Appendix C
Framework Operational Considerations

Table of Contents

1. Framework Operation/Execution

The framework executes in a non-real-time fashion using batch processes which operate in a predefined sequence to simulate traffic within the three analysis hours. When a traffic model executes it will simulate traffic conditions simultaneously for the three analysis hours. This processing, across the analysis hours, repeats for each iteration of the framework to achieve MOE convergence (a framework iteration has nothing to do with a time interval). Modifications are made to link speed, during each framework iteration, and these changes are used by the planning assignment model, between iterations, to reassign vehicle trips onto the network to achieve minimum travel times. Subsequent iterations of the framework will operate on refined speed and volume predictions, from the previous iteration, by using the revised trip data. The process is repeated over eight iterations which are usually sufficient to see convergence in link speed and volume estimates for each analysis hour. (A weighted average technique is used to determine new speeds factoring in previous iteration results.) When the last framework iteration occurs a stable condition will have been obtained for the study area network. The framework generates individual link predictions for several MOEs (speed, volume, VHD, emissions, etc.) given a specific set of ATMS user services that are selected by the framework operator.

A batch file called TL.BAT is invoked initially to setup the environment for the BTRIEVE database which is fundamental to operation of the SYSTEM II planning model. The TL.BAT batch file then invokes BTRIEVE to get the database running through a short batch file called STARTBT.BAT. This operation insures that BTRIEVE is running so that subsequent calls to SYSTEM II programs can be successfully executed.

Depending upon the type of ATMS services selected by a scenario, one of five specific batch files will be called; COMl_RUN.BAT, COM2_RUN.BAT, PL_RUNBAT, PE_F_RUN.BAT or PE_D_RUN.BAT. These files are used to control framework execution in a variety of different ways so that specific ATMS services can be enabled and modeled by the framework. Each of these files, at this level, are structured similarly in the way they invoke reassignment and execute the read speed functions for each iteration. They differ however, in the type of calls made to the traffic simulation models and the setup of data files passed to these models.

After execution of the assignment module the DUMPLINK.BAT and DUMPTURN.BAT files are called which invoke SYSTEM II programs to preprocess link data for freeway and arterial simulations respectively.

The ClINTRFC.BAT, C2INTRFC.BAT, PLINTRFC.BAT and PEINTRFC.BAT files directly control execution of the FREQ model. They make calls to PRESIM, FREQPL (lane simulation), FREQPE (simulation of entry ramps) and POSTSIM for data processing. The freeway analysis is done independently for south bound and north bound traffic. Within a single call to FREQPE or FREQPL all 12 time intervals (15 minutes each) are evaluated which represent the three analysis hours. The ClINTRFCBAT, C2INTRFC.BAT, PLINTRFC.BAT and PEINTRFCBAT batch files will call the TRANSYT-7F arterial simulation when all freeway simulation has completed, for a particular iteration, by invoking RUNSYS.BAT.

Execution of TRANSYT-7F is controlled by the RUNSYS.BAT file. This batch file calls several interface programs that condition the network data for arterial analysis. The invocation of TRANSYT-7F is accomplished by a call to the RUN7F.BAT file. Arterial TRANSYT-7F simulation is based upon discrete 15 minute time slices and multiple calls to TRANSYT-7F are made by RUN7F.BAT to span the entire analysis period of three hours. When arterial analysis is completed for all signalized sub-networks, for the three analysis hours, a speed compare executable SPEEDCMP is executed to compute weighted link speed, volume and delays for the next iteration of the framework.

At the conclusion of each iteration the file READSPD?.BAT (?=l-8) converts each of the SYSTEM II analysis hour data files into an ASCII representation. The reassignment batch file REASSGN?.BAT is used to update link volumes associated with assignments before the next iteration of the framework. These batch files make calls to SYSTEM II executables.

After the framework completes eight traffic simulation iterations the determination of ATMS impacts is performed by calling the batch file IMPACTSBAT. This batch file sets up the SYSTEM II environment and invokes BTRIEVE. The next batch file to be invoked is called EMISRUN.BAT and this file executes the program EMIS which calculates running emissions using EMFAC and MOBILE emission rates.

The last batch file invoked, to determine impacts, is called SAFERUN.BAT which executes the program SAFETY to calculate safety related MOEs for property damage only (PDO), injury and total accidents.

A super batch file called FRAME_WK.BAT can be used to launch execution of the entire framework. One simple command line is all that is required to start and control the framework execution and no further intervention is necessary by the user. With invocation of the FRAME_WK.BAT file a user selects the desired ATMS services to be modeled. The selection is performed by passing input parameters to the batch file in the form of Pl, Fl and Tl. (2 and 3 are also valid, refer to table 12-1 in section 12 of framework report).

The framework can be executed by typing the following DOS commands ending each line with a return key:

```
CD D:\
```

FRAME_WK A B C

where: A=Pl, P2 or P3

B=Fl, F2, F3, FlF2 or FlF3

C=Tl,T2orT3

any other sequence of input parameters are invalid.

For example, the framework can be invoked by typing;

FRAME_WK P2 FIF3 *T-2*

which selects ATMS scenario #4 to simulate synchronized freeway ramp metering, optimizing free flow, combined with fixed time signal coordination on the parallel arterial.

The complete FRAME_WK.BAT file is listed in section 1.2. All other batch files used by FRAME_WK.BAT are included in section 8. Note that TL.BAT must reside in the directory D:\TL. It is assumed that the AUTOEXEC.BAT file will start BTRIEVE initially to support execution of any other planning model utility that may be requested by the user for initialization or setup.

