combination of tabs and spaces. Figure 107 shows a sample county input file.

```
COUNTY 1 Upshur 459
```

**Figure 107. Sample County Name Input File.**

### VMT Listing Input File (/VMT)

The /VMT identifier (required) and its corresponding text string specify the path and filename of the VMT listing input file. This input file consists of the identifier text Hour, the hour number (beginning with hour 1, which is 12:00-to-1:00 a.m.), and the path and filename for the respective hourly VMT and speeds file. The hourly VMT and speeds files in this input file can be the output from either the TRANSVMT or VirtualLinkVMT utilities. Figure 108 shows a sample VMT listing input file.

```
Hour 1 C:\MOVESemscal\Input\VMT\upshur_2005fr_summer.T01  
Hour 2 C:\MOVESemscal\Input\VMT\upshur_2005fr_summer.T02  
Hour 3 C:\MOVESemscal\Input\VMT\upshur_2005fr_summer.T03
```

**Figure 108. Sample VMT Listing Input File.**

### VMT Road Type Codes Input File (/VMTRD)

The /VMTRD identifier (required) and its corresponding text string specify the path and filename of the VMT road type codes input file, which specifies each functional class (or road type) code used in the hourly VMT input files and a description of the functional class that the utility will use when writing the tab-delimited output file. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of the text *VMT\_RDCODE* and *VMT\_RDdesc*, separated by any combination of tabs and/or spaces.



Although the data can appear in any order, it is suggested the *VMT\_RDCODE* text be first, followed by the *VMT\_RDdesc* text to avoid any possible issues that might arise with functional class descriptions that contain multiple words. Following the header line, the user should list each functional class code in the hourly VMT input files along with the functional class description. Figure 109 shows a sample VMT road type codes input file.

* NOTE: Lists the road type codes in the VMT input files and their descriptions.		
* NOTE: VMT_RdDesc will appear in the tab delimited output file as shown.		
VMT_RDcode	VMT_RDdesc	
0	RUR IH FWY	110
2	RUR OTH PRIN ART	130
3	RUR MINOR ART	150

**Figure 109. Sample VMT Road Type Codes Input File.**

### VMT Road Type Designations Input File (/RDTD)

The /RDTD identifier (required) and its corresponding text string specify the path and filename of the VMT road type designations input file that correlates each link road type/area type combination to a VMT mix road type, a rates road type, and a MOVES road type. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 46 shows the contents of the VMT road type designations input file header line.

**Table 46. VMT Road Type Designations Input File Header Line.**

Header Line Text	Description
VMT_RDcode	Link VMT and Speeds Road Type Code
VMT_ATcode	Link VMT and Speeds Area Type Code
VMTMIX_RDcode	VMT Mix Road Type Code
RATES_RDcode	Rates Road Type Code
MOVES_RDcode	MOVES Road Type Code
VMT_RDdesc	Link VMT and Speeds Road Type Code Description <sup>1</sup>
VMT_ATdesc	Link VMT and Speeds Area Type Code Description <sup>1</sup>
MOVES_RDdesc	MOVES Road Type Code Description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes data.

Following the header line, the user must list each link road type code/area type code combination (each on a separate line) with the combinations' VMT mix road type code, rates road type code, and MOVES road type code, separated by any combination of tabs and/or spaces. Figure 110 shows a sample VMT road type/MOVES road type input file.



\* NOTE: Lists the road type and area codes in the VMT input files and their road type designations.  
 \* NOTE: Descriptions (VMT\_RDdesc, VMT\_ATdesc, MOVES\_RDdesc) are not used and are optional.

VMT_RDcode	VMT_ATcode	VMT_RDdesc	VMT_ATdesc	VMTMIX_RDcode	RATES_RDcode	MOVES_RDcode
0	3	3	3	3	RUR IH FWY 110	Rural_Unrest
2	3	2	2	2	RUR OTH PRIN ART 130	Rural_Rest
3	3	2	2	2	RUR MINOR ART 150	Rural_Rest

**Figure 110. Sample VMT Road Type Designations Input File.**

**Pollutants/Tab-Delimited Output Units Input File (/POLL)**

The /POLL identifier (required) and its corresponding text string specify the path and filename of the pollutants/tab-delimited output units input file, which specifies the MOVES pollutant ID and MOVES process ID combinations that the emissions will be calculated for and the units for the tab-delimited output file. The utility will only calculate the emissions for those combinations specified in this input file.

To output the composite emissions, use a MOVES process ID of 0. Otherwise, the utility will not write the composite emissions. The available mass options for the tab-delimited output units are grams, pounds, tons, and NONE (link emissions will be calculated but emissions output will not appear in the tab-delimited output), with the TEQ mass type (mass type is determined using the designation for each pollutant from MOVES) being the mass units and the text “-TEQ” (i.e., grams-TEQ) and the gmole mass type being the mass units without the trailing “s” and the text “-moles (i.e., gram-moles). Unit conversions between mass types (i.e., grams to pounds, grams-TEQ to pounds-TEQ, gram-moles to pound-moles) are performed internally by the utility. Unit conversions between unit types (i.e., gram-moles to grams, grams-TEQ to grams) cannot be performed by this utility (must be performed at the emissions rate level) and result in an error.

The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of the text *pollutantID*, *processID*, and *TabOutput\_Units*, separated by any combination of tabs and/or spaces. Following the header line, the user should list each pollutant, process (including processID = 0 for composite emissions) and the desired tab-delimited output units (separated by any combination of tabs and/or spaces). Figure 111 shows a sample pollutants/tab-delimited output units input file.

```

* NOTE: Emissions for only the pollutants and processes listed here will be calculated and appear in
* link emissions.
* NOTE: The TabOutput_Units can be Grams, Pounds, Tons, Grams-TEQ, Pounds-TEQ,
* Tons-TEQ, Gram-moles, Pound-moles, Ton-moles, or None (None means the link
* emissions will be calculated but the emissions will not be included in the tab output).
* NOTE: Unit conversion between mass types (gram-moles to pound-moles or grams to pounds) are
* performed by the utility.
* NOTE: Conversion between unit types (gram-mole to grams) cannot be performed and will result
* in an error.
* NOTE: Composites (processID = 0) must be included for composite emissions to be written to tab
* output file.
pollutantID      processID      TabOutput_Units
2                1             POUNDS
2                2             POUNDS
2                15            POUNDS
2                16            POUNDS
2                17            POUNDS
2                90            POUNDS
2                91            POUNDS
2                0             POUNDS

```

**Figure 111. Sample Units Input File.**

### 24-Hour VMT Mix Input File (/VMX)

The /VMX identifier (optional, only used if the 24-hour VMT mix is to be used in calculations) and its corresponding text string specify the path and filename of the 24-hour VMT mix input file that specifies the 24-hour mix of vehicles by SUT/fuel type combination by road type, with codes 2 through 9 being the MOVES roadway types (see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 47 shows the contents of the 24-hour VMT mix input file header line.

**Table 47. 24-Hour VMT Mix Input File Header Line.**

Header Line Text	Description
VMX_RDcode	VMT Mix Road Type Code
MOVES_STcode	MOVES SUT Code
MOVES_FTcode	MOVES Fuel Type Code
VMTmix	VMT Mix Value
VMX_RDdesc	VMT Mix Road Type Code Description <sup>1</sup>
MOVES_STdesc	MOVES SUT Code Description <sup>1</sup>
MOVES_FTdesc	MOVES Fuel Type Code Description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes and VMT mix data.

Following the header line, the user must list each MOVES SUT/MOVES fuel type combination for each VMT mix road type (each on a separate line) with the combinations' associated VMT mix value, separated by any combination of tabs and/or spaces. The utility will calculate the emissions only for those MOVES SUT/MOVES fuel type combinations in this input file. Figure 112 shows a sample 24-Hour VMT mix input file.

* NOTE: Waco TxDOT District – 2000 Friday						
* NOTE: Descriptions (VMX_RDdesc, MOVES_STdesc, MOVES_FTdesc) are not used and are optional.						
VMX_RDcode	MOVES_STcode	MOVES_FTcode	VMTmix	VMX_RDdesc	MOVES_STdesc	MOVES_FTdesc
2	11	1	0.00055	Rural_Rest	MC	Gas
2	21	1	0.56562	Rural_Rest	PC	Gas
2	21	2	0.00170	Rural_Rest	PC	Diesel

**Figure 112. Sample 24-Hour VMT Mix Input File.**

### Time Period Designation Input File (/TMPD)

The /TMPD identifier (optional, only used if the time period VMT mix is to be used in the calculations) and its corresponding text string specify the path and filename of the time period designation input file, which specifies the time period VMT mix (AM Peak, Midday, PM Peak, or Overnight) each hour will use in the calculations. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 48 shows the contents of the time period designation input file header line.

**Table 48. Time Period Designation Mix Input File Header Line.**

Header Line Text	Description
TIMEPER	Hour
TIMEPERIOD	Time Period

Following the header line, the user must list each hour (1, 2, 3, ..., 23, 24) with the hour's associated time period (AM Peak, Midday, PM Peak, Overnight), separated by any combination of tabs and/or spaces. Figure 113 shows a sample time period designation input file.

TIMEPER	TIMEPERIOD
01	Overnight
02	Overnight
03	Overnight
04	Overnight
05	Overnight
06	Overnight
07	AM Peak
08	AM Peak
09	AM Peak
10	Midday
11	Midday
12	Midday
13	Midday
14	Midday
15	Midday
16	Midday
17	PM Peak
18	PM Peak
19	PM Peak
20	Overnight
21	Overnight
22	Overnight
23	Overnight
24	Overnight

**Figure 113. Sample Time Period Designation Input File.**

**AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/VMXA, /VMXM, /VMXP, /VMXO)**

The /VMXA, /VMXM, /VMXP, and /VMXO identifiers (optional, only used if the time period VMT mix is to be used in the calculations) and corresponding text string specify the path and filename of the AM Peak, Midday, PM Peak, and Overnight VMT mix input files, respectively. These input files specify the time period mix of vehicles by SUT/fuel type combination for each of the MOVES roadway types (codes 2 through 9, see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. The format of the time period VMT mix input files is the same as the 24-hour VMT mix input file (Table 47 shows the contents of the header line and Figure 112 shows a sample of the input file).

**ttirateperdistance Emissions Rates Database (/RDIST)**

The /RDIST identifier (required) and its corresponding text string specify the database name for the ttirateperdistance (VMT-based) emissions rates, which can be the RatesCalc or RatesAdj utility ttirateperdistance database table output. For this input, the utility only requires the

database name (the table name `ttirateperdistance` is automatically set by the utility). See the sample `jcf` (Figure 105) for an example of this input.

### **ttirateperstart Emissions Rates Database (/RSTRT)**

The `/RSTRT` identifier (optional, only used if the off-network emissions are to be calculated using the rate-per-activity emissions rates) and its corresponding text string specify the database name for the `ttirateperstart` (starts-based) off-network emissions rates, which can be the `RatesCalc` or `RatesAdj` utility `ttirateperstart` database table output. For this input, the utility only requires the database name (the table name `ttirateperstart` is automatically set by the utility). See the sample `jcf` (Figure 105) for an example of this input.

### **ttirateperhour Emissions Rates Database (/RHOURL)**

The `/RHOURL` identifier (optional, only used if the off-network emissions are to be calculated using the rate-per-activity emissions rates) and its corresponding text string specify the database name for the `ttirateperhour` (SHI-based and APU hours-based) off-network emissions rates, which can be the `RatesCalc` or `RatesAdj` utility `ttirateperhour` database table output. For this input, the utility only requires the database name (the table name `ttirateperhour` is automatically set by the utility). See the sample `jcf` (Figure 105) for an example of this input.

### **ttiratepershp Emissions Rates Database (/RSHP)**

The `/RSHP` identifier (optional, only used if the off-network emissions are to be calculated using the rate-per-activity emissions rates) and its corresponding text string specify the database name for the `ttiratepershp` (SHP-based) off-network emissions rates, which can be the `RatesCalc` or `RatesAdj` utility `ttiratepershp` database table output. For this input, the utility only requires the database name (the table name `ttiratepershp` is automatically set by the utility). See the sample `jcf` (Figure 105) for an example of this input.

Following the header line, the user must list each `MOVES SUT/MOVES` fuel type combination (each on a separate line) with the combinations' associated population value, separated by any combination of tabs and/or spaces. **Error! Reference source not found.** shows a sample vehicle population by `SUT/fuel` type input file.

### **SHP by SUT/Fuel Type Input File (/SHPI)**

The `/SHPI` identifier (optional, only used if the `ttiratepershp` database is in the `jcf`) and its corresponding text string specify the path and filename of the `SHP` by `SUT/fuel` type input file, which is a tab-delimited file that specifies the hourly `SHP` by `SUT/fuel` type combination. This input file is the same as the tab-delimited `SHP` by `SUT/fuel` type output file created by the `OffNetActCalc` utility. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 49 shows the contents of the `SHP` by `SUT/fuel` type input file header line.

**Table 49. SHP by SUT/Fuel Type Input File Header Line.**

Header Line Text	Description
hourID	Hour
sourceTypeID	MOVES SUT Code
fuelTypeID	MOVES Fuel Type Code
St_Ftdesc	SUT/Fuel Type Description <sup>1</sup>
SHP	Source Hours Parked

<sup>1</sup> Optional, and not used by the utility.

Following the header line, the user must list each hour and MOVES SUT/MOVES fuel type combination (each on a separate line) with the combinations' associated SHP value, separated by any combination of tabs and/or spaces. Figure 114 shows a sample SHP by SUT/fuel type input file.

* Comments can be added by placing a * in first column.				
hourID	sourceTypeID	fuelTypeID	St_Ftdesc	SHP
1	11	1	MC_Gas	49861.654303471
1	21	1	PC_Gas	1789310.46722702
1	21	2	PC_Diesel	1781.72249051851
1	31	1	PT_Gas	577213.244539177

**Figure 114. Sample SHP by SUT/Fuel Type Input File.**

### Starts by SUT/Fuel Type Input File (/STRTI)

The /STRTI identifier (optional, only used if the ttrateperstart database is in the jcf) and its corresponding text string specify the path and filename of the starts by SUT/fuel type input file, which is a tab-delimited file that specifies the hourly starts by SUT/fuel type combination. This input file is the same as the tab-delimited starts by SUT/fuel type output file created by the OffNetActCalc utility. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 50 shows the contents of the starts by SUT/fuel type input file header line.

**Table 50. Starts by SUT/Fuel Type Input File Header Line.**

Header Line Text	Description
hourID	Hour
sourceTypeID	MOVES SUT Code
fuelTypeID	MOVES Fuel Type Code
St_Ftdesc	SUT/Fuel Type Description <sup>1</sup>
Starts	Starts

<sup>1</sup> Optional, and not used by the utility.

Following the header line, the user must list each hour and MOVES SUT/MOVES fuel type combination (each on a separate line) with the combinations' associated starts value, separated by any combination of tabs and/or spaces. Figure 115 shows a sample starts by SUT/fuel type input file.

* Comments can be added by placing a * in first column.				
hourID	sourceTypeID	fuelTypeID	St_Ftdesc	Starts
1	11	1	MC_Gas	90.0849489282618
1	21	1	PC_Gas	52356.9849265487
1	21	2	PC_Diesel	52.1349515568211
1	31	1	PT_Gas	28649.5704557021

**Figure 115. Sample Sarts by SUT/Fuel Type Input File.**

### SHI and APU Hours by SUT/Fuel Type Input File (/EIAI)

The /EIAI identifier (optional, only used if the ttirateperhour database is in the jcf) and its corresponding text string specify the path and filename of the SHI and APU hours by SUT/fuel type input file, which is a tab-delimited file that specifies the hourly SHI and APU hours by SUT/fuel type combination. This input file is the same as the tab-delimited SHI and APU hours by SUT/fuel type output file created by the OffNetActCalc utility. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 51 shows the contents of the SHI and APU hours by SUT/fuel type input file header line.

**Table 51. SHI and APU Hours by SUT/Fuel Type Input File Header Line.**

Header Line Text	Description
hourID	Hour
sourceTypeID	MOVES SUT Code
fuelTypeID	MOVES Fuel Type Code
St_Ftdesc	SUT/Fuel Type Description <sup>1</sup>
Hotelling	Hotelling Hours
ExtIdle	Extended Idle Hours
APU	APU Hours

<sup>1</sup> Optional, and not used by the utility.

Following the header line, the user must list each hour and MOVES SUT/MOVES fuel type combination (each on a separate line) with the combinations' associated hotelling, SHI, and APU hours values, separated by any combination of tabs and/or spaces. Figure 116 shows a sample SHI and APU hours by SUT/fuel type input file.

* Comments can be added by placing a * in first column.						
hourID	sourceTypeID	fuelTypeID	St_Ftdesc	Hotelling	ExtIdle	APU
1	62	2	CLhT_Diesel	16.0678	14.6153	1.4524
2	62	2	CLhT_Diesel	23.7152	21.5715	2.1437
3	62	2	CLhT_Diesel	25.4519	23.1512	2.3007

**Figure 116. Sample SHI and APU Hours by SUT/Fuel Type Input File.**

### MySQL Connection Inputs File (/MYSQL)

The /MYSQL identifier (required) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 117 shows a sample MySQL connections input file.



```

* Comments can be added by placing * in first column.
* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by
* any combination of tabs and/or spaces.
* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any
* combination of tabs and/or spaces.
* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any
* combination of tabs and/or spaces.
* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.
Port Number = 3307
Username = moves
Password = moves

```

**Figure 117. Sample MySQL Connection Inputs File.**

### SCC Input File (/SCCI)

The /SCCI identifier (optional, only used if the SCC output is to be created) and its corresponding text string specify the path and filename of the SCC input file, which specifies each SCC code and associated fuelTypeID, sourceTypeID, roadTypeID, and processID. For the emissions in the activity and emissions by SCC output file to be the same as the tab-delimited output file, each fuelTypeID, sourceTypeID, roadTypeID, and processID combination associated with the emissions must be included in this input file (i.e., an SCC code may appear more than once in this input file to aggregate activity and emissions data). The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 52 shows the contents of the SCC input file header line.

**Table 52. SCC Input File Header Line.**

Header Line Text	Description
SCC	SCC Code
fuelTypeID	MOVES Fuel Type Code
sourceTypeID	MOVES SUT Code
roadTypeID	MOVES Road Type Code
processID	MOVES Process Code

Following the header line, the user must list each hour and MOVES SUT/MOVES fuel type combination (each on a separate line) with the combinations' associated hoteling, SHI, and APU hours values, separated by any combination of tabs and/or spaces. Figure 118 shows a sample SCC input file.