1.1 Framework Batch File Execution Steps

Super batch file FRAME_WK.BAT is used to control execution of the framework for the novice user. The FRAME_WK.BAT file performs a number of operations which are described in the following execution steps:

1) CD TL\PLAN\P1 (or P2, P3)

Select the planning model base scenario directory that contains setup files. These directories contain SYSTEM II planning model output data for trip assignments, link volumes and turning movements.

```
2) COPY Pl_LINK.DAT ..\I880LINK.DAT      (or P2, P3)
   COPY P1_TRN?.DAT ..\TURN?.DAT         (or P2, P3)
   COPY Pl_AM?.DAT ..\HR?_TRIP.DAT       (or P2, P3)
```

Copy the information from the base scenario directory into the working planning model directory for execution. This operation loads the network link structure, turning information and trip assignments into the planning model directory. These files are different for base scenarios Pl, P2 and P3 since they represent a different combinations of HOV, ramp meter by-pass and lack of HOV lanes.

Pl - HOV lane

P2 - HOV lane + HOV ramp metering by-pass

P3 - no HOV lane + HOV ramp metering by-pass

3) CD TL\ASIM\DEMO

Change the default directory to select the DEMO subdirectory under the arterial simulation directory ASIM. This will setup the appropriate destination area for the following copy function.

4) COPY DEMO-Tl." ..\DEMO.* (or T2, T3)

This operation transfers the link-node association files that are used to provide translation information between the planning and traffic simulation models for link-node data. Each model will use a different definition of nodes and links and the DEMO.LNA files are used to provide the necessary mapping or equivalence information to identify the same node and link reference used between the models. Theses files are copied into the traffic simulation directory ASTM.

5) CD TL\ASIM\TIN

Modify the default directory to point to the traffic signal sub-networks.

6) COPY NET?-Tl.TIN ..\NET?.TIN (or T2, T3)

This operation will transfer the eight traffic signal coordination sub-network input files into the working directory for the arterial traffic simulation model.

7) INVOKE C:\TL.BAT

Calls the batch file used to set the paths and variables for the SYSTEM II planning model. It selects input files for calibration programs, screen control and planning model environment definition. This batch file also calls the BTRIEVE utility that is used to begin executing the database environment which is required to run the SYSTEM II planning model. (SYSTEM II resides on top of the BTRIEVE database)

8) CD TL\PLAN

Change default directory to the planning model.

9) INVOKE COMl_RUN (or COM2_RUN, PL_RUN, PE_D_RUN, PE_F_RUN)

These are the main scenario batch files that control execution of the particular freeway FREQ programs within the Framework. There are four configurations of FREQ that can be invoked.

PL_RUN **(F1)** - HOV lane analysis enabled

PE_F_RUN (F2) - Fixed Ramp Metering

PE_D_RUN (F3) - Dynamic Ramp Metering

COMl_RUN (FlF2) - HOV lane analysis enabled + Fixed Ramp Metering

COM2_RUN (FlF3) - HOV lane analysis enabled + Dynamic Ramp Metering

The batch files referenced above also launch execution of the TRANSYT-7F arterial simulation programs. These batch files control model execution to enable a total of 8 iterations for each analysis hour. They are also used to call SYSTEM II (planning model) for reassignment of the trip distribution after each iteration.

10) COPY I880LlNK.DAT D:\TL\PLAN\PlFlT3\I880LlNK.DAT (or P?F?F?T?)

After execution of the planning and traffic models are complete, the link data contained in the PLAN directory is saved into one of the appropriate scenario results directories such as PlFlT3. The destination directory selected is dependent upon the particular scenario being executed. This operation will save link data results for later analysis. The file I880LlNK.DAT contains all operational MOEs calculated during framework execution. This file serves as the source location to extract operational MOEs for statistical analysis.

11) COPY I880LINK.DAT D:\TL\IMP\I880LINK.DAT

This operation transfers a copy of the link results data into the impacts (IMP) directory so that the MOEs associated with safety, fuel consumption and emissions can be determined.

12) EXECUTE IMPACTS.BAT

This batch file sets the path and variables for determining the impacts. It then calls BTRIEVE.

13) EXECUTE EMISRUN.BAT

Calculates the emissions by executing EMIS which uses EMFAC and MOBIL emission rates. This program also determines total fuel consumption. The batch file also executes the PERF routine to calculate an emissions report and creates a print file. (*.PRN)

14) EXECUTE SAFERUN.BAT

This batch file calculates the safety MOEs and generates a PERF report to log the results (*.PRN).

15) COPY I880LINK.EMS, *.PRN, *.ASC, *.OUT

D:\TL\PLAN\PlFlT3\ I880LINK.EMS, *.PRN, *.ASC, *.OUT (or P?F?T?)

At the conclusion of the framework execution saves a copy of the impacts data generated by the impact routines into the appropriate scenario directory for future analysis. The I880LINK.EMS file contains MOEs for emissions, fuel consumption and safety factors. This file is used as an information source to extract emissions, fuel and safety MOEs for statistical analysis. Generally, the PERF print files (*.PRN) are not directly used for MOE analysis but can be referenced to debug problems detected during the setup of new Framework transportation services.

1.2 Batch File FRAME_WK.BAT

A complete copy of the FRAME_WK.BAT batch file is listed below.

```
echo off
REM    Creation date March 3 1,1995 Revision D
REM
REM    Batch file for controlling the Volpe ITS Framework execution.
REM    Disk drive should be set for D: before calling Frame_wk.bat A B C
REM    Where: A, B and C are characters;
REM    A represents P 1, P2 or P3
REM    B represents Fl, F2, F3, FlF2, or FlF3
REM    C represents T 1, T2, or T3
REM    Any other combinations of P, F and T are invalid.
cls
echo **** VOLPE ITS FRAMEWORK ****
echo.
echo The ITS Framework services were selected as follows:
echo --Planning Model = %l
echo --Freeway Model = %2
echo --Arterial Model = %3
echo.
Pause
REM    Validate the inputs here to be sure they are valid.
if "%1 "==" Pl" goto VALIDA
if "%1 "==" P2" goto VALIDA
if "%1 "==" P3" goto VALIDA
if "%1 "==" pl" goto VALIDA
if "%1 "==" p2" goto VALIDA
if "%1 "==" p3" goto VALIDA
echo Invalid input parameter for Planning Model selection
goto END
:VALIDA
if “%2”==“Fl” goto VALIDB
if “%2”==“F2” goto VALIDB
if "%2"=="F3” goto VALIDB
if “%2”==“FlF2” goto VALIDB
if “%2”==“FlF3” goto VALIDB
if "%2"=="f1” goto VALIDB
if "%2"=="f2” goto VALIDB
if "%2"=="f3" goto VALIDB
if "%2"=="f1f2” goto VALIDB
if “%2”==“ flf3” goto VALIDB
if "%2"=="flF2” goto VALIDB
if “%2 "=="flF3”goto VALIDB
if "%2"=="Flf2” goto VALIDB
if “%2”==“F1f3” goto VALIDB
echo Invalid input parameter for Freeway Model selection
goto END
:VALlDB
if "%3"=="Tl”goto VALIDC
if "%3"=="T2”goto VALIDC
if "%3"=="T3”goto VALIDC
if "%3"=="t 1” goto VALIDC
if "%3"=="t2" goto VALIDC
if "%3"=="t3" goto VALIDC
echo Invalid input parameter for Arterial Transit Model selection
goto END
:VALIDC
echo Beginning Framework Execution...
REM    Create a results directory given the ATMS services specified.
md \tl\plan\% 1%2%3
REM    Select the appropriate setup directory with Planning model data.
REM    Transfer selected data files into Planning directory.
cd \tl\plan\% 1
copy %l_link.dat ..\i880link.dat
copy %l_turnl.dat ..\turnl .dat
copy % l_turn2.dat ..\turn2.dat
copy %1_turn3.dat ..\turn3.dat
copy %1_am1.dat ..\hrl_trip.dat
copy % l_am2.dat ..\hr2_trip.dat
copy % l_am3.dat ..\hr3_trip.dat
REM    Transfer the association and equivalence files between models.
cd \tl\asim
de1 demo.*
cd \tl\asim\demo
copy demo-%3.* ..\demo.*
REM    Transfer the signal coordination sub-net files into the Arterial Simulation
REM   directory.
```

```
cd \tl\asim
de1 net*.tin
del net*.ios
de1 net*.tof
cd \tl\asim\tin
copy netl-%3.tin ..\netl.tin
copy net2-%3.tin ..\net2.tin
copy net3-%3.tin ..\net3.tin
copy net4-%3.tin ..\net4.tin
copy net5%3.tin ..\netUin
copy net6-%3.tin ..\net6.tin
copy net7-%3.tin ..\net7.tin
copy net8-%3.tin ..\net8.tin
REM     Call Btrieve to set up Planning model environment.
REM     Assumes that tl.bat file in the D:\tl directory.
cd \tl
call tl.bat
REM     Start execution of the appropriate ATMS users services as specified
REM   previously.
cd \tl\plan
if “%2”==“Fl” call pl_run.bat
if "%2"=="f 1” call pl_run.bat
if “%2”==“F2” call pe_f_runbat
if "%2"=="f2” call pe_f_run.bat
if “%2”==“F3” call pe_d_run.bat
if "%2"=="f3” call pe_d_runbat
if “%2”==-“FlF2” call coml_run.bat
if "%2"=="flf2” call coml_run.bat
if "%2"=="Flf2" call coml_run.bat
if "%2"=="flF2" call coml_run.bat
if “%2”==“FlF3” call com2_run.bat
if "%2"=="flf3” call com2_run.bat
if “%2”==“Flf3” call com2_run.bat
if “%2“==“f lF3” call com2_run.bat
REM     Save resulting link data into the appropriate directory for future analysis.
REM     Also save the print, average and .asc files for analysis latter.
cd \tl\plan
copy *.dat \tl\plan\% 1%2%3\*.dat
copy *.prn \tl\plan\%l%2%3\*.prn
copy *.avg \tl\plan\% 1%2%3\*.avg
copy *.asc \tl\plan\%1%2%3\*.asc
REM     Setup the Impacts directory to calculate safety, fuel and emission MOEs
REM     and invoke the appropriate processing.
copy i880link.dat \tl\imp\i880link.dat
cd \tl\imp
```

```
call impacts.bat
call emisrun.bat
call saferun.bat
REM     Save the impacts, reports, output and interface files for future analysis
cd \tl\imp
copy i880link.ems \tl\plan\% 1%2%3\i880link.ems
copy *.prn \tl\plan\% 1%2%3\*.pm
copy *.asc \tl\plan\% 1%2%3\*.asc
copy *.out \tl\plan\%1%2%3\*.out
REM     Extract result data from planning directory and format for STATISTICA'.
REM     New STATISTICA formatted file is called statist.asc
cd \tl
call tl.bat
cd \tl\plan\% 1%2%3
call \tl\plan\statexp.bat
cd \tl
:END
echo.
echo ++Framework execution completed!
```

[1] Note: A statistical analysis package, STATISTICA, was employed in the prototype. Any similar package may be substituted.

2. Operational Requirements

The amount of time required to run the modeling framework is dependent on the type of computer hardware being used. There are numerous options available to reduce the actual run time required by the framework. The most effective option is to decrease the number of cycle lengths that the TRANSIT-7F arterial simulation optimization procedure must test. Other approaches include using the TRANSYT-7F "quick optimization" step sizes, and eliminating unnecessary output reports from TRANSYT-7F. Optional outputs include time-space diagrams, arterial corridor summary tables, timing tables, initial performance tables and flow profile plots. Arterial street modeling in the framework only requires an input summary table and the final performance table from TRANSYT-7F runs.