* Comments can be added by placing a * in first column.				
SCC	fuelTypeID	sourceTypeID	roadTypeID	processID
2201210501	1	21	5	1
2202210501	2	21	5	1
2203210501	3	21	5	1

**Figure 118. Sample SCC Input File.**

### SCC Pollutants Input File (/SCCP)

The /SCCP identifier (optional, only used if the SCC output is to be created) and its corresponding text string specify the path and filename of the SCC pollutants input file, which specifies each pollutant that is to be included in the SCC output file and associated SCC pollutants and output units. This input file allows the user to combine multiple MOVES pollutants into one, aggregated pollutant. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of specific text separated by any combination of tabs and/or spaces. Table 53 shows the contents of the SCC input file header line.

**Table 53. SCC Pollutants Input File Header Line.**

Header Line Text	Description
pollutantID	MOVES pollutantID
SCCpollutant	Aggregated pollutant that will be in the SCC output
OutputUnits	Output Units

Following the header line, the user must list each pollutantID and associated SCC pollutant and output units to be included in the SCC output. If an SCC pollutant is to consist of more than one pollutantID, each occurrence must be on a separate line (only one combination of pollutantID, SCC pollutant, and output units can be on a single line). Figure 119 shows a sample SCC pollutants input file.

* Comments can be added by placing a * in first column.		
pollutantID	SCCpollutant	OutputUnits
2	CO	POUNDS
3	NOX	POUNDS
87	VOC	POUNDS

**Figure 119. Sample SCC Pollutants Input File.**

### Tab-Delimited Output File (/TOUT)

The /TOUT identifier (required) and its corresponding text string specify the path and filename of the tab-delimited output file. The tab-delimited output file lists summaries of the VMT, VHT, calculated speed (VMT/VHT), and the emissions as specified in the units input file by roadway type (link road types, MOVES off-network road type, and total) for each SUT/fuel type combination in the VMT mix input file. The utility lists these summaries for each hour and a 24-hour summary is included as well.

### Hourly Link Output Files Listing Input File (/LNKO)

The /LNKO identifier (optional, only used if the link output is to be created) and its corresponding text string specify the path and filename of the hourly link output files listing input file. Due to the increase in execution time of the utility, this input file is optional and the user should only use it if the user wishes to output the hourly link emissions. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. Other than any optional comments, the utility considers each line as an hourly link output file and should consist of the identifier hour, the hour number (beginning with hour 1, which is 12:00 to 1:00 a.m.), and the path and filename of the hourly link output file. Figure 120 shows a sample hourly link output file listing input file.

```
* Comments can be added by placing a * in first column.  
Hour 1 C:\MOVESemscalc\Output\MOVESemscalc_VLink.T01  
Hour 2 C:\MOVESemscalc\Output\MOVESemscalc_VLink.T02  
Hour 3 C:\MOVESemscalc\Output\MOVESemscalc_VLink.T03  
Hour 4 C:\MOVESemscalc\Output\MOVESemscalc_VLink.T04  
Hour 5 C:\MOVESemscalc\Output\MOVESemscalc_VLink.T05
```

**Figure 120. Sample Hourly Link Output Files Listing Input File.**

The hourly link output files in this input file have a specific format set by the utility. For the rateperdistance-based emissions, these output files contain the link ANODE, link BNODE, the link roadway classification code, the MOVES road type code, the MOVES pollutant code, the MOVES process code, the link emissions, and the emissions units (grams, gram-moles, or grams-TEQ) for each SUT/fuel type combination by pollutant and process (including a process of “00” for the composite emissions) for each link. For the off-network emissions, these output files contain the link ANODE (automatically set to 99999 by the utility), link BNODE (automatically set to 99999 by the utility), the link roadway classification code (automatically set to 99 by the utility), the MOVES road type code (automatically set to 01 by the utility), the MOVES pollutant code, the MOVES process code, the off-network emissions, and the emissions units (grams, gram-moles, or grams-TEQ) for each SUT/fuel type combination by pollutant and process (including a process of 00 for the composite emissions). Table 54 shows the format of the hourly link output files.

**Table 54. Hourly Link Output Files Format.**

Field Description	Start Column	End Column	Maximum Character Length	Example
Link ANODE <sup>1</sup>	1	5	5	01916
Link BNODE <sup>1</sup>	7	11	5	10719
Link Roadway Classification Code <sup>2</sup>	13	14	2	08
MOVES Road Type Code	16	17	2	02
MOVES Pollutant Code	19	21	3	002
MOVES Process Code <sup>3</sup>	23	24	2	01
SUT/Fuel Type Combination 1 Emissions	26	35	10 (includes decimal point)	0.03855901
SUT/Fuel Type Combination 2 Emissions <sup>4</sup>	37	46	10 (includes decimal point)	0.03855901
Units	Emissions End Column + 2	Units Start Column + 14	15	grams

<sup>1</sup> Link Nodes are set as 99999 for the off-network emissions.

<sup>2</sup> Link roadway classification code is set to 99 for the off-network emissions.

<sup>3</sup> The process code for the pollutant composite emissions is set to 00.

<sup>4</sup> The SUT/fuel type combination emissions continue across the line until the utility writes all of the SUT/fuel type combinations in the VMT mix input file.

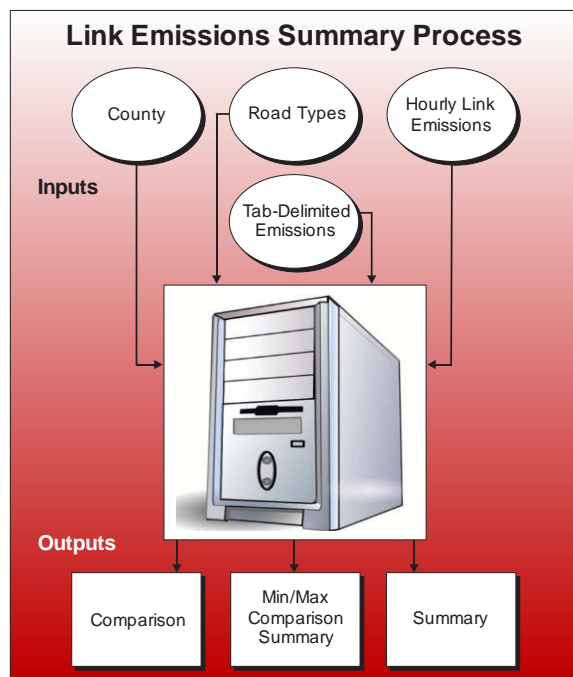
### Activity and Emissions by SCC Output File (/SCCO)

The /SCCO identifier (optional, only used if this output file is to be created) and its corresponding text string specify the path and filename of the activity and emissions by SCC output file. This tab-delimited output file lists emissions summaries for each pollutant and associated activity by SCC.

## LINKEMSSUM: LINK EMISSIONS SUMMARY

The LinkEmsSum utility calculates the total emissions from hourly link emissions output files created by the EmsCalc utility by pollutant, process, and roadway type and creates a tab-delimited output file similar to the tab-delimited output file created EmsCalc. As the utility calculates the total emissions, it also checks the hourly link emissions output file for the proper format.

The utility also has the capability to produce a tab-delimited comparison between the LinkEmsSum tab-delimited output file and the EmsCalc tab-delimited output file. The utility also has the capability to produce an hourly comparison by pollutant and process that includes the totals from LinkEmsSum, the totals from EmsCalc, a comparison of the totals, a minimum of the overall comparison, and a maximum of the overall comparison.



### OPERATION

The LinkEmsSum utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until finding the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until finding the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or LinkEmsSum cannot open any of the input files due to an incorrect path and/or filename or any other access error, then LinkEmsSum results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the header input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). The utility saves this line as the output header, which the utility uses in the tab-delimited output file.

The next input file the utility processes is the VMT road type codes input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMT\_RDCODE* and *VMT\_RDdesc* (capitalization does not matter) and sets the location of the VMT road type code and VMT road type description data, respectively. The utility then reads each remaining line and saves the VMT road type description in the VMT road type description array based on the VMT road type code. The utility also adds the “Off-Network” and “TOTALS” to the VMT road type description array as VMT road type code 41 and 42, respectively.

The next input file the utility processes is the hourly link emissions listing input file. The utility reads each line, checking the data in the first column. If the data in the first column is the asterisk symbol (\*), the utility recognizes the line as a comment and ignores the line. Otherwise, the utility recognizes the line as a data line. For each data line in the input file, the utility saves the hourly link emissions file (segment three of the line) in the link emissions input array based on the file’s respective hour (segment two of the line).

The next input file the utility processes is the tab-delimited input file. The utility reads each line until locating the text *ROADWAY* as the first word in the line. The utility then saves the SUT/fuel type combinations (segments three through the last segment of the line) in the SUT/fuel type array.

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

The next input file the utility processes is the output units input file (this is the same input file as the pollutants/tab-delimited output units input file used with the EmsCalc utility). The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *pollutantID*, *processID*, and *TabOutput\_Units* (capitalization does not matter) and sets the location of the pollutantID, processID, and tab-delimited output units data, respectively. For each remaining line, the utility saves the tab-delimited output units in the units part of the output units array based on the pollutantID and processID. While processing this input file, the checks the pollutantID for the total energy consumption or fuel consumption pollutants (pollutantID 91, 92, 93, 99) and the processID for the refueling emissions processes (processID 18, 19, 99). If any of these are found, the utility writes a warning to the listing output file and skips the line (i.e., no data is saved for the pollutantID and processID values).

The utility then begins processing the hourly link emissions files. For the first hour, the utility opens the hourly link emissions file from the link emissions input array. For each line, the utility then checks the format of the link emissions line. If the format is not correct, then the

utility results in an error condition and terminates the current execution of the utility. Otherwise, the utility sets the temporary road type code (segment three of the line), the temporary pollutant code (segment five of the line), and the temporary process code (segment six of the line). If the pollutant and process are in the output units array, then the utility saves the SUT/fuel type emissions (segment seven through the next to last segment) in the link emissions array by hour, temporary road type code, temporary pollutant code, and temporary process code. The utility repeats this process until processing the entire hourly link emissions file.

After processing the first hour, the utility repeats this process for the remaining 23 hours in the link emissions input array. The utility then applies the correct unit conversion factor from the units conversion part of the output units array (based on the pollutant and process) to the link emissions array and calculates the road type totals, SUT/fuel type totals, and the 24 hour totals.

The utility then processes the tab-delimited input file to save the emissions in the tab-delimited input emissions array. The utility reads each line until it reaches the first data line (skipping all of the header lines). For each data line, the utility sets the temporary hour array code (25 for the 24-hour data, 1 for hour 0:00 AM TO 0:59 AM, 2 for hour 1:00 AM TO 1:59 AM, etc.), the temporary activity/pollutant array code used to save the data (based on the activity/emissions data column and the data in the link emissions array with activity lines being skipped), the temporary process array code (based on the activity/emissions data column and the data in the link emissions array with 0 used for composite emissions), the temporary road type array code (using the road type data column and the VMT road type description array), and saves the emissions data in the tab-delimited input emissions array.

The utility then performs the comparison between the link emissions array and the tab-delimited input emissions array. For each hour, pollutant, process, road type, and SUT/fuel type, the utility subtracts the tab-delimited input emissions (taken from the tab-delimited input emissions array) from the link emissions (taken from the link emissions array). The utility saves this difference in the comparison emissions array by hour, road type, pollutant, process, and SUT/fuel type combination. If this difference is less than the minimum difference (taken from the minimum part of the minimum/maximum emissions comparison array by pollutant, process, and hour), the utility saves the difference in the minimum part of the minimum/maximum emissions comparison array (by pollutant, process, and hour). If this difference is greater than the maximum difference (taken from the maximum part of the minimum/maximum emissions comparison array by pollutant, process, and hour), the utility saves the difference in the maximum part of the minimum/maximum emissions comparison array (by pollutant, process, and hour).

The utility then writes the tab-delimited output file and the tab-delimited emissions comparison output file. The utility writes the header lines (output header, county name, and the data header line consisting of the text HOUR, ACTIVITY/EMISSIONS, ROADWAY TYPE, and the SUT/fuel type combination abbreviations) to the output file. For the first hour, the utility writes the hour (hour 1 is 0:00 AM TO 0:59 AM, hour 2 is 1:00 AM TO 1:59 AM, etc.), the activity/emissions text, and the emissions from the link emissions array by road type (including the MOVES off-network road type and the SUT/fuel type combination totals) for each pollutant and process in the output units array to the tab-delimited output file. Using a similar procedure,



the utility writes the comparison data to the tab-delimited emissions comparison output file using the comparison emissions array instead of the link emissions array, with the text *COMPARISON* added to the output header. For both of these output files, the activity/emissions text consists of the user-specified units, the text *OF*, the pollutant abbreviation, and the text *EMISSIONS* for the composite emissions. For the process-based emissions, the data output header consists of the user-specified units, the text *OF*, the pollutant abbreviation, the process abbreviation, and the text *EMISSIONS*. The utility repeats this output procedure for each hour and for the 24-hour totals. As the comparison is being written, the utility also saves and writes a warning message to the listing output file for those cases where the difference between the values is greater than 0.001 or less than -0.001.

The utility then writes the tab-delimited totals, minimum, and maximum comparison summary output file. For each pollutant and process (including the composite emissions), the utility writes the link total emissions (from the link emissions array), the tab-delimited input emissions (from the tab-delimited input emissions array), the comparison total (from the comparison emissions array), the comparison minimum (from the minimum part of the minimum/maximum emissions comparison array), and the comparison maximum (from the maximum part of the minimum/maximum emissions comparison array) by hour (including the 24-hour values) to this output file. If the tab-delimited 24-hour comparison output file is found in the jcf, the utility also writes the 24-hour values to the 24-hour comparison output file. This output file is unique given that 24-hour values from multiple executions of the utility may be appended in this output file.

## **JCF INPUT FILE**

The jcf input file directs the LinkEmsSum utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (LinkEmsSum in this case), or the path and filename of the input or output file. Table 55 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and these characters do not affect the execution of the utility.



**Table 55. LinkEmsSum File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/THDR	Header Input File	Path and Filename
/VMTRD	VMT Road Type Codes Input File	Path and Filename
/LNKI	Hourly Link Emissions Listing Input File	Path and Filename
/TABI	Tab-Delimited Input File	Path and Filename
/UNIT	Units Input File	Path and Filename
/TOUT	Tab-Delimited Output File	Path and Filename
/TCOUT	Tab-Delimited Emissions Comparison Output File <sup>2</sup>	Path and Filename
/TMOUT	Tab-Delimited Totals, Minimum, and Maximum Comparison Summary <sup>2</sup>	Path and Filename
/TDOUT	Tab-Delimited 24-Hour Totals, Minimum, and Maximum Comparison Summary <sup>2</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional, and only used if the comparisons are to be performed.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (LinkEmsSum), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 121 shows a sample jcf.

/JOB	LinkEmsSum VLink
/STEP	Step1 LinkEmsSum
/LST	C:\LinkEmsSum\Output\VLink\LinkEmsSum_VLink.lst
/THDR	C:\LinkEmsSum\Input\VLink_Hdr.txt
/VMTRD	C:\LinkEmsSum\Input\VLink_RdTypeCodes.txt
/LNKI	C:\LinkEmsSum\Input\VLink_LinkEms_InputList.txt
/TABI	C:\EmsCalc\Output\VLink\EmsCalc_VLink.TAB
/UNIT	C:\LinkEmsSum\Input\EmsCalc_rates_Units.txt
/TOUT	C:\LinkEmsSum\Output\VLink\LinkEmsSum_VLink.TAB
/TCOUT	C:\LinkEmsSum\Output\VLink\LinkEmsSum_VLink_Comp.TAB
/TMOUT	C:\LinkEmsSum\Output\VLink\LinkEmsSum_VLink_TotsMinMax.TAB

**Figure 121. Sample LinkEmsSum jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The LinkEmsSum utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (LinkEmsSum in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of all input files (some may include contents as well);
- The unit conversion factor by pollutant and process;
- The path and filename of all tab-delimited output files;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **Header Input File (/THDR)**

The /THDR identifier (required) and its corresponding text string specify the path and filename of the header input file, which specifies the header that the utility will use when writing the tab-delimited output file. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header that the user intends to appear in the tab-delimited output file. Figure 122 shows a sample header input file.

```
* Comments can be added by placing a * in first column.  
* The non-comment line will appear in the tab-delimited output file.  
LinkEmsSum Emissions Run
```

**Figure 122. Sample Header Input File.**

**VMT Road Type Codes Input File (/VMTRD)**

The /VMTRD identifier (required) and its corresponding text string specify the path and filename of the VMT road type codes input file, which specifies each link road type code used in the hourly link emissions input files and a description of the functional class that the utility will use when writing the tab-delimited output file. This input file is the same VMT road type codes input file used with the EmsCalc utility. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of the text *VMT\_RDCODE* and *VMT\_RDdesc*, separated by any combination of tabs and/or spaces. Although the data can appear in any order, the *VMT\_RDCODE* text should appear first, followed by the *VMT\_RDdesc* text to avoid any possible issues that might arise with road type descriptions that contain multiple words. Following the header line, list each functional class code used in the hourly VMT input files along with the functional class description. Figure 123 shows a sample VMT road type codes input file.

```
* Comments can be added by placing a * in first column.  
Hour 1 C:\MOVESemscal\Output\MOVESemscal_VLink.T01  
Hour 2 C:\MOVESemscal\Output\MOVESemscal_VLink.T02  
Hour 3 C:\MOVESemscal\Output\MOVESemscal_VLink.T03  
Hour 4 C:\MOVESemscal\Output\MOVESemscal_VLink.T04  
Hour 5 C:\MOVESemscal\Output\MOVESemscal_VLink.T05
```

**Figure 123. Sample VMT Road Type Codes Input File.**

**Hourly Link Emissions Listing Input File (/LNKI)**

The /LNKI identifier (required) and its corresponding text string specify the path and filename of the hourly link emissions listing input file. This input file is the same as the hourly link output files listing input file used with the EmsCalc utility. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. Other than any optional comments, each line is considered an hourly link output file and should consist of the identifier Hour, the hour number (beginning with hour 1, which is 12:00 to 1:00 a.m.), and the path and filename of the hourly link output file. Figure 124 shows a sample hourly link emissions listing input file.

```

* NOTE: Lists the road type codes in the VMT input files and their descriptions.
* NOTE: VMT_RdDesc will appear in tab-delimited output file as shown.
VMT_RDcode      VMT_RDdesc
0                RUR IH FWY 110
2                RUR OTH PRIN ART 130
3                RUR MINOR ART 150

```

**Figure 124. Sample Hourly Link Emissions Listing Input File.**

### Tab-Delimited Input File (/TABI)

The /TABI identifier (required) and its corresponding text string specify the path and filename of the tab-delimited input file. This input file is the tab-delimited output from the EmsCalc utility. The tab-delimited input file lists summaries of the VMT, VHT, calculated speed (VMT/VHT), and the emissions as specified in the units input file by roadway type (link road types, MOVES off-network road type, and total) for each SUT/fuel type combination. These summaries should include each hour and a 24-hour summary as well.