2.1 Hardware Requirements

The framework was designed to execute on DOS based IBM compatible personal computers with a minimum of 640 Kbytes of memory. The planning, traffic simulation, emission and safety models are designed for non-Windows based operating environments. The statistical analysis package, selected by the Volpe Center, requires Microsoft Windows V3.1 or better with a minimum of 4 Mbytes of memory. Practical considerations suggest an operational environment be composed of at least a 486 class machine with an integrated math coprocessor, 4 Mbytes of memory and a hard drive of at least 300 Mbytes.

The hardware configuration used by the Volpe Center staff to conduct testing of the various ATMS scenarios consisted of a 90 Mhz Pentium processor, 8 Mbytes of memory and a 500 Mbyte hard drive. An additional 250 Mbyte removable drive was installed in the workstation which served to provide the dedicated framework files and directory structures required. This removable drive also enables convenient sharing of framework data files among project team members since these data files can typically be 10 Mbytes or greater in size.

2.1.1 DOS Memory Requirements

The TRANSYT-7F program (up to Release 6 - Version 4) requires at least 587 thousand bytes of free DOS memory (below "640 K" or "low memory") to run. Therefore, any memory resident programs that could reduce the availability of memory below this threshold should not be installed when running the arterial street simulation. All other modules will function properly with substantially less free DOS memory.

2.1.2 Disk Storage Requirements

The amount of hard disk space occupied by programs and data files is highly dependent upon the corridor network, the number of TRANSYT-7F networks, and the number of time intervals being simulated. All of the framework executable program modules require approximately 1,650 thousand bytes of hard disk storage. The following formula may be used to estimate the amount of hard disk storage space required by the program modules, input and output files, and configuration files. The formula does not account for the turning movement volume database and the link attribute database provided by the planning model.

$$\text{Hard Disk Space (kilobytes)} = 1681 + (T + 1)(16N + 31) + 1.75(I)(P)$$

where I = Total number of intersections in all networks

T = Number of simulation time intervals

N = Number of TRANSYT-7F networks

P = Number of planning model time periods

For example, the largest network used in the demonstration project data set included 52 intersections in eight networks with three planning model time intervals and twelve simulation time intervals. The computed amount of hard disk space is 5,646 kilobytes. Therefore, approximately six megabytes of free hard disk space should be provided, excluding the turning movement volume and link attribute databases. Note that a majority of this space is required to hold data files created during the execution of the framework.

2.2 Software Requirements

The planning model software package used for this application of the framework is SYSTEM II version 2. A fully validated SYSTEM II model of Alameda County, the location of the study corridor, was available at the time the project began and was incorporated into the framework prototype. SYSTEM II is a transportation modeling and information system developed by JHK & Associates for microcomputer applications. SYSTEM II performs the traditional elements of trip generation, trip distribution, mode choice and assignment in addition to other innovative features designed to meet the needs of detailed subarea analysis. Regional information is stored in relational databases with five levels of geographic aggregation and a series of analysis years. SYSTEM II also provides graphic editors that permit the user to edit and view data in a graphical presentation format. During the course of the study, software was developed to link to other planning models, EMME/2, TRANPLAN and MINUTP. The interested reader is referred to chapter 5 in the Program Reference Guide (1994).

TRANSYT-7F version 4 is a macroscopic, deterministic traffic simulation program that can simulate or optimize traffic signal timing on a coordinated traffic signal network. The program can optimize cycle lengths, phase lengths and offsets to minimize delay, stops, fuel consumption and user costs, but does not optimize the pattern of signal phasing used at each intersection. This pattern must be selected by the user in advance. TRANSYT-7F is maintained by the McTrans Center for Microcomputers in Transportation at the University of Florida at Gainesville. The other framework modules developed for arterial street simulation modeling have been found to work well with TRANSYT-7F Version 4, Release 4 through Release 6. Release 7 of TRANSYT-7F includes some changes in the column format of the output tables and will not work with the framework at this time.

The freeway simulation component of the model framework relies upon FREQ, a macroscopic simulation model that simulates traffic operations in a freeway corridor. The function of FREQ in the model framework is to provide refined estimates of the operational characteristics of the freeway facilities in the study area. In the context of the model framework, FREQ version 11 (latest version of FREQ) is used to simulate only freeway mainline segments and ramps.

The first version of the model (FREQl) was developed in 1968 by the Institute of Transportation Studies at UC-Berkeley supported by a Caltrans research grant. Since then there have been eleven improved versions of the model. The Institute of Transportation Studies has the software copyright and provides on-call software support. There have been hundreds of FREQ applications installed throughout the United States. FREQ was originally developed to run on mainframe computers but in 1989 it was converted to run on IBM-compatible microcomputers. To accommodate microcomputer limitations, the model was separated into two sub-programs that would analyze priority lane (PL) and priority entry (PE) scenarios.

Both MOBILE 5a and EMFAC7F estimate hot and cold start emissions and hot stabilized exhaust emissions for carbon monoxide (CO), hydrocarbons (HC), and oxides of nitrogen (NOx). MOBILE 5a, is a model developed by the Environmental Protection Agency (EPA) and is used in every state except California, and EMFAC7F, is the current version of the model developed by the California Air Resources Board. Emission rates from the two rate models were applied to estimates of link volume and speed produced by the model framework. Link-specific estimates of emissions were derived using three programs: TOD (Time-of-Day), EMISSION (programs within SYSTEM II), and EMIS, a program developed specifically for this project.

3. Directory Structures

To execute the framework it is necessary to establish a specific set of directories on a single disk drive that can be used for storing programs and data files. Each step of the iterative framework process requires particular files to produce results for subsequent steps. These files generally remain within a given sub-directory, with the exception of those files used to transfer information between sub-directories. For example, while the planning directory (D:\TL\PLAN\*.*) contains turning movement files, trip table files, and a linkfile, only ASCII files are used to transfer data to both the freeway simulation (D:\TL\FSIM\*.*) and arterial simulation (D:\TL\ASIM\*.*) directories.