### Output Units Input File (/UNIT)

The /UNIT identifier and its corresponding text string specify the path and filename of the output units input file, which specifies the MOVES pollutant ID and MOVES process ID combinations that the emissions will be calculated for and the units for the tab-delimited output file. This input file is the same as the pollutants/tab-delimited output units input file used with the EmsCalc utility. The utility will only calculate the emissions for those combinations specified in this input file.

To output the composite emissions, use a MOVES process ID of 0. Otherwise, the utility will not write the composite emissions. The available mass options for the tab-delimited output units are grams, pounds, tons, and NONE (link emissions will be calculated but emissions output will not appear in the tab-delimited output), with the TEQ mass type (mass type is determined using the designation for each pollutant from MOVES) being the mass units and the text “-TEQ” (i.e., grams-TEQ) and the gmole mass type being the mass units without the trailing “s” and the text “-moles (i.e., gram-moles). Unit conversions between mass types (i.e., grams to pounds, grams-TEQ to pounds-TEQ, gram-moles to pound-moles) are performed internally by the utility. Unit conversions between unit types (i.e., gram-moles to grams, grams-TEQ to grams) cannot be performed by this utility (must be done at the emission rate level) and result in an error.

The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line, which consists of the text *pollutantID*, *processID*, and *TabOutput\_Units*, separated by any combination of tabs and/or spaces. Following the header line, the user should list each pollutant, process (including processID = 0 for composite emissions) and desired tab-delimited output units (separated by any combination of tabs and/or spaces). Figure 125 shows a sample output units input file.

\* NOTE: Emissions for only the pollutants and processes listed here will be calculated and appear in link emissions.

\* NOTE: The TabOutput\_Units can be Grams, Pounds, Tons, Grams-TEQ, Pounds-TEQ, Tons-TEQ, Gram-moles, Pound-moles, Ton-moles, or None (None means the link emissions will be calculated but the emissions will not be included in the tab output).

\* NOTE: Unit conversion between mass types (gram-moles to pound-moles or grams to pounds) are performed by the utility.

\* NOTE: Conversion between unit types (gram-mole to grams) cannot be performed and will result in an error.

\* NOTE: Composites (processID = 0) must be included for composite emissions to be written to tab output file.

pollutantID	processID	TabOutput_Units
2	1	POUNDS
2	2	POUNDS
2	15	POUNDS
2	16	POUNDS
2	17	POUNDS
2	90	POUNDS
2	91	POUNDS
2	0	POUNDS

**Figure 125. Sample Output Units Input File.**

**Tab-Delimited Output File (/TOUT)**

The /TOUT identifier and its corresponding text string specify the path and filename of the tab-delimited output file. The tab-delimited output file lists summaries of the link emissions as specified in the units input file by roadway type (link functional classes, MOVES off-network road type, and total) for each SUT/fuel type combination in the tab-delimited input file. These summaries include each hour and a 24-hour summary as well.

**Tab-Delimited Emissions Comparison Output File (/TCOUT)**

The /TCOUT identifier and its corresponding text string specify the path and filename of the tab-delimited emissions comparison output file. The tab-delimited emissions comparison output file lists summaries of the comparison between the link emissions as specified in the units input file and the emissions from the tab-delimited input file by roadway type (link functional classes, MOVES off-network road type, and total) for each SUT/fuel type combination in the tab-delimited input file. These summaries include each hour and a 24-hour summary as well.

**Tab-Delimited Totals, Minimum, and Maximum Comparison Summary Output File (/TMOUT)**

The /TMOUT identifier and its corresponding text string specify the path and filename of the tab-delimited totals, minimum, and maximum comparison summary output file. For each pollutant and process, this output file lists the link emissions total, tab-delimited input file total, a comparison of the totals, the minimum of the comparison performed to create the tab-delimited emissions comparison output file, and the maximum of the comparison performed to create the tab-delimited emissions comparison output file by hour.

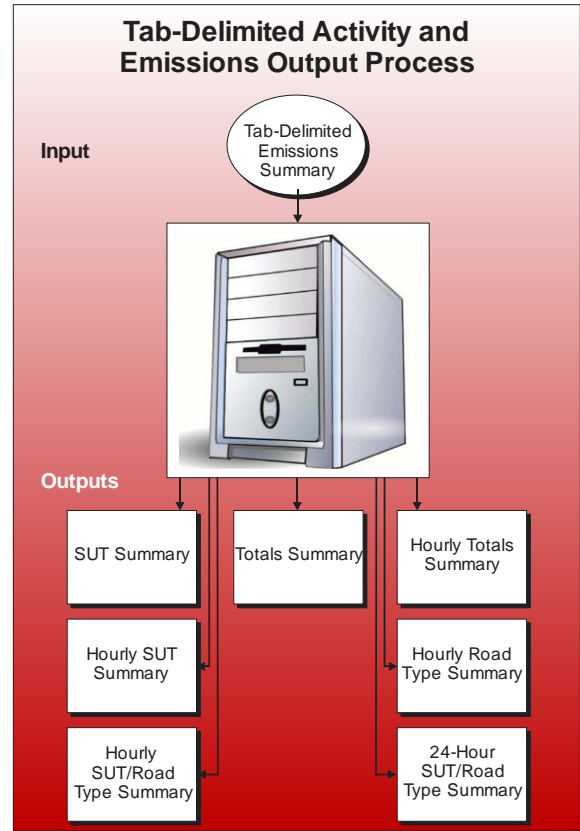
## **Tab-Delimited 24-hour Totals, Minimum, and Maximum Comparison Summary Output File (/TDOUT)**

The /TDOUT identifier and its corresponding text string specify the path and filename of the tab-delimited 24-hour totals, minimum, and maximum comparison summary output file. For each pollutant and process, this output file lists the 24-hour link emissions total, tab-delimited input file total, a comparison of the totals, the minimum of the comparison performed to create the tab-delimited emissions comparison output file, and the maximum of the comparison performed to create the tab-delimited emissions comparison output file. Output from multiple executions of this utility may be appended to this output file.

## TABFILETOTALS: EMSCALC TAB-DELIMITED OUTPUT SUMMARY

The TabFileTotals utility creates summary report files for one or more user-specified EmsCalc tab-delimited output files. The user can specify the EmsCalc tab-delimited output files by either listing the direct path and filename of the EmsCalc tab-delimited output file or by listing the path and filename using the asterisk symbol (\*), which allows the user to specify multiple EmsCalc tab-delimited output files in one run of the utility.

This utility has seven optional output files: SUT tab-delimited output file (24 hour VMT, VHT, speed, off-network activity, and emissions totals by SUT), totals tab-delimited output file (24 hour VMT, VHT, speed, off-network activity, and emissions totals), hourly totals tab-delimited output file (hourly VMT, VHT, speed, off-network activity, and emissions), hourly SUT tab-delimited output file (hourly VMT, VHT, speed, off-network activity, and emissions totals by SUT), hourly road type totals tab-delimited output file (hourly VMT, VHT, speed, off-network activity, and emissions totals by road type), hourly SUT/road type totals tab-delimited output file (hourly road type total VMT, VHT, speed, off-network activity, and emissions totals by SUT), and 24-hour SUT/road type totals tab-delimited output file (24-hour road type total VMT, VHT, speed, off-network activity, and emissions totals by SUT).



### OPERATION

The TabFileTotals utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until finding the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens jcf input file. The utility reads each line of the jcf until locating the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or TabFileTotals cannot open any of the input files due to an incorrect path and/or filename or any other access error, then TabFileTotals results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.



The first input file the utility processes is the header input file. If this input file is missing from the jcf, then this input file is not processed and no header will be included in the tab-delimited output files. Otherwise, the utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column. The utility saves this line as the header line, which the utility uses in the tab-delimited output files.

The next input file the utility processes is the tab-delimited file name with wildcard input. The utility extracts the name of each tab-delimited EmsCalc output file located in the appropriate directory (as specified in the tab-delimited file name with wildcard input) and saves the path and filename of these files in the tab-delimited input file array. The process of assembling the data and writing the output (discussed below) will be performed for each tab-delimited EmsCalc output file separately.

The utility then builds the pollutants array, county array, SUT/fuel type array, and the road type array based on the data in the first tab-delimited EmsCalc output file (from the tab-delimited input file array). The utility sets the first three positions in the pollutants array to VMT, VHT, and Speed, respectively. For the first tab-delimited EmsCalc output file in the tab-delimited input file array, the utility reads the second line and sets the county name for this file as the text following the colon symbol (:).

After skipping the next three lines, the utility reads the next line, which is the data header line consisting of the text HOUR, ACTIVITY/EMISSIONS, ROADWAY TYPE, and the SUT/fuel type combinations including the TOTALS category. If the text is the roadway type line and the SUT/fuel type combinations are not in the SUT/fuel type array, then the utility adds the SUT/fuel type combinations to the SUT/fuel type array.

The utility then reads each line until locating the first data line for the 24-hour output (text 24 HOUR in the hour data column). For this line (and each remaining line), the utility adds the road type text to the road type array and the activity/emissions text to the pollutants array (if the text is not already in the array). The utility then processes the tab-delimited EmsCalc output file and saves the VMT, VHT, speed, off-network activity, and emissions data in the data array. The utility skips the first five lines, which are all header lines. All remaining lines are considered data lines. For each data line, the utility sets a temporary hour code (based on the data in the HOUR column), temporary activity/pollutant code (based on the data in the ACTIVITY/EMISSIONS column and the pollutants array), temporary road type code (based on the ROADWAY TYPE column and the road type array), and saves the activity/emissions data for each SUT/fuel type combination (including the TOTALS) in the data array.

The utility then writes the totals tab-delimited output file. If this output file is missing from the jcf, then the utility does not write this output file. If this file does not exist (utility is creating the file), the utility writes the header line (from the header input file) and the data header, which consists of the text *County*, *Data Type*, *UNITS*, and *24 HOUR* to the totals tab-delimited output file, with each set of text separated by a tab. The utility writes the county, the VMT, the units, and the 24-hour data (all taken from the data array) to the output file, with each set of data separated by a tab. The utility also writes the VHT, speed, off-network activity, and emissions data in a similar manner.



The utility then writes the SUT tab-delimited output file. If this output file is missing from the jcf, then the utility does not write this output file. If this file does not exist (utility is creating the file), the utility writes the header line (from the header input file) and the data header, which consists of the text *County, Data Type, UNITS*, and the SUT/fuel type combinations from the SUT/fuel type array to the totals tab-delimited output file, with each set of text separated by a tab. The utility writes the county, the VMT, the units, and the 24-hour data by SUT/fuel type combination (all taken from the data array) to the output file, with each set of data separated by a tab. The utility also writes the VHT, speed, off-network activity, and emissions data in a similar manner.

The utility then writes the hourly totals tab-delimited output file. If this output file is missing from the jcf, then the utility does not write this output file. If this file does not exist (utility is creating the file), the utility writes the header line (from the header input file) and the data header, which consists of the text *County, Data Type, UNITS*, and the hours (i.e., Hour 1, Hour 2, ... , Hour 24, 24 hour), with each set of text separated by a tab. The utility writes the county, the VMT, the units, and the hourly totals data (all taken from the data array) to the output file, with each set of data separated by a space. The utility also writes the VHT, speed, off-network activity, and emissions data in a similar manner.

The utility then writes the SUT tab-delimited output file. If the text for this output file is missing from the jcf, then the utility does not write this output file. If this file does not exist (utility is creating the file), the utility writes the header line (from the header input file) and the data header, which consists of the text *County, Data Type, UNITS, Hour*, and the SUT/fuel type combinations from the SUT/fuel type array to the totals tab-delimited output file, with each set of text separated by a tab. The utility writes the county, the VMT, the units, the hour (i.e., Hour 1, Hour 2, ... , Hour 24, 24 hour), and the hourly data by SUT/fuel type combination (all taken from the data array) to the output file for each hour (including the 24-hour data), with each set of data separated by a space. The utility also writes the VHT, speed, off-network activity, and emissions data in a similar manner.

The utility then writes the hourly road type totals tab-delimited output file. If the text for this output file is missing from the jcf, then the utility does not write this output file. If this file does not exist (utility is creating the file), the utility writes the header line (from the header input file) and the data header, which consists of the text *County, Data Type, UNITS, Roadtype*, and the hours (i.e., Hour 1, Hour 2, ... , Hour 24, 24 hour), with each set of text separated by a tab. The utility writes the county, the VMT, the units, the road type (taken from the road type array), and the hourly road type totals data (all taken from the data array except the road type) to the output file, with each set of data separated by a space. The utility also writes the VHT, speed, off-network activity, and emissions data in a similar manner.

The utility then writes the hourly SUT/road type tab-delimited output file. If the text for this output file is missing from the jcf, then the utility does not write this output file. If this file does not exist (utility is creating the file), the utility writes the header line (from the header input file) and the data header, which consists of the text *County, Data Type, UNITS, Hour, Roadtype*, and the SUT/fuel type combinations from the SUT/fuel type array to the totals tab-delimited output file, with each set of text separated by a tab. For the first tab-delimited EmsCalc output file in the

tab-delimited input file array, the utility writes the county, the VMT, the units, the hour (i.e., Hour 1, Hour 2, ... , Hour 24, 24 hour), the road type (taken from the road type array), and the hourly road type data by SUT/fuel type combination (all taken from the data array except the road type) to the output file for each road type and hour (including the 24-hour data), with each set of data separated by a space. The utility also writes the VHT, speed, off-network activity, and emissions data in a similar manner. The utility repeats this procedure (setting the temporary county, building the data arrays, and writing the output) for each tab-delimited EmsCalc output file in the tab-delimited input file array.

## JCF INPUT FILE

The jcf input file directs the TabFileTotals utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (TabFileTotals in this case), or the path and filename of the input or output file. Table 56 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 56. MOVESTabFileTotals File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/TABI	Tab-Delimited Input File Name With Wildcard <sup>2</sup>	Path and Filename
/HDRO	Header Input File <sup>3</sup>	Path and Filename
/SOUT	SUT Tab-Delimited Output File <sup>3</sup>	Path and Filename
/TOUT	Totals Tab-Delimited Output File <sup>3</sup>	Path and Filename
/THOUT	Hourly Totals Tab-Delimited Output File <sup>3</sup>	Path and Filename
/SHOUT	Hourly SUT Tab-Delimited Output File <sup>3</sup>	Path and Filename
/TROUT	Hourly Road Type Totals Tab-Delimited Output File <sup>3</sup>	Path and Filename
/SROUT	Hourly SUT/Road Type Totals Tab-Delimited Output File <sup>3</sup>	Path and Filename
/DSROUT	24-hour SUT/Road Type Totals Tab-Delimited Output File <sup>3</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> May include the wildcard, the asterisk symbol (\*), to access more than one tab-delimited input file.

<sup>3</sup> Optional.

The only format requirement for the input and output file lines listed in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (TabFileTotals), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 126 shows a sample jcf.

/JOB	TabFileTotals
/STEP	Step1 TabFileTotals
/LST	C:\TabFileTotals\Output\TabFileTotals.lst
/TABI	C:\TabFileTotals\Input\tabfiles\MOVESemscal_*.TAB
/HDRO	C:\TabFileTotals\Input\Header.txt
/SOUT	C:\TabFileTotals\Output\TabFileTotals_STout.tab
/TOUT	C:\TabFileTotals\Output\TabFileTotals_Totals.tab
/THOUT	C:\TabFileTotals\Output\TabFileTotals_HrTotals.tab
/SHOUT	C:\TabFileTotals\Output\TabFileTotals_HrSTout.tab
/TROUT	C:\TabFileTotals\Output\TabFileTotals_RdTypeTotals.tab
/SROUT	C:\TabFileTotals\Output\TabFileTotals_RdTypeST.tab
/DSROUT	C:\TabFileTotals\Output\TabFileTotals_24hrRdTypeST.tab

**Figure 126. Sample TabFileTotals jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. The utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The TabFileTotals utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (TabFileTotals in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;

- JCF through the step being executed;
- The tab input file name with a wildcard;
- The path and filename of each EmsCalc tab-delimited output file to process;
- The path, filename, and contents of the header input file (if found in the jcf);
- The path and filename of all output files;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **Tab-Delimited Input File Name with Wildcard (/TABI)**

The /TABI identifier (required) and its corresponding text string specify the path and filename of the tab-delimited input file name with a wildcard. This is not a path and file name to an external input file. This input is only the path and filename (including the wildcard character, the asterisk symbol [\*]) of the tab-delimited EmsCalc output files.

### **Header Input File (/HDRO)**

The /HDRO identifier (optional) and its corresponding text string specify the path and filename of the header input file, which specifies the header that will be used by the utility when writing the tab-delimited output file. This input file is optional. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header that the user intends to appear in the tab-delimited output file. Figure 127 shows a sample header input file.

<p>* The user can add comments by placing the symbol * in the first column          * The utility recognizes the first line without a * in the first column as the          * header.          TabFileTotals Tab-Delimited Output Header</p>
--

**Figure 127. Sample Header Input File.**

### SUT Tab-Delimited Output File (/SOUT)

The /SOUT identifier and its corresponding text string specify the path and filename of the SUT tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity and emissions), the data units, and 24-hour SUT data (including the totals data) for each tab-delimited EmsCalc output file. Figure 128 shows a sample SUT tab-delimited output file.

TabFileTotals Tab-Delimited Output Header							
County	Data Type	UNITS	MC_Gas	PC_Gas	PC_Diesel	...	TOTALS
Brazoria 039	VMT	Miles	7050.020	7007533.624	7007.198	...	9902656.72
Brazoria 039	VHT	Hours	170.337	169204.669	169.006	...	235686.151
Brazoria 039	Speed	MPH	41.388	41.414	41.461	...	42.016
Brazoria 039	SHP	Hours	506004.398	17009186.52	51119.752	...	23533373.28
Brazoria 039	Extended Idle Hours	Hours	0	0	0	...	20226.403
Brazoria 039	APU Hours	Hours	0	0	0	...	191.035
Brazoria 039	Starts	Starts	9389.986	4291769.18	12899.305	...	5819026.408
Brazoria 039	CO	POUNDS	231.305	282664.785	22.433	...	540283.701
Brazoria 039	NOx	POUNDS	8.395	27541.414	20.599	...	75298.916
Chambers 071	VMT	Miles	3054.066	3051187.79	3052.345	...	4699755.76
Chambers 071	VHT	Hours	54.772	54675.696	54.726	...	83816.735
Chambers 071	Speed	MPH	55.759	55.805	55.774	...	56.071
Chambers 071	SHP	Hours	406004.398	27009186.52	41119.752	...	43533373.28
Chambers 071	Extended Idle Hours	Hours	0	0	0	...	1226.403
Chambers 071	APU Hours	Hours	0	0	0	...	91.035
Chambers 071	Starts	Starts	8389.986	5291769.18	22899.305	...	7819026.408
Chambers 071	CO	POUNDS	123.0685	232756.330	18.714	...	472201.855
Chambers 071	NOx	POUNDS	5.262	22702.836	17.282	...	61328.545

**Figure 128. Sample SUT Tab-Delimited Output File.**

### Totals Tab-Delimited Output File (/TOUT)

The /TOUT identifier and its corresponding text string specify the path and filename of the totals tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity, and emissions), the data units, and 24-hour data for each tab-delimited EmsCalc output file. Figure 129 shows a sample totals tab-delimited output file.

TabFileTotals Tab-Delimited Output Header			
County	Data Type	UNITS	24 HOUR
Brazoria 039	VMT	Miles	9902656.72
Brazoria 039	VHT	Hours	235686.151
Brazoria 039	Speed	MPH	42.01628596
Brazoria 039	SHP	Hours	23533373.28
Brazoria 039	Extended Idle Hours	Hours	20226.4036
Brazoria 039	APU Hours	Hours	191.0357124
Brazoria 039	Starts	Starts	5819026.408
Brazoria 039	CO	POUNDS	540283.7013
Brazoria 039	NOx	POUNDS	75298.91621
Chambers 071	VMT	Miles	4699755.76
Chambers 071	VHT	Hours	83816.73556
Chambers 071	Speed	MPH	56.07180629
Chambers 071	SHP	Hours	43533373.28
Chambers 071	Extended Idle Hours	Hours	1226.4036
Chambers 071	APU Hours	Hours	91.0357124
Chambers 071	Starts	Starts	7819026.408
Chambers 071	CO	POUNDS	472201.8551
Chambers 071	NOx	POUNDS	61328.54571

**Figure 129. Sample Totals Tab-Delimited Output File.**

**Hourly Totals Tab-Delimited Output File (/THOUT)**

The /THOUT identifier and its corresponding text string specify the path and filename of the hourly totals tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity, and emissions), the data units, and hourly data for each tab-delimited EmsCalc output file. Figure 130 shows a sample hourly totals tab-delimited output file.

TabFileTotals Tab-Delimited Output Header							
County	Data Type	UNITS	Hour 1	Hour 2	...	Hour 24	24 HOUR
Brazoria 039	VMT	Miles	67289.91	45537.99	...	153475.94	9902656.72
Brazoria 039	VHT	Hours	1561.920	1056.753	...	3566.322	235686.151
Brazoria 039	Speed	MPH	43.081	43.092	...	43.0347	42.016
Brazoria 039	SHP	Hours	1007252.26	1009262.827	...	1002817.712	23533373.28
Brazoria 039	Extended Idle Hours	Hours	1738.177	2692.114	...	975.648	20226.403
Brazoria 039	APU Hours	Hours	16.416	25.426	...	9.214	191.035
Brazoria 039	Starts	Starts	34406.897	27489.564	...	74693.969	5819026.408
Brazoria 039	CO	POUNDS	2902.695	2643.154	...	7127.070	540283.701
Brazoria 039	NOx	POUNDS	1195.151	1112.640	...	1732.219	75298.916
Chambers 071	VMT	Miles	45243.25	30618.22	...	103191.8	4699755.76
Chambers 071	VHT	Hours	734.445	496.876	...	1677.772	83816.735
Chambers 071	Speed	MPH	61.601	61.621	...	61.505	56.0718
Chambers 071	SHP	Hours	1863272.557	1866991.818	...	1855069.278	43533373.28
Chambers 071	Extended Idle Hours	Hours	105.392	163.233	...	59.157	1226.403
Chambers 071	APU Hours	Hours	7.823	12.116	...	4.391	91.0357
Chambers 071	Starts	Starts	46232.551	36937.730	...	100366.294	7819026.408
Chambers 071	CO	POUNDS	2652.790	2479.412	...	6538.670	472201.855
Chambers 071	NOx	POUNDS	1144.966	1078.821	...	1617.144	61328.545

**Figure 130. Sample Hourly Totals Tab-Delimited Output File.**

**Hourly SUT Tab-Delimited Output File (/SHOUT)**

The /SHOUT identifier and its corresponding text string specify the path and filename of the hourly SUT tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity, and emissions), the data units, hour, and hourly SUT/fuel type combination data for each tab-delimited EmsCalc output file. Figure 131 shows a sample hourly SUT tab-delimited output file.

TabFileTotals Tab-Delimited Output Header							
County	Data Type	UNITS	Hour	MC_Gas	PC_Gas	...	TOTALS
Brazoria 039	VMT	Miles	Hour 1	47.8800678	47586.83777	...	67289.91
Brazoria 039	VMT	Miles	Hour 2	32.4025145	32204.07306	...	45537.99
....	....	....	....	....	....	...	....
Brazoria 039	VMT	Miles	Hour 24	109.2056625	108536.8582	...	153475.94
Brazoria 039	VMT	Miles	24 HOUR	7050.020771	7007533.624	...	9902656.72

**Figure 131. Sample Hourly SUT Tab-Delimited Output File.**

### Hourly Road Type Totals Tab-Delimited Output File (/TROUT)

The /TROUT identifier and its corresponding text string specify the path and filename of the hourly road type totals tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity, and emissions), the data units, road type, and hourly road type data for each tab-delimited EmsCalc output file. Figure 132 shows a sample hourly road type tab-delimited output file.

TabFileTotals Tab-Delimited Output Header							
County	Data Type	UNITS	Roadtype	Hour 1	...	Hour 24	24 HOUR
Brazoria 039	VMT	Miles	URBAN INTERSTATE	0	...	0	0
Brazoria 039	VMT	Miles	URBAN OTHER FREEWAY	0	...	0	0
Brazoria 039	VMT	Miles	TOLL ROADS	3425.59	...	7813.11	621839.42
Brazoria 039	VMT	Miles	RAMPS (FWY/TOLL/FRNT)	212.91	...	485.69	32576.76
....	....	....	....	....	....	...	....
Brazoria 039	VMT	Miles	LOCAL (INTRAZONAL)	397	...	905.47	51875.06
Brazoria 039	VMT	Miles	Off-Network	0	...	0	0
Brazoria 039	VMT	Miles	TOTALS	67289.91	...	153475.94	9902656.72

**Figure 132. Sample Hourly Road Type Tab-Delimited Output File.**

### Hourly SUT/Road Type Totals Tab-Delimited Output File (/SROUT)

The /SROUT identifier and its corresponding text string specify the path and filename of the hourly SUT/road type totals tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity, and emissions), the data units, hour, road type, and hourly SUT data by road type for each tab-delimited EmsCalc output file. Figure 133 shows a sample hourly SUT/road type tab-delimited output file.

TabFileTotals Tab-Delimited Output Header								
County	Data Type	UNITS	Hour	Roadtype	MC_Gas	PC_Gas	....	TOTALS
Brazoria 039	VMT	Miles	Hour 1	URBAN INTERSTATE	0	0	....	0
Brazoria 039	VMT	Miles	Hour 1	TOLL ROADS	2.4748138	2467.255993	....	3425.59
....	....	....	....	....	....	....	....	....
Brazoria 039	VMT	Miles	Hour 1	Off-Network	0	0	....	0
Brazoria 039	VMT	Miles	Hour 1	TOTALS	47.8800678	47586.83777	....	67289.91

**Figure 133. Sample Hourly SUT/Road Type Tab-Delimited Output File.**



### 24-hour SUT/Road Type Totals Tab-Delimited Output File (/DSROUT)

The /DSROUT identifier and its corresponding text string specify the path and filename of the 24-hour SUT/road type totals tab-delimited output file. This optional tab-delimited output file lists the county, data type (VMT, VHT, speed, off-network activity, and emissions), the data units, hour, road type, and hourly SUT data by road type for each tab-delimited EmsCalc output file. Figure 134 shows a sample 24-hour SUT/road type tab-delimited output file.

TabFileTotals Tab-Delimited Output Header							
County	Data Type	UNITS	Roadtype	MC_Gas	PC_Gas	....	TOTALS
Brazoria 039	VMT	Miles	URBAN INTERSTATE	0	0	....	0
Brazoria 039	VMT	Miles	TOLL ROADS	2.4748138	2467.255993	....	3425.59
....	....	....	....	....	....	....	....
Brazoria 039	VMT	Miles	Off-Network	0	0	....	0
Brazoria 039	VMT	Miles	TOTALS	47.8800678	47586.83777	....	67289.91

**Figure 134. Sample 24-hour SUT/Road Type Tab-Delimited Output File.**



## MISCELLANEOUS UTILITIES



## MOVESACTIVITYINPUTBUILD: MOVES ACTIVITY INPUT BUILDER

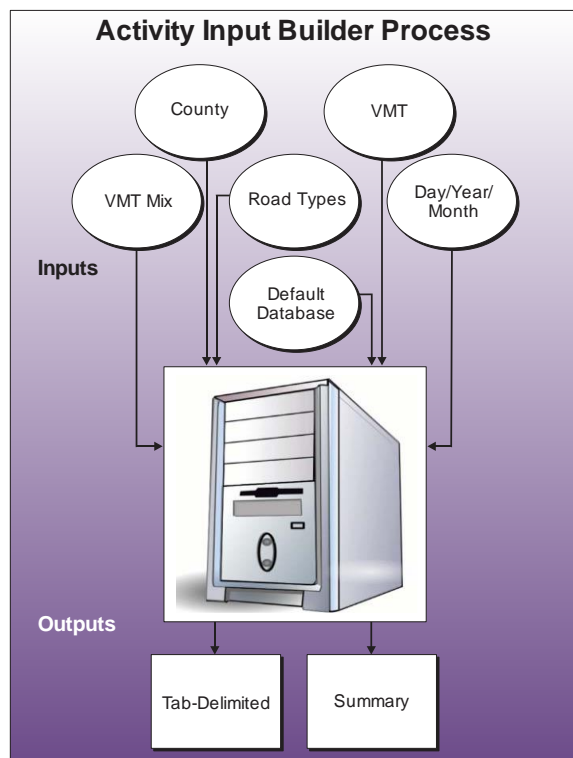
The MOVESactivityInputBuild utility assembles data into tab-delimited text files with formats consistent with MOVES database tables for the following tables: avgspeddistribution, hourvmtfraction, dayvmtfraction, hpmsvtypeday, monthvmtfraction, roadtype, roadtypedistribution, state, year, zone, zoneroadtype, sourcetypeofdayvmt, sourcetypeage starts, and hotelling. The utility also produces tab-delimited VMT summary (by hour, link road type, and link area type) and a VHT summary (by hour, link road type, link area type, and MOVES average speed bin ID) files.

### OPERATION

The MOVESactivityInputBuild utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until finding the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until finding the next step, saving the path and filename of all of the required inputs.

If any of the required input files are missing from the jcf, or MOVESactivityInputBuild cannot open any of these input files due to an incorrect path and/or filename or any other access error, then MOVESactivityInputBuild results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the county name input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). The utility ignores the first set of text (the text *COUNTY*), uses the second set of text (a numeric value) for the county number, and uses the sets of text for the county name, with a space separating each remaining set of text. The utility then writes the county name and county number to the listing output file.



The next input file the utility processes is the VMT listing input file. The utility reads each line of the file, checking the data in the first column. If the data in the first column is the asterisk symbol (\*), then the utility recognizes the line as a comment and ignores the line. Otherwise, the utility recognizes the line as a data line. For each data line in the input file, the utility ignores the first set of text (the text *Hour*) and saves the hourly VMT file (the third set of text) in part 1 of the hourly inputs array based on the file's respective hour (the second set of text).

The utility then sets the internal MOVES base data arrays, which the utility uses in the calculation and output process. The MOVES base data arrays contains the all of the MOVES hour ID, road type ID, pollutant ID, process ID, average speed bin ID, source type ID, and fuel type ID values. The MOVES base data arrays also contain abbreviations for the MOVES descriptions of these values. For the MOVES hour ID values (which range from 1 to 24), the abbreviations are set as the MOVES hour ID values. Appendix A shows the descriptions used in this utility for the other MOVES ID values.

The next input file the utility processes is the VMT road type designations input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMT\_RDCODE*, *VMT\_ATCODE*, *VMTMIX\_RDCODE*, *RATES\_RDCODE*, and *MOVES\_RDCODE* (capitalization does not matter) and sets the location of the VMT road type code, VMT area type code, VMT mix road type, emissions rate road type code, and MOVES road type code data, respectively. The utility then reads each remaining line and saves the VMT mix road type code, rates road type code, and MOVES road type code in the VMT road type designations array based on the VMT road type and VMT area type.

The next input file the utility processes is the VMT road type/ramp designation input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *VMT\_RDCODE*, *VMT\_ATCODE*, and *RAMP\_DESIG*, and sets the location of the VMT road type code, VMT area type code, and ramp designation data, respectively. The utility then reads each remaining line and saves the ramp designation (either *Yes* for ramp or *No* for non-ramp) in the VMT roadtype/ramp array based on the VMT road type and VMT area type.

If the time period VMT mix is found in the jcf, the utility then processes the time period designation input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *TIMEPER* and *TIMEPERIOD* (capitalization does not matter) and sets the location of the hour and time period data, respectively. The utility then reads each remaining line and saves the time period (must be AM Peak, Midday, PM Peak, or Overnight) in the time periods array based on the hour.

The next input file the utility processes is either the 24-hour VMT mix input file or the AM Peak VMT mix input file. If the time period designation input file is found in the jcf, then the utility processes the AM peak VMT mix file. Otherwise, the utility processes the 24-hour VMT mix input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the

header line consisting of the text *VMX\_RDCODE*, *MOVES\_STCODE*, *MOVES\_FTCODE*, and *VMTMIX* and sets the location of the VMT mix road type code, MOVES SUT code, MOVES fuel type code, and VMT mix data, respectively. For each remaining line, the utility saves the VMT mix in the first section of the VMT mix array based on the VMT mix road type code and MOVES SUT code. The utility repeats this process until processing all of the data lines. The utility also calculates the total of the VMT mix for each SUT across all of the MOVES road types in the VMT mix input file (i.e., the all road types VMT mix by SUT) and saves this data in the VMT mix array as well.

If the time period designation input file is found in the jcf, the utility also processes the Midday, PM Peak, and Overnight VMT mix input files. Using the same process as the AM Peak VMT mix input file, the utility processes each of the remaining VMT mix input files, saving the Midday VMT mix data in section 2 of the VMT mix array, the PM Peak data in section 3 of the mix array, and the Overnight VMT mix data in section 4 of the VMT mix array.

The next input the utility processes is the day ID input. If the identifier for the *hourvmtfraction*, *averagespeeddistribution*, or *dayvmtfraction* output is in the jcf, then this input is required. Only the appropriate *dayID* (5 or 2) is required following the day ID input identifier in the jcf (i.e., there is no input file). The next input the utility processes is the year ID input. If the identifier for the *hpsmvtyperedayoutput* is in the jcf, then this input is required. Only the appropriate *yearID* is required following the year ID input identifier in the jcf (i.e., there is no input file).

The next input the utility processes is the month ID input. If the identifier for the *monthvmtfraction* output is in the jcf, then this input is required. Only the appropriate *monthID* (value between 1 and 12) is required following the month ID input identifier in the jcf (i.e., there is no input file).

The next input file the utility processes is the FIPS input. If the identifier for the *zone*, *zoneroadtype*, *starts*, or *hotellinghours* output is in the jcf, then this input is required. Only the appropriate FIPS (or MOVES *countyID*) is required following the FIPS input identifier in the jcf (i.e., there is no input file).

The utility then processes the *sourcetypeyear* input file. If the identifier for the *sourcetypepage* or *hotellinghours* output is found in the jcf, then this input is required. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *yearID*, *sourceTypeID*, and *sourceTypePopulation*, and sets the location of the *yearID*, *sourceTypeID*, and vehicle population data, respectively. The utility then reads each remaining line and saves the vehicle population in the vehicle population array based on the *sourceTypeID* when the *yearID* from the line is equal to the *yearID* input.

The utility then processes the *sourcetypeagedistribuiton* input file. If the identifier for the *sourcetypepage*, *starts*, or *hotellinghours* output is found in the jcf, then this input is required. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting

of the text *sourceTypeID*, *yearID*, *ageID*, and *ageFraction*, and sets the location of the *sourceTypeID*, *yearID*, *ageID* and *age fraction* data, respectively. The utility then reads each remaining line and saves the *age fraction* data in the SUT *age distribution* array based on the *sourceTypeID* and *ageID* when the *yearID* from the line is equal to the *yearID* input.

The utility then processes the *sourcetypeage* input file. If the identifier for the *sourcetypeage* or *hotellinghours* output is found in the jcf, then this input is required. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *ageID*, *sourceTypeID*, *survivalRate*, *relativeMAR*, *functioningACFraction*, and *functioningACFractionCV* and sets the location of the *ageID*, *sourceTypeID*, *survivalRate*, *relativeMAR*, *functioningACFraction*, and *functioningACFractionCV* data, respectively. The utility then reads each remaining line and saves the *survivalRate*, *relativeMAR*, *functioningACFraction* and *functioningACFractionCV* data in the *sourcetypeage* input array based on the *sourceTypeID* and *ageID*.

The utility then processes the *starts* input file. If the identifier for the *starts* output is found in the jcf, then this input is required. This input file is the output (or equivalent format) from the *OffNetActCalc* utility. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, *St\_Ftdesc*, and *Starts* and sets the location of the *hourID*, *sourceTypeID*, and *starts* data, respectively (the other data is not required for this utility). The utility then reads each remaining line and saves the *starts* data in the *starts* input array based on the *hourID* and *sourceTypeID*. The utility then processes the *hotelling hours* input file. If the identifier for the *hotelling hours* output is found in the jcf, then this input is required. This input file is the output (or equivalent format) from the *OffNetActCalc* utility. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicating a comment line). This line is the header line consisting of the text *hourID*, *sourceTypeID*, *fuelTypeID*, *St\_Ftdesc*, *Hotelling*, *ExtIdle*, and *APU* and sets the location of the *hourID*, *sourceTypeID*, and *hotelling* data, respectively (the other data is not required for this utility). The utility then reads each remaining line and saves the *hotelling* data in the *hotelling* input array based on the *hourID* and *sourceTypeID*.

The utility then processes the *MySQL connection inputs* file. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” sign as the *MySQL connection port number*, *username*, and *password*, respectively. The only required input is the *port number*. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to *MySQL* using the inputs from the *MySQL connection inputs* file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

The utility then begins the calculation process. For each hourly *VMT* file in Part 1 of the *hourly inputs* array, the utility reads each line of the *hourly VMT* file and compares the *county number* from the input line (segment three of the line) to the *county number* from the *county name* input file. If these *county numbers* are not the same, then the utility skips this data line.



Otherwise, the utility then sets temporary values for the link road type (segment four of the line), link area type (segment nine of the line), link speed (segment six of the line), link VMT (segment seven of the line), and link VHT (calculated as link VMT/link speed). The utility adds this link VMT to the link data VMT summary array based on the hour being processed, link road type, and link area type. The utility adds the link VHT to the link data VHT summary array based on the hour being processed, link road type, link area type, and the average speed bin ID (determined using the link speed and the MOVES average speed bin descriptions, see Appendix A).