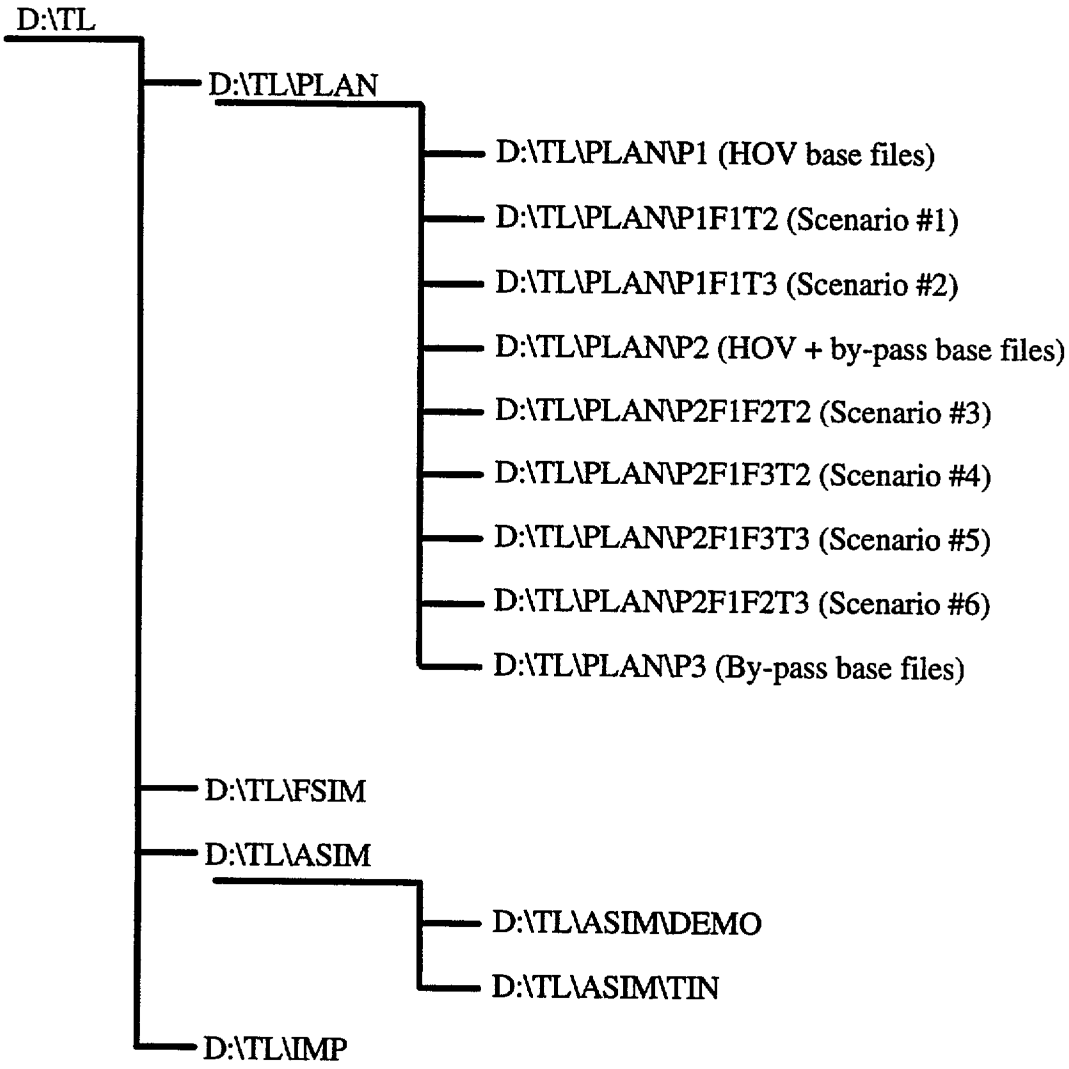
D:\TL
D:\TL\PLAN
D:\TL\PLAN\P1 (HOV base files)
D:\TL\PLAN\P1F1T2 (Scenario #1)
D:\TL\PLAN\P1F1T3 (Scenario #2)
D:\TL\PLAN\P2 (HOV + by-pass base files)
D:\TL\PLAN\P2F1F2T2 (Scenario #3)
D:\TL\PLAN\P2F1F3T2 (Scenario #4)
D:\TL\PLAN\P2F1F3T3 (Scenario #5)
D:\TL\PLAN\P2F1F2T3 (Scenario #6)
D:\TL\PLAN\P3 (By-pass base files)
D:\TL\FSIM
D:\TL\ASIM
D:\TL\ASIM\DEMO
D:\TL\ASIM\TIN
D:\TL\IMP

Figure C-1: Framework Directory Structure

The following sections describe the FILE naming conventions used and also identifies the primary files that are contained within each sub-directory.

3.1 Planning Model Directory (D:\TL\PLAN\*.*)

The batch files (*.BAT) contained in this directory are used to run a series of commands that perform specific tasks. Definitions of batch file names used by the framework follow:

a) TL.BAT - This file is invoked to place the user in the proper start directory. It sets the correct path and opens the appropriate SYSTEM II Calibration, Control, and Setup files. The creation of base link files require similar batch files (TLPl.BAT, TLP2.BAT, etc.) to call the proper *.SET files. The third and fourth characters in the batch filename (Pl, P2, etc.) match the directory name of the particular base link file.

b) COM2_RUN.BAT - This file is used to run a scenario. Each scenario run is described by the portion of the batch filename preceding the underscore. (Thus, COM2 indicates a run involving a combination of FREQPL and FREQPE, version #2.) Other examples might include PL_RUN.BAT, PE_F_RUN.BAT, PE_D_RUN.BAT, and COMl_RUN.BAT.

c) C2INTRFC.BAT - This file is called by COM2_RUN.BAT. The ??INTRFC.BAT file is defined by the first two characters of the filename. Other examples include PLINTRFC.BAT, PEINTRFC.BAT, and ClINTRFC.BAT.

d) REASSGN8.BAT - This file runs a series of DATAEDIT programs and the planning model assignment. The last digit in the filename (from 1-8) indicates the iteration being run.

e) READSPD8BAT - This file reads the LINK?.ASC file and can be used to produce performance reports for each iteration. The iteration is indicated by the last digit in the filename (from O-8).

f) Other batch files without fixed naming conventions include DUMPLINK.BAT and DUMPTURN.BAT which are contained in this appendix.