The utility also adds the link VHT to Part 2 of the road type VHT array based on the MOVES road type (determined using the link road type, link area type, and the VMT road type designations array). If the link is designated as a ramp (determined using the link road type, link area type, and the VMT roadtype/ramp array), then the utility also saves the link VHT in Part 1 in of the road type VHT array. The utility then calculates the 24-hour total VMT in the link data VMT summary array by link road type and link area type. The utility also calculates the hourly total VHT in the link data VHT summary by hour link road type and link area type. Additionally, the utility calculates a 24-hour total VHT as well by link road type, link area type, and average speed bin ID.

The utility then creates the MOVES road type VMT array (based on the MOVES road types), the hour VMT array (based on the SUTs in the VMT mix input file, the MOVES road types, and hour), and the average speed distribution array (based on the SUTs in the VMT mix input file, the MOVES road types, hour, and the MOVES average speed bins). For each link road type /link area type combination in the link data VMT summary array, the utility adds the 24-hour VMT from the link data VMT summary array to the MOVES road type VMT array based on the MOVES road type (determined using the link road type, link area type, and the VMT road type designations array). For each hour, link road type, and link area type combination in the link data VMT summary array, the utility applies the VMT mix from the VMT mix array (either the 24-hour VMT mix or the appropriate time period VMT as assigned by the time period designation array) to the VMT from the link data VMT summary array; thus distributing the VMT to each of the SUTs in the VMT mix input file. The utility then adds this SUT VMT to the hour VMT array.

The utility maintains a 24-hour total by SUT and MOVES road type as well. For each hour, link road type, link area type, and average speed bin ID combination in the link data VHT summary array, the utility applies the appropriate road type VMT mix (determined using the link road type, link area type and the road type designations data) from the VMT mix array (either the 24-hour VMT mix or the appropriate time period VMT as assigned by the time period designation array) to the VHT from the link data VHT summary array; thus distributing the VHT to each of the SUTs in the VMT mix input file. The utility adds this SUT VHT to the average speed distribution array and maintains a total VHT for all average speed bins by SUT, MOVES road type, and hour.

The utility then begins the process of extracting the MOVES defaults for each SUT where the all road types VMT mix is equal to 0. For each SUT where the all road types VMT mix in the VMT mix array is equal to 0, the utility uses the appropriate MySQL scripts to extract the road

type distribution data from the MOVES default roadtypedistribution database table and saves this data in the MOVES default road type distribution array. The utility extracts the average speed distribution data and hour VMT fraction data from the MOVES default avgspeddistribution and hourvmtfraction database tables in a similar manner and saves this data in the MOVES default average speed distribution array and MOVES default hour VMT fractions array, respectively.

The utility then writes the roadtypedistribution output. If the tab-delimited roadtypedistribution output is in the jcf, the utility creates the road type distribution array (based on the SUTs in the VMT mix input file and the MOVES road types). For each SUT input in the VMT mix input file, the utility aggregates the road type VMT from the hour VMT array by hour, saves the data in the road type distribution array (i.e., one VMT value for each road type), and converts the data to a distribution by dividing the road type VMT by the total VMT.

If the text for the tab-delimited roadtypedistribution output is in the jcf, the utility opens the output file and writes the header line consisting of the text *sourceTypeID*, *roadTypeID*, and *roadTypeVMTFraction*, with each set of text separated by a tab. For each SUT in the VMT mix input file, the utility writes the data for MOVES road type 1, with the roadTypeVMTFraction equal to 1. The utility then checks the all road types VMT mix from the VMT mix array. If this all road types VMT mix is greater than 0, the utility then writes the SUT, MOVES roadtype, and the road type VMT fractions for each road type from the road type distribution array for each MOVES roadtype. Otherwise, the utility writes the SUT, MOVES roadtype, and the road type VMT fractions from the MOVES default road type distribution array.

The utility then writes the hourvmtfraction output. If the tab-delimited hourvmtfraction output is in the jcf, the utility creates the hourly VMT fraction array (based on the SUTs in the VMT mix input file, the MOVES road types, and hour). For the first SUT input in the VMT mix input file, the VMT fractions in the VMT fraction array for MOVES road type one are set to a value of 1 for the first hour and 0 for the remaining 24 hours. Then, for each MOVES road type, the utility converts the VMT from the hour VMT array to a distribution and saves this data in the hourly VMT fraction array. The utility repeats this process for each SUT input in the VMT mix input file; thus creating an hourly VMT distribution for each SUT/MOVES road type combination.

If the text for the tab-delimited hourvmtfraction output is in the jcf, the utility opens the output file and writes the header line that consists of the text *sourceTypeID*, *roadTypeID*, *dayID*, *hourID*, and *hourVMTFraction*, with each set of text separated by a tab. For each SUT in the VMT mix input file, the utility checks the all road types VMT mix from the VMT mix array. If this all road types VMT mix is greater than 0, then the utility writes the SUT, MOVES road type, the day ID (from the day ID input), hour, and the hourly VMT fractions (from the hourly VMT fraction array) for each MOVES road type. Otherwise, the utility writes the data in a similar manner except the hourly VMT fractions are from the MOVES default hour VMT fractions array.

The utility then writes the avgspeddistribution output. If the tab-delimited avgspeddistribution output is in the jcf, the utility checks the all road types VMT mix from the VMT mix array for each SUT in the VMT mix input file. If this all road types VMT mix is

greater than 0, then the utility calculates the average speed distribution by MOVES road type and hour. If the MOVES road type is 1, then the utility sets the average speed distribution to a value of 1 for the first speed bin and 0 for the remaining speed bins. Otherwise, the utility converts the VHT in the average speed distribution array to a distribution of VHT over the 16 MOVES speed bins by SUT, MOVES road type, and hour.

If the text for the tab-delimited *avgspeeddistribution* output is in the jcf, the utility opens the output file and writes the header line that consists of the text *sourceTypeID*, *roadTypeID*, *hourDayID*, *avgSpeedBinID*, and *avgSpeedFraction*, with each set of text separated by a tab. For each SUT in the VMT mix input file, the utility checks the all road types VMT mix from the VMT mix array. If this all road types VMT mix is greater than 0, then the utility writes the SUT, the MOVES road type, the hour/day ID (combination of the hour and the day ID from the day ID input), the average speed bin ID, and the average speed fraction (from the average speed distribution array) for each average speed bin. The utility repeats this process for each hour and MOVES road type. If the all road types VMT mix is equal to 0, the utility writes the data in a similar manner except the hourly VMT fractions are taken from the MOVES default average speed distribution array.

The utility then writes the roadtype output. If the tab-delimited roadtype output is in the jcf, the utility converts the VHT in Part 1 of the road type VHT array to a fraction of the road type VHT attributable to ramps by dividing the ramp VHT (Part 1 of the road type VHT array) by the total road type VHT (Part 2 of the road type VHT array) for each road type.

If the text for the tab-delimited roadtype output is in the jcf, the utility opens the output file and writes the header line that consists of the text *roadTypeID*, *roadDesc*, *rampFraction*, *isAffectedByOnroad*, *isAffectedByNonroad*, and *shouldDisplay*, with each set of text separated by a tab. For each MOVES road type, the utility writes the road type ID, the road type description, the ramp fraction (from Part 1 of the road type VHT array), the value of 1 (*isAffectedByOnroad* value automatically set by the utility), the value of 0 (*isAffectedByNonroad* value automatically set by the utility), and the value of 1 (*shouldDisplay* value automatically set by the utility) to the output file. If MOVES road type 2 does have VMT, the utility also writes the MOVES ramp road types (roadTypeID 6 and 8, see Appendix A for road type descriptions) to the output file with the road type ID, the road type description, the ramp fraction (0 for roadTypeID 6, 1 for roadTypeID 8), the value of 1 (*isAffectedByOnroad* value automatically set by the utility), the value of 0 (*isAffectedByNonroad* value automatically set by the utility), and the value of 0 (*shouldDisplay* value automatically set by the utility) to the output file. If MOVES road type 4 does have VMT, the utility also writes the MOVES ramp road types (roadTypeID 7 and 9, see Appendix A for road type descriptions) to the output file with the road type ID, the road type description, the ramp fraction (0 for roadTypeID 7, 1 for roadTypeID 9), the value of 1 (*isAffectedByOnroad* value automatically set by the utility), the value of 0 (*isAffectedByNonroad* value automatically set by the utility), and the value of 0 (*shouldDisplay* value automatically set by the utility) to the output file.

The utility then writes the *hpmsvtypeday* output. If the tab-delimited *hpmsvtypeday* output is in the jcf, the utility aggregates the hourly VMT array by HPMS vehicle type (each SUT is

assigned an HPMS vehicle type, shown in Table 57), and saves this data in the HPMS vehicle type VMT array.

The utility then opens the output file and writes the header line that consists of the text *yearID*, *monthID*, *dayID*, *HPMSvtypeID*, and *VMT*, with each set of text separated by a tab. For each HPMS vehicle type, the utility writes the the year ID (from the year ID input), the monthID (from the monthID input), the dayID (from the dayID input), the HPMS vehicle type ID, and the HPMS vehicle type VMT (from the HPMS vehicle type VMT array), to the tab-delimited output file. Table 57 sows the SUT/HPMS vehicle type correlations.

**Table 57. SUT/HPMS Vehicle Type Correlations.**

HPMS Vehicle Type	MOVES SUT
10	11
25	21
	31
	32
40	41
	42
	43
50	51
	52
	53
	54
60	61
	62

The utility then writes the year output. If the tab-delimited output year output is in the jcf, the utility extracts the fuel year ID from the MOVES default year database table for the user-specified year (from the year ID input) using MySQL scripts.

If the text for the tab-delimited year output is in the jcf, the utility opens the output file and writes the header line that consists of the text *yearID*, *isBaseYear*, and *fuelYearID*, with each set of text separated by a tab. The utility then writes the year (from the year ID input), the isBaseYear data (automatically set to Y by the utility), and the fuel year ID to the output file.

The utility then writes the two summary output files — the tab-delimited link VMT summary output file and the tab-delimited link VHT summary output file. If the text for the tab-delimited link VMT summary output is in the jcf, the utility opens the output file and writes the header line that consists of the text *Hour*, *Roadtype*, *Area Type*, and *VMT*, with each set of text separated by a tab. The utility then writes the hour, link road type, link area type, and VMT from the link data VMT summary array for each hour, link road type, and link area type. The utility writes a 24-hour VMT total by link road type and link area type to this output file as well.

If the text for the tab-delimited link VHT summary output is in the jcf, the utility opens the output file and writes the header line that consists of the text *Hour*, *Roadtype*, *Area Type*, *avgSpeedBinID*, and *VHT*, with each set of text separated by a tab. The utility then writes the hour, link road type, link area type, average speed bin, and VHT from the link data VHT summary array for each hour, link road type, link area type, and average speed bin. The utility writes a 24-hour VHT total by link road type, link area type, and average speed bin to this output file as well.

If the text for the tab-delimited state output is in the jcf, then the utility uses the appropriate MySQL scripts to extract the MOVES default data from the state default database table (for stateID = 48) . This data is then written to the tab-delimited state output file.

If the text for the tab-delimited zone output is in the jcf, then the utility uses the appropriate MySQL scripts to extract the MOVES default data from the zone default database table for countyID > 48,000 and countyID < 49,000. This data, with the start allocation factors, idle allocation factors, and SHP allocation factors updated with values of 1, are written to the tab-delimited zone output.

If the text for the tab-delimited zoneroadtype output is in the jcf, then the utility uses the appropriate MySQL scripts to extract the MOVES default data from the zoneroadtype default database table (for zoneID > 480,000 and zoneID < 490,000). This data, with the SHO allocation factors updated with values of 1, are written to the tab-delimited zoneroadtype output.

If the text for the tab-delimited monthvmtfraction output is in the jcf, then the utility uses the appropriate MySQL scripts to extract the MOVES default data from the monthvmtfraction default database. The monthvmtfraction data, with the month VMT fractions updated to values of 1 for the user-specified month ID (from the month ID input) and 0 for all remaining months, are written to the tab-delimited monthvmtfraction output.

If the text for the tab-delimited dayvmtfraction output is in the jcf, then the utility uses the appropriate MySQL scripts to extract the MOVES default data from the dayvmtfraction default database. The dayvmtfraction data, with the day VMT fractions updated to values of 1 for the user-specified day ID (from the day ID input) and 0 for all remaining days, are written to the tab-delimited dayvmtfraction output.

The utility then writes the sourcetypedayvmt output. The utility aggregates the VMT from the hourly VMT array (contains VMT by sourceTypeID, roadTypeID, and hourID) by sourceTypeID and saves the sourceTypeID VMT in the sourcetypedayvmt array. The utility then

opens the tab-delimited output file and writes the header line consisting of the text *yearID*, *monthID*, *dayID*, *sourceTypeID*, and *VMT*, with each data segment separated by a tab. For each *sourceTypeID*, the utility writes the *yearID* (from the *yearID* input), *monthID* (from the *monthID* input), *dayID* (from the *dayID* input), the *sourceTypeID*, and the *VMT* from the *sourcetypeofdayvmt* array, with each data segment separated by a tab.

If the *sourcetypeage* or *hotellinghours* output is found in the *jcf*, the utility then calculates the adjusted relative mileage accumulation rates (*relativeMAR*). Using calculation procedures similar to MOVES, the utility calculates the travel fractions by SUT and age within each HPMS vehicle type using the vehicle population (*sourcetypeyear* input), the age distributions (*sourcetypeagedistribution* input), and the *relativeMAR* (*sourcetypeage* input). The utility then calculates the sum of the travel fractions within each HPMS vehicle type by SUT (proportion of travel for each SUT within the respective HPMS vehicle type). Using the SUT *VMT* array (based on the the hourly *VMT* and speeds files and the *VMT* mix), the utility calculates the fractions of *VMT* within each HPMS vehicle type by SUT. The utility then calculates the *relativeMAR* adjustment factors for each SUT by dividing the SUT *VMT* array-based fractions by the travel fraction-based fractions. The utility then calculates the adjusted *relativeMAR* by applying the *relativeMAR* adjustment factors to the input *relativeMAR* (from the *sourcetypeage* input). If the *sourcetypeage* output is found in the *jcf*, the utility then writes the data from the *sourcetypeage* input to the *sourcetypeage* output with the new adjusted *relativeMAR*.

If the *starts* output is found in the *jcf*, the utility then calculates the *starts* by *ageID* by multiplying the *starts* input array by the age distributions (*sourcetypeagedistribution* input) and saving the *starts* by hour, *sourceTypeID*, and *ageID* in the *starts* output array. The utility then writes the *starts* output in a format consistent with the MOVES *starts* table using the *starts* output array. If the *hoteling hours* output is found in the *jcf*, the utility then calculates the *hoteling hours* by *hourID* and *ageID* for SUT 62 by multiplying the *hoteling hours* input array by SUT 62 specific travel fractions. The travel fractions are calculated by multiplying the SUT age distribution for SUT 62 by the adjusted *relativeMAR* and normalizing these fractions across all *ageID* categories. The utility then writes the *hoteling hours* output in a format consistent with the MOVES *hoteling* table using the *hoteling hours* by *hourID* and *ageID*.

## JCF INPUT FILE

The *jcf* input file directs the *MOVESactivityInputBuild* utility to each of the input and output files. Each line of the *jcf* input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (*MOVESactivityInputBuild* in this case), or the path and filename of the input or output file. Table 58 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the *jcf* input file and this does not affect the execution of the utility.



**Table 58. MOVESactivityInputBuild File Type Identifiers.**

<b>Identifier</b>	<b>Description</b>	<b>Text String Description</b>
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/CTY	County Name Input File	Path and Filename
/VMT	VMT Listing Input File	Path and Filename
/RDTD	VMT Road Type Designations Input File	Path and Filename
/VMX	VMT Mix Input File <sup>2</sup>	Path and Filename
/TMPD	Time Period Designation Input File <sup>3</sup>	Path and Filename
/VMXA, /VMXM, /VMXP, /VMXO	AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files, <sup>3</sup> respectively	Path and Filename
/DAYID, /YRID, /MONID	ID Inputs – Day ID, Year ID, and MonthID, respectively	Values
/RMPRD	VMT Road Type/Ramp Designation Input File	Path and Filename
/MVSDB	MOVES Default Database	Database Name
/MYSQL	MySQL Connection Inputs File	Path and Filename
/FIPS	FIPS (MOVES countyID) Input <sup>4</sup>	Value
/STYI	sourcetypeyear Input File <sup>4</sup>	Path and Filename
/STADI	sourcetypeagedistribution Input File <sup>4</sup>	Path and Filename
/STAI	sourcetypeage Input File <sup>4</sup>	Path and Filename
/STRTI	Starts Input File <sup>4</sup>	Path and Filename
/HOTI	Hotelling Input File <sup>4</sup>	Path and Filename
/TRDTD	Tab-Delimited roadtypedistribution Output	Path and Filename
/THRVF	Tab-Delimited hourvmtfraction Output	Path and Filename
/TASPD	Tab-Delimited avgspeeddistribution Output	Path and Filename
/TRDTT	Tab-Delimited roadtype Output	Path and Filename
/TLVMT, /TLVHT	Tab-Delimited Summary Output Files – VMT and VHT, respectively	Path and Filename
/TYEAR	Tab-Delimited year Output	Path and Filename
/TSTAT	Tab-Delimited state Output <sup>4</sup>	Path and Filename
/TZONE	Tab-Delimited zone Output <sup>4</sup>	Path and Filename

**Table 58. MOVESactivityInputBuild File Type Identifiers (Continued).**

<b>Identifier</b>	<b>Description</b>	<b>Text String Description</b>
/TZRDT	Tab-Delimited zoneroadtype Output <sup>4</sup>	Path and Filename
/TMVF	Tab-Delimited monthvmtfraction Output <sup>4</sup>	Path and Filename
/TDVF	Tab-Delimited dayvmtfraction Output <sup>4</sup>	Path and Filename
/THVDY	Tab-Delimited hpmsvtypeday Output <sup>4</sup>	Path and Filename
/TSTDV	Tab-delimited sourcetypeofdayvmt Output <sup>4</sup>	Path and Filename
/TSTAO	Tab-Delimited sourcetypeage Output <sup>4</sup>	Path and Filename
/STRTO	Tab-Delimited starts Output <sup>4</sup>	Path and Filename
/HHTO	Tab-Delimited Hotelling Output <sup>4</sup>	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Only if the 24-hour VMT mix is to be used by the utility.

<sup>3</sup> Only if the time period VMT mix is to be used by the utility.

<sup>4</sup> Optional.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (MOVESactivityInputBuild), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 135 shows a sample jcf.



/JOB	MOVESactivityInputBuild Sample jcf
/STEP	Step1 MOVESactivityInputBuild
/LST	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk.lst
/CTY	C:\MOVESactivityInputBuild\Input\Brazoria.CTY
/VMT	C:\MOVESactivityInputBuild\Input\TDM_LINKINPUTFILESLIST.TXT
/RDTD	C:\MOVESactivityInputBuild\Input\HGB_RdType_MOVESdesig.txt
/VMX	C:\MOVESactivityInputBuild\Input\SUTmix\HoustonDist_VMTmix_w.txt
/DAYID	5
/YRID	2018
/MONID	7
/RMPRD	C:\MOVESactivityInputBuild\Input\HGB_RdType_RampDesig.txt
/MVSDB	movesdb20141021
/MYSQL	C:\MOVESactivityInputBuild\Input\MySQL_ConnInputs_MOVES2014.txt
/FIPS	48039
/STYI	C:\MOVESactivityInputBuild\Input\sourcetypeyear_Input.tab
/STADI	C:\MOVESactivityInputBuild\Input\sourcetypeagedist_Input.tab
/STAI	C:\MOVESactivityInputBuild\Input\sourcetypeyear_Input.tab
/STRTI	C:\OffNetActCalc\Output\hgb1hr_48039_OffNetActCalc_2018swkd_Starts.tab
/HOTI	C:\OffNetActCalc\Output\hgb1hr_48039_OffNetActCalc_2018swkd_ExtIdle.tab
/TRDTD	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_rtdist.tab
/THRVF	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_hrvmtfract.tab
/TASPD	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_avgspddist.tab
/TRDTT	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_roadtypetable.tab
/TLVMT	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_linkVMTsumm.tab
/TLVHT	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_linkVHTsumm.tab
/TYEAR	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_year.tab
/TSTAT	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_state.tab
/TZONE	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_zone.tab
/TZRDT	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_zoneroadtype.tab
/TMVF	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_monthvmtfract.tab
/TDVF	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_dayvmtfract.tab
/THVDY	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_hpmsvtypeday.tab
/TSTDV	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_sourcypedayvmt.tab
/TSTAO	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_sourcetypepage.tab
/STRTO	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_starts.tab
/HHTO	C:\MOVESactivityInputBuild\Output\Wk\MOVESactInBld_Wk_hotellinghrstable.tab

**Figure 135. Sample MOVESactivityInputBuild jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. This utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The MOVESactivityInputBuild utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step

name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (MOVESactivityInputBuild in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of all input files (some may include contents as well);
- The name of the MOVES default database;
- The path and filename of each hourly VMT and speeds file as it is being processed;
- The path and filename or database name of all output files;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **County Name Input File (/CTY)**

The /CTY identifier (required) and its corresponding text string specify the path and filename of the county name input file. The county name input file specifies the county code within the hourly VMT and speeds files for generating the output. This input file consists of a single line containing the identifier text *COUNTY*, the county number, the county name, and the county FIPS code in a three-digit format (i.e., 000). A tab, space, or any combination of tabs and spaces can separate all of the data within the input line. This input file is also the same input file used in the MOVESEmsCalc utility. Figure 136 shows a sample county input file.



```
COUNTY 1 Upshur 459
```

**Figure 136. Sample County Input File.**

### VMT Listing Input File (/VMT)

The /VMT identifier (required) and its corresponding text string specify the path and filename of the VMT listing input file. This input file consists of the identifier text *Hour*, the hour number (beginning with hour 1, which is 12:00-to-1:00 a.m.), and the path and filename for the respective hourly VMT and speeds file. The hourly VMT and speeds files in this input file can be the output from either the TRANSVMT or VirtualLinkVMT utilities. Figure 137 shows a sample VMT listing input file.

Hour 1	C:\MOVESactivityInputBuild\Input\VMT\upshur_2005fr_summer.T01
Hour 2	C:\MOVESactivityInputBuild\Input\VMT\upshur_2005fr_summer.T02
Hour 3	C:\MOVESactivityInputBuild\Input\VMT\upshur_2005fr_summer.T03

**Figure 137. Sample VMT and Speeds Listing Input File.**

### VMT Road Type Designations Input File (/RDTD)

The /RDTD identifier (required) and its corresponding text string specify the path and filename of the VMT road type designations input file that correlates each link road type/area type combination to a VMT mix road type, a rates road type, and a MOVES road type. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 59 shows the contents of the VMT road type designations input file header line.

**Table 59. VMT Road Type Designations Input File Header Line.**

Header Line Text	Description
VMT_RDcode	Link VMT and Speeds Road Type Code
VMT_ATcode	Link VMT and Speeds Area Type Code
VMTMIX_RDcode	VMT Mix Road Type Code
RATES_RDcode	Rates Road Type Code
MOVES_RDcode	MOVES Road Type Code
VMT_RDdesc	Link VMT and Speeds Road Type Code Description <sup>1</sup>
VMT_ATdesc	Link VMT and Speeds Area Type Code Description <sup>1</sup>
MOVES_RDdesc	MOVES Road Type Code Description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes data.

Following the header line, the user must list each link road type code/area type code combination (each on a separate line) with the combinations' VMT mix road type code, rates road type code, and MOVES road type code separated by any combination of tabs and/or spaces. Figure 138 shows a sample VMT road type/MOVES road type input file.

* NOTE: Lists the roadtype and area codes in the VMT input files and their road type designations.							
* NOTE: Descriptions (VMT_RDdesc, VMT_ATdesc, MOVES_RDdesc) are not used and are optional.							
VMT_RDcode	VMT_ATcode	VMTMIX_RDcode	RATES_RDcode	MOVES_RDcode	VMT_RDdesc	VMT_ATdesc	MOVES_RDdesc
0	3	3	3	3	RUR IH FWY 110	RUR	Rural_Unrest
2	3	2	2	2	RUR OTH PRIN ART 130	RUR	Rural_Rest
3	3	2	2	2	RUR MINOR ART 150	RUR	Rural_Rest

**Figure 138. Sample VMT Road Type Designations Input File.**

### 24-Hour VMT Mix Input File (/VMX)

The /VMX identifier (optional, only if the 24-hour VMT mix is to be used) and its corresponding text string specify the path and filename of the 24-hour VMT mix input file that specifies the analysis 24-hour mix of vehicles by SUT/fuel type combination by road type, with codes 2 through 9 being the MOVES roadway types (see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). This input file is optional and should only be used if the user intends for the 24-hour VMT mix be used in the calculations. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 60 shows the contents of the 24-hour VMT mix input file header line.

**Table 60. 24-Hour VMT Mix Input File Header Line.**

Header Line Text	Description
VMX_RDcode	VMT Mix Road Type Code
MOVES_STcode	MOVES SUT Code
MOVES_FTcode	MOVES Fuel Type Code
VMTmix	VMT Mix Value
VMX_RDdesc	VMT Mix Road Type Code Description <sup>1</sup>
MOVES_STdesc	MOVES SUT Code Description <sup>1</sup>
MOVES_FTdesc	MOVES Fuel Type Code Description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes and VMT mix data.

Following the header line, the user must list each MOVES SUT/MOVES fuel type combination for each VMT mix road type (each on a separate line) with the combinations' associated VMT mix value, separated by any combination of tabs and/or spaces. The utility will calculate the emissions only for those MOVES SUT/MOVES fuel type combinations in this input file. Figure 139 shows a sample 24-Hour VMT mix input file.

* NOTE: Waco TxDOT District -- 2000 Friday						
* NOTE: Descriptions (VMX_RDdesc, MOVES_STdesc, MOVES_FTdesc) are not used and are optional.						
VMX_RDcode	MOVES_STcode	MOVES_FTcode	VMTmix	VMX_RDdesc	MOVES_STdesc	MOVES_FTdesc
2	11	1	0.00055	Rural_Rest	MC	Gas
2	21	1	0.56562	Rural_Rest	PC	Gas
2	21	2	0.00170	Rural_Rest	PC	Diesel

**Figure 139. Sample 24-Hour VMT Mix Input File.**

### Time Period Designation Input File (/TMPD)

The /TMPD identifier (optional, only if the time period VMT mix is to be used) and its corresponding text string specify the path and filename of the time period designation input file, which specifies which time period VMT mix (AM Peak, Midday, PM Peak, or Overnight) each hour will use in the calculations. This input file is optional and should only be used if the time period VMT mix input files are in the jcf. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 61 shows the contents of the time period designation input file header line.

**Table 61. Time Period Designation Input File Header Line.**

Header Line Text	Description
TIMEPER	Hour
TIMEPERIOD	Time Period

Following the header line, the user must list each hour (1, 2, 3, ..., 23, 24) with the hour's associated time period (AM Peak, Midday, PM Peak, or Overnight), separated by any combination of tabs and/or spaces. Figure 140 shows a sample VMT mix input file.

TIMEPER	TIMEPERIOD
01	Overnight
02	Overnight
03	Overnight
04	Overnight
05	Overnight
06	Overnight
07	AM Peak
08	AM Peak
09	AM Peak
10	Midday
11	Midday
12	Midday
13	Midday
14	Midday
15	Midday
16	Midday
17	PM Peak
18	PM Peak
19	PM Peak
20	Overnight
21	Overnight
22	Overnight
23	Overnight
24	Overnight

**Figure 140. Sample Time Period Designation Input File.**

**AM Peak, Midday, PM Peak, and Overnight VMT Mix Input Files (/VMXA, /VMXM, /VMXP, /VMXO)**

The /VMXA, /VMXM, /VMXP, and /VMXO identifiers (optional, only if the time period VMT mix is to be used) and corresponding text string specify the path and filename of the AM Peak, Midday, PM Peak, and Overnight VMT mix input files, respectively. These input files specify the analysis 24-hour mix of vehicles by SUT/fuel type combination by road type, with codes 2 through 9 being the MOVES roadway types (see Appendix A). Any additional road types must be included after the MOVES roadway types (i.e., codes after 9). This input file is optional and should only be used if the user intends for the time period VMT mix be used in the calculations. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. The format of the time period VMT mix input files is the same as the 24-hour VMT mix input file (Table 61 shows the contents of the header line and Figure 140 shows a sample of the input file).

### Day ID Input (/DAYID)

The /DAYID identifier (optional, only required if the hourvmtfraction, avgspeddistribution, or dayvmtfraction output is included in the jcf) and its corresponding value specify the day ID input that designates the MOVES day ID code for performing the calculations. Since this input is only a value (5 or 2 following the identifier), no input file is required.

### Year ID Input (/YRID)

The /YRID identifier (optional, only required if the hpmsvtypeday or year output is included in the jcf) and its corresponding value specify the year ID input that designates the year for performing the calculations. Since this input is only a value, no input file is required.

### Month ID Input (/MONID)

The /MONID identifier (optional, only required if the monthvmtfraction output is included in the jcf) and its corresponding value specify the month ID input that designates the MOVES month ID code for performing the calculations. Since this input is only a value (5 or 2 following the identifier), no input file is required.

### VMT Road Type/Ramp Designation Input File (/RMPRD)

The /RMPRD identifier (optional, only required if the roadtype output is included in the jcf) and its corresponding text string specify the path and filename of the VMT road type/ramp designation input file that designates each link road type/area type combination as either ramp or not ramp. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The first line of this input file, other than any optional comments, must be the header line that consists of specific text that the user can separate by any combination of tabs and/or spaces. Table 62 shows the contents of the VMT road type/ramp designation input file header line.

**Table 62. VMT Road Type/Ramp Designation Input File Header Line.**

Header Line Text	Description
VMT_RDcode	Link VMT and Speeds Road Type Code
VMT_ATcode	Link VMT and Speeds Area Type Code
RAMP_desig	Ramp Designation (“Yes” for ramp, “No” for not ramp)
VMT_RDdesc	Link VMT and Speeds Road Type Code Description <sup>1</sup>
VMT_ATdesc	Link VMT and Speeds Area Type Code Description <sup>1</sup>

<sup>1</sup> Optional, but must appear after codes and ramp designation data.

Following the header line, the user must list each link road type code/area type code combination (each on a separate line) with the combinations ramp designation (*Yes* for ramp, *No* for not ramp), separated by any combination of tabs and/or spaces. Figure 141 shows a sample VMT road type/ramp designation input file.

\* NOTE: Lists the roadtype and area codes in the VMT input files and their ramp designations.  
 \* NOTE: Descriptions (VMT\_RDdesc, VMT\_ATdesc) are not used and are optional.  
 \* NOTE: Desig "NO" indicates not ramp, "YES" indicates is ramp.

VMT_RDcode	VMT_ATcode	RAMP_desig	VMT_RDdesc	VMT_ATdesc
0	1	NO	RUR IH FWY 110	RUR
2	1	NO	RUR OTH PRIN ART 130	RUR
3	1	NO	RUR MINOR ART 150	RUR

**Figure 141. Sample VMT Road Type/Ramp Designation Input File.**

### MOVES Default Database (/MVSDB)

The /MVSDB identifier (required) and its corresponding text string specify the name of the database location for the MOVES default database tables. This database must be in the MySQL data folder. Only the name of the database is required.

### MySQL Connection Inputs File (/MYSQL)

The /MYSQL identifier (required) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 142 shows a sample MySQL connections input file.

\* Comments can be added by placing \* in first column.  
 \* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by any combination of tabs and/or spaces.  
 \* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any combination of tabs and/or spaces.  
 \* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.  
 \* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any combination of tabs and/or spaces.  
 \* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.  
 \* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.

```
Port Number = 3307
Username = moves
Password = moves
```

**Figure 142. Sample MySQL Connection Inputs File.**

### FIPS (MOVES countyID) Input (/FIPS)

The /FIPS identifier and its corresponding input value specify the county FIPS (or MOVES countyID) used in the calculations and output, where applicable. The FIPS code used as this



input is the five digit code and should be the same code used as the MOVES countyID for the county. Since this input consists of only a value, no path and/or filename are required.

**sourcetypeyear Input File (/STYI)**

The /STYI identifier and its corresponding text string specify the path and filename for the tab-delimited sourcetypeyear input file. This input file is the output from the VehPopulationBuild utility (or equivalent format) and is only required if the sourcetypeage output or houtellinghours output are in the jcf. Figure 143 shows a sample of the sourcetypeyear input file.

yearID	sourceTypeID	salesGrowthFactor	sourceTypePopulation	migrationrate
2011	11	1	7223	1
2011	21	1	134475	1
2011	31	1	64098.0571163603	1
2011	32	1	16628.9428836397	1
2011	41	1	201.673655423883	1
2011	42	1	83.1461561835308	1
2011	43	1	260.052871467639	1
2011	51	1	254.745670009116	1
2011	52	1	4829.55332725615	1
2011	53	1	932.29838954725	1
2011	54	1	192.828319659678	1
2011	61	1	922	1
2011	62	1	791.840217884703	1

**Figure 143. Sample sourcetypeyear Input File.**

**sourcetypeagedistribution Input File (/STADI)**

The /STADI identifier and its corresponding text string specify the path and filename for the tab-delimited sourcetypeagedistribution input file. This input file is the input to the MOVES run and is only required if the sourcetypeage output or houtellinghours output are in the jcf. Figure 144 shows a sample of the sourcetypeagedistribution input file.

sourceTypeID	yearID	ageID	ageFraction
11	2011	0	0.02948913
11	2011	1	0.035165
11	2011	2	0.08486778
11	2011	3	0.0977433
11	2011	4	0.11352623
11	2011	5	0.10217361
11	2011	6	0.07822234
11	2011	7	0.0595320
11	2011	8	0.0715769

**Figure 144. Sample sourcetypeagedistribution Input File.**

### **sourcetypeage Input File (/STAI)**

The /STAI identifier and its corresponding text string specify the path and filename for the tab-delimited sourcetypeage input file. This input file is the input to the MOVES run and is only required if the sourcetypeage output or houtellinghours output are in the jcf. Figure 145 shows a sample of the sourcetypeage input file.

ageID	sourceTypeID	survivalRate	relativeMAR	functioningACFraction	functioningACFractionCV
0	11	1	1	0	0
1	11	0.978999	0.533999979	0	0
2	11	0.939999	0.404000014	0	0
3	11	0.939999	0.333999991	0	0
4	11	0.939999	0.289000004	0	0

**Figure 145. Sample sourcetypeage Input File.**

### **Starts Input File (/STRTI)**

The /STRTI identifier and its corresponding text string specify the path and filename for the tab-delimited starts input file. This input file is the output (or equivalent format) from the OffNetActCalc utility and is only required if the starts output is in the jcf. Figure 146 shows a sample of the starts input file.

hourID	sourceTypeID	fuelTypeID	St_Ftdesc	Starts
1	11	1	MC_Gas	16.3124232
1	21	1	PC_Gas	3589.04559149853
1	21	2	PC_Diesel	14.4271935014657
1	31	1	PT_Gas	3031.1573613007
1	31	2	PT_Diesel	43.0880148913207

**Figure 146. Sample Starts Input File.**

### **Hotelling Input File (/HOTI)**

The /HOTI identifier and its corresponding text string specify the path and filename for the tab-delimited hotelling input file. This input file is the output (or equivalent format) from the OffNetActCalc utility and is only required if the hotelling output is in the jcf. Figure 147 shows a sample hoteling input file.

hourID	sourceTypeID	fuelTypeID	St_Ftdesc	Hotelling	ExtIdle	APU
1	62	2	CLhT_Diesel	15.454304	15.171462	0.282842
2	62	2	CLhT_Diesel	23.178552	22.754342	0.424210
3	62	2	CLhT_Diesel	24.853239	24.398379	0.454860
4	62	2	CLhT_Diesel	25.007922	24.550231	0.457691
5	62	2	CLhT_Diesel	13.183665	12.942380	0.241285

**Figure 147. Sample Hotelling Input File.**

### **roadtypedistribution Tab-Delimited Output (/TRDTD)**

The /TRDTD identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited roadtypedistribution output. This output file is optional and contains data consistent with the format of the MOVES default roadtypedistribution database table.

### **hourvmtfraction Tab-Delimited Output (/THRVF)**

The /THRVF identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited hourvmtfraction output. This output file is optional and contains the data consistent with the format of the MOVES default hourvmtfraction database table.

### **avgspeeddistribution Tab-Delimited Output (/TASPD)**

The /TASPD identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited avgspeeddistribution output. This output file is optional and contains the data consistent with the format of the MOVES default avgspeeddistribution table.

### **roadtype Tab-Delimited Output (/TRDTT)**

The /TRDTT identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited roadtype output. This output file is optional and contains the data consistent with the format of the MOVES default roadtype table.

### **Tab-Delimited Link VMT Summary Output File (/TLVMT)**

The /TLVMT identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited link VMT summary output file. This output file is optional and lists the link VMT by hour, link roadway type, and link area type.

### **Tab-Delimited Link VHT Summary Output File (/TLVHT)**

The /TLVHT identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited link VHT summary output file. This output file is optional and lists the link VHT by hour, link roadway type, link area type, and MOVES average speed bin ID.