There are several planning directory data (*.DAT) files that are used by various SYSTEM II programs. Examples of some data files include:

a) I880LINK.DAT - A generic name for the SYSTEM II link file used in the iterative process.

b) TURN3.DAT - A generic name for one of three turning movement files created by SYSTEM II's ASSIGN program. The last digit in the filename indicates the corresponding hour (i.e. 1 = pre-peak hour, 2 = peak hour, 3 = post-peak hour).

c) HR3_TRIP.DAT - One of three hourly trip tables used by the ASSIGN program. The digit following HR in the filename (3, in this case) indicates the applicable study hour.

d) I880NODE.DAT - The resident SYSTEM II node file. This file does not change for any of the scenarios.

e) I880CODE.DAT - The resident SYSTEM II code file. This file does not change for any of the scenarios. Other code files include PARTCODE.DAT and SCRNCODE.DAT.

Equivalence files are generally used by SYSTEM II's PERF program to define travel corridors for reporting purposes. There are no specific naming convention for these files, although the first two characters in the filename are generally "EQ".

Interface ASCII (*.ASC, *.OUT) files are used to transfer data between the planning and traffic simulation models. These files provide the main vehicle for sharing network information between models. An example of these files are listed below.

a) LINK3.ASC - Created by a DATAEDIT program, this file contains hourly 1inkfile volume data in ASCII format. The last digit in the filename indicates the applicable study hour. Other files are LINKl.ASC and LINK2.ASC.

b) TURN3.ASC - Created by a DATAEDIT program, this file contains hourly turning movement data in ASCII format. The last digit in the filename indicates the applicable study hour. Other files are TURNl.ASC and TURN2.ASC.

c) HOUR3.0UT - This is one of three hourly speed files that return from the simulation directories. The last digit in the filename indicates the applicable study hour. Other files are HOURl.OUT and HOUR2.0UT.

All print files (*.PRN) are produced during the iterative process and are immediately copied into a scenario results directory. See the Scenario Results (section 3.7) below for more detail on print files.

Set files (*.SET) are SYSTEM II files for the active planning directory and the base scenario directories. For the planning model directory the set files are simply CALIB.SET, CONTROL.SET, AND SETUP.SET. For each of the base scenario directories, the filenames are preceded by the two characters defining the scenario. (e.g. PlCALIB.SET, PlCONTROL.SET, and PlSETUP.SET all apply to the Pl base files.)

3.2 Freeway Simulation Directory (D:\TL\FSIM\*.*)

There are many input (???BAM.INP, P??BAM.OUT) files that are used exclusively to describe a FREQ network. Those files with the .OUT extension are, in fact, FREQ *.INP files as well:

a) PLNBAM.INP - The first two characters of the filename defme the type of FREQ analysis to be performed. The third character defines the direction of traffic to be analyzed. Other files include PLSBAM.INP, F2SBAM.INP, F2NBAM.INP, F3SBAM.INP, F3NBAM.INP, etc.

b) FREQPE.OUT - This is the product of the PRESIM program. It is actually a FREQ .INP file, despite the .OUT extension. This may be either a PE or PL FREQ file, depending on the sixth character in the filename. It is immediately renamed as one of the files defined below in (c).

c) PESBAM.OUT - Formerly a FREQPE.OUT file, this file is a freeway *.INP file with adjusted capacities as calculated by PRESIM. The only variables in the filename are the second and third characters. Other possible filenames are PENBAM.OUT, PLSBAM.OUT, and PLNBAM.OUT.

The output (FR*.OUT, P??BAM.AUX) files are produced by a FREQ run. They exist in either a generic output form or are renamed:

a) FREQl lPL.OUT - Generic form of output that may include performance tables and contour plots. Priority entry analysis would produce a FREQl lPE.OUT file. These files are immediately renamed in the batch file process.

b) FRQPEAUX.OUT - Generic auxiliary output that is automatically produced if requesting (DAY+l) results. May also be FRQPLAUX.OUT. These files are immediately renamed in the batch file process.

c) PENBAM.AUX - Renamed auxiliary output file. Run type (PE or PL) and north or south bound direction (NB or SB) are part of the filename to help defme the file. Other possible filenames include PESBAM.AUX, PLNBAM.AUX and PLSBAM.AUX.

All control (??PRE?B.CTL) files consist of a sequence of filenames and program options, that are read by the PRESIM routine. Each scenario type and direction of traffic has a unique control file:

COPRENB.CTL - The first two characters define the scenario run. The sixth character defmes the direction of traffic. Examples of other control files are PLPRESB.CTL, PEPRENB.CTL, etc.

Freeway equivalence (EQUV???B.PRE, EQUV???B.PST) files are used to coordinate the compilation of data for both the PL and PE FREQ modules:

> EQUVCONB.PRE - All equivalence filenames start with EQW. The fifth and sixth characters identify the scenario as combination (CO), priority lane (PL), or priority entry (PE). The seventh character shows the direction of traffic. The extension defines whether the data is coming into or leaving from FREQ. Other possible combinations include EQUVPLSB.PST, EQUVPENB.PRE, EQWCONB.PST, etc.

Interface ASCII (*.ASC, HOUR?.OUT) tiles, such as, LINK?.ASC and HOUR?.OUT files are used to transfer volume and speed data. There is one unique ASCII file in the FSIM Directory:

> PLPEEQNB.ASC - This file (and PLPEEQSB.ASC) is renamed PLPEEQW.ASC and is called by PRESIM to synchronize the reading of the PL and PE *.INP files.