### **year Tab-Delimited Output (/TYEAR)**

The /TYEAR identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited year output. This output file is optional and contains the data in a format consistent with the MOVES default year database table.

### **state Tab-Delimited Output (/TSTAT)**

The /TSTAT identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited state output. This output is optional and contains the data in a format consistent with the MOVES default state database table.

### **zone Tab-Delimited Output (/TZONE)**

The /TZONE identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited zone output. This output is optional and contains data in a format consistent with the MOVES default zone database table.

### **zoneroadtype Tab-Delimited Output (/TZRDT)**

The /TZRDT identifier (optional) and its corresponding text string specify the path and filename of the zoneroadtype output. This output is optional and contains data in a format consistent with the MOVES default zoneroadtype database table.

### **monthvmtfraction Tab-Delimited Output (/TMVF)**

The /TMVF identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited monthvmtfraction output. This output is optional and contains data in a format consistent with the MOVES default monthvmtfraction database table.

### **dayvmtfraction Tab-Delimited Output (/TDVF)**

The /TDVF identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited dayvmtfraction output. This output is optional and contains data in a format consistent with the MOVES default dayvmtfraction database table.

### **hpmsvtpeday Tab-Delimited Output (/THVDY)**

The /THVDY identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited hpmsvtpeday output. This output is optional and contains data in a format consistent with the MOVES hpmsvtpeday database table.

### **sourcetypedayvmt Tab-Delimited Output (/TSTDV)**

The /TSTDV identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited sourcetypedayvmt output. This output is optional and contains data in a format consistent with the MOVES sourcetypedayvmt database table.

### **sourcetypeage Tab-Delimited Output (/TSTAO)**

The /TSTAO identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited sourcetypeage output. This output is optional and contains data in a format consistent with the MOVES default sourcetypeage database table with the adjusted relativeMAR. The relativeMAR data are adjusted so that the travel fractions calculated by MOVES will reflect the 24-hour VMT mix from the calculations by the utility.

### **starts Tab-Delimited Output (/STRTO)**

The /STRTO identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited starts output. This output is optional and contains data in a format consistent with the MOVES starts database table.

### **Hotelling Hours Tab-DelimitedOutput (/HHTO)**

The /HHTO identifier (optional) and its corresponding text string specify the path and filename of the tab-delimited hoteling hours output. This output is optional and contains data in a format consistent with the MOVES hoteling hours database table.

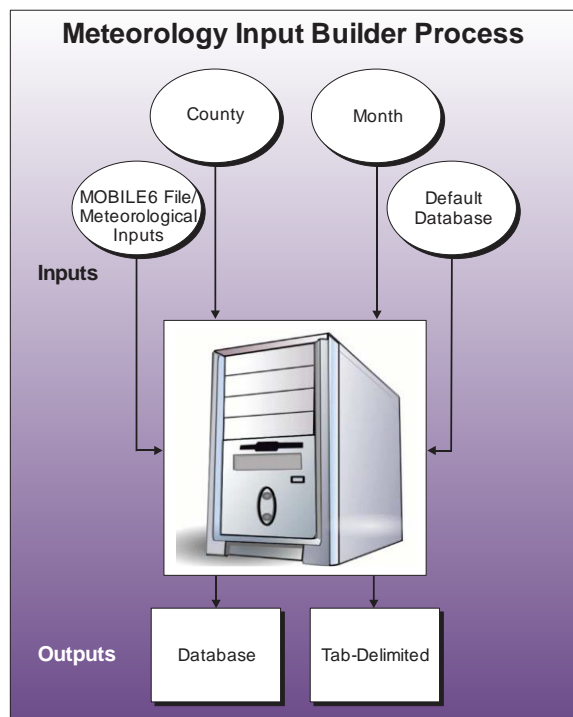


## MOVESMETINPUTBUILD: MOVES METEOROLOGY INPUT BUILDER

The MOVESmetInputBuild utility creates the zonemonthhour and county database tables in a format consistent with inputs to MOVES from either a MOBILE6 input file or a text file listing the meteorological data. A tab-delimited output format of these database tables is available as well.

### OPERATION

The MOVESmetInputBuild utility reads the jcf until it reaches the step to run. The utility writes each line of the desired step to the listing file designated in the desired step. After reaching the appropriate step, the utility then reads each line of the jcf until finding the listing file input line, thus determining the path and filename of the listing file. The utility then closes the jcf input file. The utility then re-opens the jcf input file. The utility reads each line of the jcf until finding the next step, saving the path and filename of all of the required inputs.



If any of the required input files are missing from the jcf, or MOVESmetInputBuild cannot open any of these input files due to an incorrect path and/or filename or any other access error, then MOVESmetInputBuild results in an error condition. In the event of an error condition, the utility writes an error to the console box and to the listing output file, terminates the execution of the current step, and returns control back to the main MOVES2014aUtl code to check the jcf for more steps to execute. The utility will also result in an error condition if any required data headers are not found or are incorrect. For more information on the requirements of the individual input files, see the input files descriptions following this section.

The first input file the utility processes is the county name input file. The utility reads each line, locating the first line that is not blank and does not have the asterisk symbol (\*) in the first column (indicates a comment line). For this line, the utility sets the county name (segment three through the next to last segment of the line, with each segment separated by a space) and the temporary FIPS code (numeric value in the last segment of the line). If the temporary FIPS code is a two-digit number, the utility combines the digits 480, the temporary FIPS code, and the digit 0 to form the zone ID (i.e., if the temporary FIPS code is 39, then the zone ID is 480390). If the county number is a three-digit number, the utility combines the digits 48, the temporary FIPS code, and the digit 0 to form the zone ID (i.e., if the temporary FIPS code is 139, then the zone ID is 481390). The utility also sets the county ID as the first five digits of the zone ID (i.e., if the zone ID is 480390, then the county ID is 48039).

The next input file the utility processes is the month ID input file. The utility reads each line, checking the data in the first column. If the data in the first column is the asterisk symbol (\*), the utility recognizes the line as a comment and ignores the line. Otherwise, the utility recognizes the line as a data line. For each data line in the input file, the utility checks the first word in the line (i.e., text prior to the equals symbol [=]) for the text *monthID* (text is not case sensitive). If the text is found, then the utility saves the numeric value following the equals symbol (=) as the monthID.

The utility then processes the meteorological data, which can be either a MOBILE6 input file or the meteorological data input file. Although one of these input files must be present in the jcf for the utility to operate properly, the user should not use both of these input files in the same step of a jcf. If the meteorological data is a MOBILE6 input file, the utility opens the MOBILE6 input file and reads each line checking for the text *HOURLY TEMPERATURES* (hourly temperatures data line), *RELATIVE HUMIDITY* (relative humidity data line), or *BAROMETRIC PRES* (barometric pressure data line) followed by the colon symbol (:).

For the hourly temperatures data line, the utility saves the 12 hourly temperatures from this data line in the hourly temperatures array beginning with hour seven (MOBILE6 inputs begin with the 6:00 a.m. hour). The utility then reads the next line, saves the first six hourly temperatures from this line in hours 19 through 24 of the hourly temperatures array and saves the last six hourly temperatures from this line in hours one through six of the hourly temperatures array.

For the relative humidity data line, the utility saves the 12 hourly temperatures from this data line in the relative humidity array beginning with hour seven (MOBILE6 inputs begin with the 6:00 a.m. hour). The utility then reads the next line, saves the first six hourly temperatures from this line in hours 19 through 24 of the relative humidity array and saves the last six hourly temperatures from this line in hours one through six of the relative humidity array.

For the barometric pressure data line, the utility sets the barometric pressure as the second segment of the barometric pressure data line. The utility also sets the altitude designation. If the barometric pressure is less than or equal to 25.8403, then the altitude designation is set as the text *H*. Otherwise, the altitude designation is set as the text *L*.

If the meteorological data is a meteorological data input file, the utility opens the meteorological data input file and reads each line checking for the text *HOURLY TEMPERATURES* (hourly temperatures data line), *RELATIVE HUMIDITY* (relative humidity data line), or *BAROMETRIC PRES* (barometric pressure data line) followed by the colon symbol (:).

For the hourly temperatures data line, the utility saves the hourly temperatures from this data line into the hourly temperatures array beginning at hour one (this input file begins with the 12:00 a.m. hour). If the utility has not saved all 24 hours in the hourly temperatures array, then the utility reads the next line and saves these hourly temperatures into the hourly temperatures array beginning with the first blank hour. The utility repeats this process until saving all the hourly temperatures for all 24 hours into the hourly temperatures array.



For the relative humidity data line, the utility saves the relative humidity from this data line into the relative humidity array beginning at hour one (this input file begins with the 12:00 a.m. hour). If the utility has not saved all 24 hours in the relative humidity array, then the utility reads the next line and saves the relative humidity values into the relative humidity array beginning with the first blank hour. The utility repeats this process until saving all the relative humidity for all 24 hours into the hourly temperatures array.

For the utility locates the barometric pressure data line, the utility sets the barometric pressure as the second segment of the barometric pressure data line. The utility also sets the altitude designation. If the barometric pressure is less than or equal to 25.8403, then the altitude designation is set as the text *H*. Otherwise, the altitude designation is set as the text *L*.

The first input file the utility processes is the MySQL connection inputs file. If any database output is to be created, this input file is required. The utility reads each line until locating the text *Port Number*, *Username*, or *Password* and saves the data after the “=” symbol as the MySQL connection port number, username, and password, respectively. The only required input is the port number. If this input is not found, then the utility terminates the current step execution and an error is written to the listing file. The utility then attempts to establish a connection to MySQL using the inputs from the MySQL connection inputs file. If this connection cannot be established, the current step execution is terminated and an error is written to the listing file.

The utility then writes the zonemonthhour database table and tab-delimited output file. If the text for the tab-delimited zonemonthhour output is in the jcf, the utility attempts to create the output file. If this output file does not exist, then the utility creates the output file and writes the header line that consists of the text *monthID*, *zoneID*, *hourID*, *temperature*, *temperatureCV*, *relHumidity*, *heatIndex*, *specificHumidity*, and *relativeHumidityCV*, with each set of text separated by a tab. Otherwise, the utility opens the output file and appends the data to the end of the file.

For each hour, the utility then writes the month ID (from the month ID input file), the zone ID, the hour, the temperature (from the hourly temperatures array), the temperatureCV data (automatically set to “null” by the utility), the relative humidity (from the relative humidity array), the heat index (automatically set to “null” by the utility), the specific humidity (automatically set to “null” by the utility), and the relativeHumidityCV data (automatically set to “null” by the utility).

If the text for the zonemonthhour database table output is in the jcf, then the utility uses the appropriate MySQL scripts to create a database table similar to the MOVES default zonemonthhour database table. If this database table already exists, then the utility writes a warning to the listing file and appends the data to the existing database table. Using MySQL scripts, the utility adds the data to the database table in the same manner as writing the tab-delimited output using the hourly temperatures array and the relative humidity array.

The utility then writes the county database table and tab-delimited output file. If this output file does not exist, then the utility creates the output file and writes the header line that consists of the text *countyID*, *stateID*, *countyName*, *altitude*, *GPAFract*, *barometricPressure*, and

*barometricPressureCV*, with each set of text separated by a tab. Otherwise, the utility opens the output file and appends the data to the end of the file. The utility then writes the county ID, the state ID (automatically set to 48 by the utility), the county name (from the county name input file), the altitude designation, the GPAFract data (automatically set to 0 by the utility), the barometric pressure, and the *barometricPressureCV* data (automatically set to “null” by the utility).

If the text for the county database table output is in the jcf, then the utility uses the appropriate MySQL scripts to create a database table similar to the MOVES default county database table. If this database table already exists, then the utility writes a warning to the listing file and appends the data to the existing database table. Using MySQL scripts, the utility adds the data to the database table in the same manner as adding the tab-delimited output.

The utility then creates the meteorological data summary. The utility calculates the minimum and maximum hourly temperatures and relative humidity values using the hourly temperatures array and the relative humidity array, respectively. The utility writes these minimum and maximum values, along with the barometric pressure, to the listing output file.

## **JCF INPUT FILE**

The jcf input file directs the MOVESmetInputBuild utility to each of the input and output files. Each line of the jcf input file consists of a file type identifier and a text string, which can either be a comment, the utility to run (MOVESmetInputBuild in this case), or the path and filename of the input or output file. Table 63 shows the file type identifiers. The utility does not recognize any other characters at the beginning of a line in the jcf input file and this does not affect the execution of the utility.

**Table 63. MOVESmetInputBuild File Type Identifiers.**

Identifier	Description	Text String Description
/JOB	Job Description <sup>1</sup>	Comment
/STEP	Step Description	Utility Name
/LST	Listing Output File	Path and Filename
/CTY	County Name Input File	Path and Filename
/MONID	Month ID Input File	Path and Filename
/M6IN	MOBILE6 Input File <sup>2</sup>	Path and Filename
/METIN	Meteorological Data Input File <sup>2</sup>	Path and Filename
/MVSDB	MOVES Default Database	Database Name
/MYSQL	MySQL Connection Inputs File	Path and Filename
/DZHM	zonemonthhour Output – Database Table	Database Name
/TZHM	zonemonthhour Output – Tab-Delimited File	Path and Filename
/DCTY	county Output – Database Table	Database Name
/TCTY	county Output – Tab-Delimited File	Path and Filename

<sup>1</sup> Only if the step is the first step in the jcf. This is only for reference purposes and not accessed by the utility.

<sup>2</sup> Optional. Only use one of the identifiers in a single step of a jcf.

The only format requirement for the input and output file lines in the jcf input file is that a space, a tab, or any combination of tabs and spaces occurs between the identifier and the text string. The step identifier (/STEP) however, has three components – the identifier (/STEP), a text string identifying the step (i.e., Step1), and the utility name (MOVESmetInputBuild), with each separated by a space, a tab, or any combination of tabs and spaces. Figure 148 shows a sample jcf.

```

/JOB      MOVESmetInputBuild Sample jcf
/STEP    Step1 MOVESmetInputBuild
/LST     C:\MOVESmetInputBuild\Output\MOVESmetInputBuild_Wk.lst
/CTY     C:\MOVESmetInputBuild\Input\Brazoria.CTY
/MONID   C:\MOVESmetInputBuild\Input\monthID_7.txt
/METIN   C:\MOVESmetInputBuild\Input\Braz_METIN.txt
/MVSDB   movesdb20141021
/MYSQL   C:\MOVESmetInputBuild\Input\MySQL_ConnInputs_MOVES2014.txt
/DZHM    MOVESmetInputBuild_Wk
/TZHM    C:\MOVESmetInputBuild\Output\MOVESmetInputBuild_zonemonthhour_Wk.tab
/DCTY    MOVESmetInputBuild_Wk
/TCTY    C:\MOVESmetInputBuild\Output\MOVESmetInputBuild_county_Wk.tab

```

**Figure 148. Sample MOVESmetInputBuild jcf.**

### **Job Identifier (/JOB)**

The /JOB identifier (optional) and its corresponding text string are comments used only as a description of the jcf. This utility does not use this identifier, but it is helpful for adding comments and a title to the jcf.

### **Step Identifier (/STEP)**

The /STEP identifier (required) and its corresponding text string name the utility to run. The MOVESmetInputBuild utility does not use this text string, but MOVES2014aUtl uses it to access the appropriate utility. The corresponding text string must contain two segments – a step name and the utility to run, with each part separated by a space, a tab, or any combination of tabs and spaces.

### **Listing File Output (/LST)**

The /LST identifier (required) and its corresponding text string specify the path and filename of the listing output file. If the path and/or filename are invalid, an error message appears on the screen and the utility will terminate. The listing file summarizes all the input and output for the specific jcf input file. The information in the listing file includes:

- Date and time of the beginning of the utility execution;
- The utility (MOVESmetInputBuild in this case) being executed;
- The MOVES2014aUtl compilation date;
- The path and filename of the jcf input file;
- JCF through the step being executed;
- The path and filename of all input files (some may include contents as well);
- The name of the MOVES default database;
- The path and filename or database name of all output files;
- A summary of the minimum and maximum values for the meteorological data;
- The total run time for the execution of the current step; and
- Date and time of the end of the utility execution.

### **County Name Input File (/CTY)**

The /CTY identifier (required) and its corresponding text string specify the path and filename of the county name input file. The county name input file specifies the county code within the hourly VMT and speeds files for generating the output. This input file consists of a single line containing the identifier text *COUNTY*, the county number, the county name, and the county FIPS code in a three-digit format (i.e., 000). A tab, space, or any combination of tabs and spaces can separate all of the data within the input line. This input file is also the same input file used in the MOVESEmsCalc utility. Figure 149 shows a sample county input file.

COUNTY 1 Upshur 459

**Figure 149. Sample County Input File.**

### **Month ID Input File (/MONID)**

The /MONID identifier (required) and its corresponding text string specify the path and filename of the month ID input file that specifies the MOVES month ID code for performing the calculations. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only input line, other than the optional comments, consists of the text *monthID* = followed by the MOVES month ID code, separated by any combination of tabs and/or spaces. Figure 150 shows a sample month ID input file.

\* Comments can be added by placing the symbol \* in first column.  
\* Format of input file is fixed, with monthID = followed by MOVES monthID code, separated by any  
\* combination of tabs and/or spaces.  
\* monthID text can be any combination of capital or lowercase letters.  
monthID = 7

**Figure 150. Sample Month ID Input File.**

### **MOBILE6 Input File (/M6IN)**

The /M6IN identifier (optional, only used if the output is to be based on this input file) and its corresponding text string specify the path and filename of the MOBILE6 input file, which is the same MOBILE6 input file used when running MOBILE6. This input file is optional. However, the text for either this input file or the meteorological data input file (/METIN) must appear in the jcf.

### **Meteorological Data Input File (/METIN)**

The /METIN identifier (optional, only used if the output is to be based on this input file) and its corresponding text string specify the path and filename of the meteorological data input file that lists the barometric pressure, hourly temperatures, and hourly relative humidity values used to build the output files. Although this input file is optional, the text for either this input file or the MOBILE6 input file (/M6IN) must appear in the jcf. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. This input file should contain three identifiers (Barometric Pressure, Hourly Temperatures, and Relative Humidity), followed by the colon symbol (:) and the respective data values, separated by any combination of tabs and/or spaces. The barometric pressure is a single value. However, the hourly temperatures and hourly relative humidity values are by hour, with the first hour being 12:00-to-1:00 a.m. Figure 151 shows a sample meteorological data input file.

```

* Comments can be added by placing the symbol * in the first column.
* Format of input file is fixed, with identifier (Barometric Pressure or Hourly Temperatures or
* Relative Humidity) followed by the symbol : and the appropriate data.
* For Hourly Temperatures and Relative Humidity, the first hour is 12:00-to-1:00 a.m.
Barometric Pressure: 29.91
Hourly Temperatures: 77.3 76.7 76.3 76.0 75.7 75.3 75.2 76.7 80.3 83.3 85.3 86.8
                        87.8 88.4 88.6 88.6 88.1 87.2 85.9 83.7 81.3 79.8 78.7 78.0
Relative Humidity: 86.8 88.2 88.8 89.0 89.3 89.7 89.6 85.1 76.2 68.2 61.7 56.9
                        53.7 52.0 52.8 52.6 54.1 57.0 61.2 67.9 75.2 80.1 83.3 85.3

```

**Figure 151. Sample Meteorological Data Input File.**

### **MOVES Default Database (/MVSDB)**

The /MVSDB identifier (optional, only used if the database output is to be created) and its corresponding text string specify the name of the database location of the MOVES default database tables. This database must be in the MySQL data folder. Only the name of the database is required.