All executable (*.EXE) files in this directory are strictly FREQ executables. They do not conform to user defined naming conventions.

3.3 Arterial Simulation Directory (D:\TL\ASIM\*.*)

The RUNSYS.BAT and RUNT7F.BAT batch (*.BAT) files adhere to no particular naming convention.

Demo files (DEMO.*) define the simulation method used by the TRANSYT-7F arterial simulation. Filenames remain unchanged from the DEMO Directory.

Turning movement control (TURN?.TMC) files are derived from the DEMO.* files. The digit following TURN in the filename indicates the applicable TRANSYT-7F sub-network.

TRANSYT input sub-network (NET*.TIN) files define the signalized networks. The NET?-n.TIN files are created by NETMAKE.EXE utilizing the base NET?.TIN files. The first number after NET in the filename (?) indicates the TRANSYT-7F network; the second number (n) marks the 15 minute time period evaluated (e.g. “?’ = l-8, and “n” = 1-12).

TRANSYT output (NET?-n.TOF) files consist of simulation data and run information. The first number following NET in the filename (?) indicates the applicable TRANSYT-7F network. The second number (n) marks the 15 minute time period evaluated.

ASCII (*.ASC, *.OUT) files enable the transfer of data into and out of the simulation package. For LINK?.ASC and TURN?.ASC, see the planning directory for more detail.

a) ARTSPD?.OUT and FAQSPD?.OUT ("?" = l-3) both contain hourly speed data. The ARTSPD?.OUT speeds are appended to the HOUR?.OUT file.

b) HOUR3.OUT - This file contains the final speed data from the simulation package. The final digit in the filename defines the hour evaluated. Other files include HOURl.OUT and HOUR20UT.

None of the executable (*.EXE) files in the ASIM **directory** follow a naming scheme.

3.4 **Arterial Simulation Demo Directory (D:\TL\ASIM\DEMO\DEMO.*)**

Demonstration (DEMO-T?.*) files ("?" =l-3) are copied into the active ASIM directory. Only files with identical ? values are copied, depending on the scenario. Once in the active directory, the hyphen, ‘T’, and the '?' value are removed from the filename.

3.5 **Arterial Simulation TIN Directory (D:\TL\ASIM\TIN\NET?.TIN)**

Network (NET?-Ta.TIN) files (? = l-8, and a = l-3) are copied into the active ASIM directory. Only files with identical “a” values are copied, depending on the scenario. Once in the active directory, both the hyphen and the “Ta” value are removed from the filename.

3.6 Base Scenario Directory (D:\TL\PLAN\P?\*.*)

The compressed PKZIP (P?BASE.ZIP) files contain the trip table, link, and turning movement data files. The first two characters of the file name match the name of the corresponding sub-directory.

Data (*.DAT) files are contained within the compressed PKZIP (*ZIP) file and constitute the necessary information to support the base iteration of the simulation process. They are copied into the active directory and renamed as follows:

a) P2_AM3.DAT - The filename for one of three base trip tables. The file is renamed HR3_TRIP.DAT in the active planning directory at the beginning of the iterative process. The digit following AM in the base filename and following HR in the renamed filename indicates the hour of the trip table. Other base trip table files include P2_AM1 .DAT, Pl_AM2.DAT, etc.

b) P2_LINK.DAT - The filename of the base link file that is stored in the base scenario subdirectory. The file is renamed I880LINK.DAT to begin the iterative process. Another example is Pl_LINK.DAT.

c) P2_TRN3.DAT - The name of one of three base turning movement files. This file **is** renamed TURN3.DAT for use in the iterative process. The final digit in both filenames indicates the hour of the turning movement data. Other base turning movement files include P2_TRN 1 .DAT, Pl_TRN2.DAT, etc.

3.7 Scenario Results Directory (D:\TL\PLAN\P?F?T?\*.*)

The files P?F?T?.ZIP contains link, turning movement, and PERF print reports. The file name matches the name of the sub-directory.

Data (*.DAT) files include the link and turning movement files from the eighth iteration of a completed scenario run. Note that the filenames remain unchanged from the active planning directory.

Print (*.PRN) files are many report files created during the iterative process. Generally, these are limited to PERF and ASSIGN reports.

PERF_8.PRN - Print report file that includes area, freeway, and parallel arterial data for each of the three peak hours in iteration 8. The fmal digit in the filename (1-8) indicates the particular iteration.

3.8 Impacts Directory (D:\TL\IMP\*.*)

Interface ASCII (???_RATE.ASC) files consist of rate information for the SAFETY model. The type of rate is identified by the three characters (ACC, INJ, or PDO) that precede the underscore. The last four characters are always RATE.

In this directory the batch (*.BAT) files perform a series of user defined commands. They include DOEMIS, EMISRUN, SAFERUN, EMSCNTRL, IMPACTS, and EMISTRIP.

All control (*.CTL) files listed below contain a list of input parameters or commands:

a) MOB_FUELCTL - The control file for MOBILE5A model. Emission type is identified by the characters following the underscore. Other filenames include MOB-CO.CTL, MOB-HC.CTL and MOB-NOX.CTL

b) SAFE_PDO.CTL- The control file for SAFETY model. Accident type is identified by the characters following the underscore. Other filenames include SAFE-INJ.CTL and SAFE-TOT.CTL.

c) VOL_FUEL.CTL - The control file for the VOLSUM program. Emission or Accident type is identified by the characters following the underscore. Other filename extensions include CO, HC, INJ, NOX, and PDO.

The data (*.DAT, *.EMS) files include equivalence, link, node, zone, and code information. The I880LINK.DAT file is used by the IMPACTS programs to produce and store safety data in the I880LlNK.EMS file. All other naming conventions for data files store safety data in the I880LINK.EMS file. All other naming conventions for data files can be found in the planning directory section.