### **MySQL Connection Inputs File (/MYSQL)**

The /MYSQL identifier (optional, only used if the database output is to be created) and its corresponding text string specify the path and filename of MySQL connection inputs file, which specifies the required data to connect to the appropriate version of MySQL. The user can add comments at any point in this input file by placing the asterisk symbol (\*) in the first column. The only required line is the data identifier “Port Number”, followed by the equals symbol (=) and the port number for the MySQL connection. Using a similar format, the user may also include the data identifiers “Username” and “Password”, with each data identifier on a separate line. Figure 152 shows a sample MySQL connection inputs file.

```

* Comments can be added by placing * in first column.
* Format of input file is fixed, with "Port Number = " followed by the MySQL port number, separated by
* any combination of tabs and/or spaces.
* Format of input file is fixed, with "Username = " followed by the MySQL username, separated by any
* combination of tabs and/or spaces.
* If no username is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* Format of input file is fixed, with "Password = " followed by the MySQL password, separated by any
* combination of tabs and/or spaces.
* If no password is required for MySQL, the text "none" (capital or lowercase letters) should be used.
* "Port Number", "Username", "Password" text can be any combination of capital or lowercase letters.
Port Number = 3307
Username = moves
Password = moves

```

**Figure 152. Sample MySQL Connection Inputs File.**

### **zonemonthhour Database Table Output (/DZHM)**

The /DZHM identifier and its corresponding text string specify the name of the database where the zonemonthhour database table is output. This output is optional and, if created, the database will be in the MySQL data folder. Only the name of the database is required and the format of this database table is identical to the MOVES default zonemonthhour database table.

### **zonemonthhour Tab-Delimited Output (/TZHM)**

The /TZHM identifier and its corresponding text string specify the path and filename of the tab-delimited version of the zonemonthhour database table output. This output file is optional and contains the same data as the zonemonthhour database table output, except in tab-delimited form.

### **County Database Table Output (/DCTY)**

The /DCTY identifier and its corresponding text string specify the name of the database where the county database table is output. This output is optional and, if created, the database will be in the MySQL data folder. Only the name of the database is required and the format of this database table is identical to the MOVES default county database table.

### **County Tab-Delimited Output (/TCTY)**

The /TCTY identifier and its corresponding text string specify the path and filename of the tab-delimited version of the county database table output. This output file is optional and contains the same data as the county database table output, except in tab-delimited form.





**APPENDIX A:  
MOVES CODES AND UTILITY ABBREVIATIONS**



**Table 64. MOVES Hours and Abbreviations.**

<b>Hour ID</b>	<b>Hour Name</b>	<b>Hour Abbreviation</b>
1	Hour beginning at 12:00 Midnight	1
2	Hour beginning at 1:00 a.m.	2
3	Hour beginning at 2:00 a.m.	3
4	Hour beginning at 3:00 a.m.	4
5	Hour beginning at 4:00 a.m.	5
6	Hour beginning at 5:00 a.m.	6
7	Hour beginning at 6:00 a.m.	7
8	Hour beginning at 7:00 a.m.	8
9	Hour beginning at 8:00 a.m.	9
10	Hour beginning at 9:00 a.m.	10
11	Hour beginning at 10:00 a.m.	11
12	Hour beginning at 11:00 a.m.	12
13	Hour beginning at 12:00 Noon	13
14	Hour beginning at 1:00 p.m.	14
15	Hour beginning at 2:00 p.m.	15
16	Hour beginning at 3:00 p.m.	16
17	Hour beginning at 4:00 p.m.	17
18	Hour beginning at 5:00 p.m.	18
19	Hour beginning at 6:00 p.m.	19
20	Hour beginning at 7:00 p.m.	20
21	Hour beginning at 8:00 p.m.	21
22	Hour beginning at 9:00 p.m.	22
23	Hour beginning at 10:00 p.m.	23
24	Hour beginning at 11:00 p.m.	24

**Table 65. MOVES Road Types and Abbreviations.**

<b>Road Type ID</b>	<b>Road Type Description</b>	<b>Road Type Abbreviation</b>
1	Off-Network	Off-Network
2	Rural Restricted Access	Rural Restricted Access
3	Rural Unrestricted Access	Rural Unrestricted Access
4	Urban Restricted Access	Urban Restricted Access
5	Urban Unrestricted Access	Urban Unrestricted Access
6	Rural Restricted without Ramps	Rural Restricted without Ramps
7	Urban Restricted without Ramps	Urban Restricted without Ramps
8	Rural Restricted only Ramps	Rural Restricted only Ramps
9	Urban Restricted only Ramps	Urban Restricted only Ramps

**Table 66. MOVES Pollutants and Abbreviations.**

<b>Pollutant ID</b>	<b>Pollutant Name</b>	<b>Pollutant Abbreviation</b>
1	Total Gaseous Hydrocarbons	Total_Gas_HC
2	Carbon Monoxide (CO)	CO
3	Oxides of Nitrogen (NO <sub>x</sub> )	NO <sub>x</sub>
5	Methane (CH <sub>4</sub> )	Methane_(CH4)
6	Nitrous Oxide (N <sub>2</sub> O)	N2O
20	Benzene	Benzene
21	Ethanol	Ethanol
22	MTBE	MTBE
23	Naphthalene	Naphthalene
24	1,3-Butadiene	1,3-Butadiene
25	Formaldehyde	Formaldehyde
26	Acetaldehyde	Acetaldehyde
27	Acrolein	Acrolein
30	Ammonia (NH <sub>3</sub> )	NH3
31	Sulfur Dioxide (SO <sub>2</sub> )	SO2
32	Nitrogen Oxide (NO)	NO
33	Nitrogen Dioxide (NO <sub>2</sub> )	NO2
34	Nitrous Acid (HONO)	HONO
35	Nitrate (NO <sub>3</sub> )	NO3
36	Ammonium (NH <sub>4</sub> )	NH4
40	2,2,4-Trimethylpentane	2,2,4-Trimethylpentane
41	Ethyl Benzene	Ethyl_Benzene
42	Hexane	Hexane
43	Propionaldehyde	Propionaldehyde
44	Styrene	Styrene
45	Toluene	Toluene
46	Xylene	Xylene
51	Chloride	Chloride
52	Sodium	Sodium
53	Potassium	Potassium
54	Magnesium	Magnesium
55	Calcium	Calcium
56	Titanium	Titanium
57	Silicon	Silicon
58	Aluminum	Aluminum
59	Iron	Iron
60	Mercury Elemental Gaseous	Mercury_Elemental_Gas
61	Mercury Divalent Gaseous	Mercury_Divalent_Gas
62	Mercury Particulate	Mercury_Part particulate
63	Arsenic Compounds	Arsenic_Compounds

**Table 66. MOVES Pollutants and Abbreviations (Continued).**

<b>Pollutant ID</b>	<b>Pollutant Name</b>	<b>Pollutant Abbreviation</b>
65	Chromium 6+	Chromium_6+
66	Manganese Compounds	Manganese_Compounds
67	Nickel Compounds	Nickel_Compounds
68	Dibenzo(a,h)anthracene particle	Dibenzo(a,h)anthracene_particle
69	Fluoranthene particle	Fluoranthene_particle
70	Acenaphthene particle	Acenaphthene_particle
71	Acenaphthylene particle	Acenaphthylene_particle
72	Anthracene particle	Anthracene_particle
73	Benz(a)anthracene particle	Benz(a)anthracene_particle
74	Benzo(a)pyrene particle	Benzo(a)pyrene_particle
75	Benzo(b)fluoranthene particle	Benzo(b)fluoranthene_particle
76	Benzo(g,h,i)perylene particle	Benzo(g,h,i)perylene_particle
77	Benzo(k)fluoranthene particle	Benzo(k)fluoranthene_particle
78	Chrysene particle	Chrysene_particle
79	Non-Methane Hydrocarbons	NMHC
80	Non-Methane Organic Gases	NMOG
81	Fluorene particle	Fluorene_particle
82	Indeno(1,2,3,c,d)pyrene particle	Indeno(1,2,3,c,d)pyrene_particle
83	Phenanthrene particle	Phenanthrene_particle
84	Pyrene particle	Pyrene_particle
86	Total Organic Gases	TOG
87	Volatile Organic Compounds	VOC
88	NonHAPTOG	NonHAPTOG
90	Atmospheric CO <sub>2</sub>	Atmospheric_CO2
98	CO <sub>2</sub> Equivalent	CO2_Equivalent
100	Primary Exhaust PM <sub>10</sub> - Total	PM10_Total_Exh
106	Primary PM <sub>10</sub> - Brakewear Particulate	PM10_Brakewear
107	Primary PM <sub>10</sub> - Tirewear Particulate	PM10_Tirewear
110	Primary Exhaust PM <sub>2.5</sub> - Total	PM2.5_Total_Exh
111	Organic Carbon	Organic_Carbon
112	Elemental Carbon	Elemental_Carbon
115	Sulfate Particulate	Sulfate_Part particulate
116	Primary PM <sub>2.5</sub> - Brakewear Particulate	PM2.5_Brakewear
117	Primary PM <sub>2.5</sub> - Tirewear Particulate	PM2.5_Tirewear
118	Composite - NonECPM	Composite_NonECPM
119	H <sub>2</sub> O (aerosol)	H2O_aerosol
120	Primary PM <sub>2.5</sub> - NonECNonSO4PM	PM2.5_NonECNonSO4PM
121	CMAQ5.0 Unspeciated (PMOTHR)	PMOTHR
122	Non-carbon Organic Matter (NCOM)	NCOM
130	1,2,3,7,8,9-Hexachlorodibenzo-p-Dioxin	1,2,3,7,8,9-HxCDD

**Table 66. MOVES Pollutants and Abbreviations (Continued).**

<b>Pollutant ID</b>	<b>Pollutant Name</b>	<b>Pollutant Abbreviation</b>
131	Octachlorodibenzo-p-dioxin	OCDD
132	1,2,3,4,6,7,8-Heptachlorodibenzo-p-Dioxin	1,2,3,4,6,7,8-HpCDD
133	Octachlorodibenzofuran	OCDF
134	1,2,3,4,7,8-Hexachlorodibenzo-p-Dioxin	1,2,3,4,7,8-HxCDD
135	1,2,3,7,8-Pentachlorodibenzo-p-Dioxin	1,2,3,7,8-PeCDD
136	2,3,7,8-Tetrachlorodibenzofuran	2,3,7,8-TCDF
137	1,2,3,4,7,8,9-Heptachlorodibenzofuran	1,2,3,4,7,8,9-HpCDF
138	2,3,4,7,8-Pentachlorodibenzofuran	2,3,4,7,8-PeCDF
139	1,2,3,7,8-Pentachlorodibenzofuran	1,2,3,7,8-PeCDF
140	1,2,3,6,7,8-Hexachlorodibenzofuran	1,2,3,6,7,8-HxCDF
141	1,2,3,6,7,8-Hexachlorodibenzo-p-Dioxin	1,2,3,6,7,8-HxCDD
142	2,3,7,8-Tetrachlorodibenzo-p-Dioxin	2,3,7,8-TCDD
143	2,3,4,6,7,8-Hexachlorodibenzofuran	2,3,4,6,7,8-HxCDF
144	1,2,3,4,6,7,8-Heptachlorodibenzofuran	1,2,3,4,6,7,8-HpCDF
145	1,2,3,4,7,8-Hexachlorodibenzofuran	1,2,3,4,7,8-HxCDF
146	1,2,3,7,8,9-Hexachlorodibenzofuran	1,2,3,7,8,9-HxCDF
168	Dibenzo(a,h)anthracene gas	Dibenzo(a,h)anthracene_gas
169	Fluoranthene gas	Fluoranthene_gas
170	Acenaphthene gas	Acenaphthene_gas
171	Acenaphthylene gas	Acenaphthylene_gas
172	Anthracene gas	Anthracene_gas
173	Benz(a)anthracene gas	Benz(a)anthracene_gas
174	Benzo(a)pyrene gas	Benzo(a)pyrene_gas
175	Benzo(b)fluoranthene gas	Benzo(b)fluoranthene_gas
176	Benzo(g,h,i)perylene gas	Benzo(g,h,i)perylene_gas
177	Benzo(k)fluoranthene gas	Benzo(k)fluoranthene_gas
178	Chrysene gas	Chrysene_gas
181	Fluorene gas	Fluorene_gas
182	Indeno(1,2,3,c,d)pyrene gas	Indeno(1,2,3,c,d)pyrene_gas
183	Phenanthrene gas	Phenanthrene_gas
184	Pyrene gas	Pyrene_gas
185	Naphthalene gas	Naphthalene_gas
1000	CB05 Mechanism	CB05_Mechanism
1001	CB05_ALD2	CB05_ALD2
1002	CB05_ALDX	CB05_ALDX
1003	CB05_BENZENE	CB05_BENZENE
1004	CB05_CH4	CB05_CH4

**Table 66. MOVES Pollutants and Abbreviations (Continued).**

<b>Pollutant ID</b>	<b>Pollutant Name</b>	<b>Pollutant Abbreviation</b>
1005	CB05_ETH	CB05_ETH
1006	CB05_ETHA	CB05_ETHA
1007	CB05_ETOH	CB05_ETOH
1008	CB05_FORM	CB05_FORM
1009	CB05_IOLE	CB05_IOLE
1010	CB05_ISOP	CB05_ISOP
1011	CB05_MEOH	CB05_MEOH
1012	CB05_OLE	CB05_OLE
1013	CB05_PAR	CB05_PAR
1014	CB05_TERP	CB05_TERP
1015	CB05_TOL	CB05_TOL
1017	CB05_UNR	CB05_UNR
1018	CB05_XYL	CB05_XYL
1500	CB6_Mechanism	CB6_Mechanism
1501	CB6_ALD2	CB6_ALD2
1502	CB6_ALDX	CB6_ALDX
1503	CB6_CH4	CB6_CH4
1504	CB6_ETH	CB6_ETH
1505	CB6_ETHA	CB6_ETHA
1506	CB6_ETOH	CB6_ETOH
1507	CB6_FORM	CB6_FORM
1508	CB6_IOLE	CB6_IOLE
1509	CB6_ISOP	CB6_ISOP
1510	CB6_MEOH	CB6_MEOH
1511	CB6_OLE	CB6_OLE
1512	CB6_PAR	CB6_PAR
1513	CB6_TERP	CB6_TERP
1514	CB6_TOL	CB6_TOL
1515	CB6_UNR	CB6_UNR
1516	CB6_XYL	CB6_XYL
1517	CB6_ACET	CB6_ACET
1518	CB6_BENZ	CB6_BENZ
1519	CB6_ETHY	CB6_ETHY
1520	CB6_KET	CB6_KET
1521	CB6_PRPA	CB6_PRPA
1522	CB6_Mechanism	CB6_Mechanism
1523	CB6_ALD2	CB6_ALD2



**Table 67. MOVES Emissions Processes and Abbreviations.**

Process ID	Process Name	Process Abbreviation
1	Running Exhaust	Running_Exh
2	Start Exhaust	Start_Exh
9	Brake Wear	Brakewear
10	Tire Wear	Tirewear
11	Evaporative Permeation	Evap_Permeation
12	Evaporative Fuel Vapor Venting	Evap_Fuel_Vent
13	Evaporative Fuel Leaks	Evap_Fuel_Leak
15	Crankcase Running Exhaust	Crank_Run_Exh
16	Crankcase Start Exhaust	Crank_Start_Exh
17	Crankcase Extended Idle Exhaust	Crank_Ext_Idle
90	Extended Idle Exhaust	Ext_Idle_Exh
91	Auxiliary Power Exhaust	Aux_Power_Exh

**Table 68. MOVES Average Speed Bins and Abbreviations.**

Average Speed Bin ID	Average Speed Bin Description	Average Speed Bin Abbreviation <sup>1</sup>
1	Speed < 2.5 mph	2.5
2	2.5 mph <= Speed < 7.5 mph	5
3	7.5 mph <= Speed < 12.5 mph	10
4	12.5 mph <= Speed < 17.5 mph	15
5	17.5 mph <= Speed < 22.5 mph	20
6	22.5 mph <= Speed < 27.5 mph	25
7	27.5 mph <= Speed < 32.5 mph	30
8	32.5 mph <= Speed < 37.5 mph	35
9	37.5 mph <= Speed < 42.5 mph	40
10	42.5 mph <= Speed < 47.5 mph	45
11	47.5 mph <= Speed < 52.5 mph	50
12	52.5 mph <= Speed < 57.5 mph	55
13	57.5 mph <= Speed < 62.5 mph	60
14	62.5 mph <= Speed < 67.5 mph	65
15	67.5 mph <= Speed < 72.5 mph	70
16	Speed >= 72.5 mph	75

<sup>1</sup> The average speed bin abbreviation is the same as the MOVES average bin speed.

**Table 69. MOVES SUTs and Abbreviations.**

<b>SUT ID</b>	<b>SUT Description</b>	<b>SUT Abbreviation</b>
11	Motorcycle	MC
21	Passenger Car	PC
31	Passenger Truck	PT
32	Light Commercial Truck	LCT
41	Intercity Bus	IBus
42	Transit Bus	TBus
43	School Bus	SBus
51	Refuse Truck	RT
52	Single Unit Short-Haul Truck	SUShT
53	Single Unit Long-Haul Truck	SULhT
54	Motor Home	MH
61	Combination Short-Haul Truck	CShT
62	Combination Long-Haul Truck	CLhT

**Table 70. MOVES Fuel Types and Abbreviations.**

<b>Fuel Type ID</b>	<b>Fuel Type Description</b>	<b>Fuel Type Abbreviation</b>
1	Gasoline	Gas
2	Diesel Fuel	Diesel
3	Compressed Natural Gas (CNG)	CNG
4	Liquid Petroleum Gas (LPG)	LPG
5	Ethanol (E-85)	Eth_E85
9	Electricity	Elec

**Table 71. MOVES Activity Types and Abbreviations.**

<b>Activity Type ID</b>	<b>Activity Type Description</b>	<b>Activity Type Abbreviation</b>
1	Distance	1
3	Extended Idle Hours	3
5	Source Hours Parked	5
6	Population	6
7	Starts	7
13	Hotelling Diesel Aux (APU Hours)	13