Examples of executable (*.EXE) files are EMIS.EXE and SAFETY.EXE program files.

Some report (*.PRN) files follow one naming convention described below. Other print files follow no particular naming convention and include: EMRATE95.PRN, STUDAREA.PRN, TOD8AM.PRN, TOD9AM.PRN, TOD10AM.PRN, and VEHTYPE%.PRN.

- The first three characters (IMP or TRP) define the type of impact results within the report. IMP relates to running exhaust, fuel consumption, and accidents. TRP is used for starting emissions for CO, HC, and NOX as well as evaporative emissions for HC.
- The fourth character (M=Mobile5a, E=EMFAC, V=VEHSIME, F=Conventional Fuel Consumption, S=Safety) identifies the model.
- The fifth and sixth characters of the filename (CO=Carbon Monoxide, HC=Hydrocarbons, NO=Nitrogen Oxides, AL=Same as TRP, FL=VEHSIME Fuel Consumption, UE=Conventional Fuel, PD=Property Damage Accidents, IN=Injury Accidents, AC=Total Accidents) identify the emission, fuel, or accident type.
- The seventh character is always “A” for arterials.
- The eighth digit (l-4) signifies the hour being reported, with “4” used to report all three hours.

Set (*.SET, *.FNT) files for the impacts directory begin with IMP followed by either CALIB, CNTRL, or SETUP. The SYSTEM96.FNT file sets the font used by SYSTEM II’s GEDIT program.

Compressed PKZIP (*ZIP) files consist of IMPSET.ZIP, I880EMIS.ZIP, and USERCODE.ZIP. The first two files are backups of the SET and link information. USERCODE defines the link types currently found in the .EMS file.

4. Freeway Initialization and Setup

The FREQ11 model is considered to be the most appropriate freeway traffic simulation available for inclusion into the model framework for several reasons. Other freeway simulation models that were examined have limitations related to freeway length, representation of ramp metering and HOV lanes, computer and data requirements, and a limited history of application development. In contrast, FREQ can analyze freeway segments of acceptable length, it has reasonable data input requirements, and has adequate capability to model ramp metering and HOV lanes. FREQ is best suited to corridor evaluations covering five or more miles of freeway, and permits cost/benefit analysis for various geometric alternatives, evaluating traffic management and traffic control alternatives (such as incident management, ramp metering, mainline HOV lanes, and HOV bypass lanes at ramp meters).

The FREQ model can be exercised once geometric and speed curve data have been specified. There are several FREQ program screens that enable the user to select specific options to specify the unique characteristics of input files FREQPL.INP and FREQPE.INP. These files define the freeway network and specify traffic simulation execution options for each iteration. There are two ways of editing any FREQ input file. The first is by going through the FREQ program interface, screen by screen, incorporating modifications at the appropriate place or by editing an existing input file (*.INP) using a commercially available editor utility.

Table C-1: Equivalence (EQUV*.PRE) Data File

Character	Link Attribute Name	Valid Values
1-12	FREQ Link Segment Number	1 - 1,000
13-21	Ramp Type	1 = on-ramp 2 = off-ramp
22-30	Volume Type	0 = mixed flow 1 = carpool
31-39	A-NODE (Planning Model)	1 - 32,000
40-48	B-NODE (Planning Model)	1 - 32,000
49-57	Factor (*100) used to divide the planning model volume into 'n' FREQ segment volumes	0 - 100

The FREQ model, during and iteration of the framework, creates two output files called FREQPL.OUT and FREQPE.OUT that consist of volume and speed data. These files contain simulation results for HOV lane analysis and ramp metering respectively. The files are only created if the particular ATMS user services, simulated by FREQPL or FREQPE, were enabled.

The EQW*.PRE equivalence file is one of the most important data files necessary to conduct the freeway simulation. This file contains data that is used to match planning model network links with the corresponding freeway (FREQ) link segments. (Refer to figure C-l for a complete definition of all fields contained in this fixed length record file.)

4.1 Calibration

Calibration of a traffic model involves adjusting several parameters to reach a desired calibration result. The FREQ calibration process can be considered to be a two step process. First the user creates a calibrated data set for existing baseline conditions, and secondly, the user then produces a calibrated FREQ*.lNP input file, simulating traffic conditions, after the ATMS improvement has been added to the model.

In order to create a calibrated data set for existing conditions, the user first runs the FREQ interface. Input variables including ramp counts and freeway/ramp geometrics will be entered in the normal interactive way expected by the FREQ interface. A number of FREQ model runs will be made with the user making modifications to such variables as speed curves and capacities, until the model is calibrated and is producing results that are consistent with actual existing conditions. The fmal data set that produces a calibrated run will be saved and is referred to as the “baseline calibrated data set”.

The baseline calibrated data set is then used to make a subsequent FREQ (FREQPL or FREQPE) run. The user exercises the FREQ interface retrieving the baseline calibrated data set and then selects the FREQPL or FREQPE capability (reflecting the operational design for an ATMS improvement; i.e. ramp metering, HOV lane, etc.). The new FREQ input file created by the program during this operation is saved and referred to as the “calibrated data set”. Upon completion of the second calibration step, the FREQ model will accurately simulate the corridor, reflecting ATMS improvements, before drivers have changed their driving behavior (i.e. all vehicles have the same time, space and occupancy patterns as before). When this second calibration has been successfully completed, and discrepancies resolved against baseline conditions, the spatial and modal shifts can be applied to simulate new traffic patterns that emerge from taking advantage of modeled ATMS services.[2]

[2] Dimitris Scapinakis et al., “Demand Estimation, Benefit Assessment, and Evaluation of On-Freeway HOV Lanes”, Institute of Transportation Studies University of California at Berkeley, Research Report UCB-ITS-RR-91-6, June 1991.

A calibrated FREQ input file is used by the FREQ model during the iterative process between the FREQ model and the planning model. At each iteration the planning model provides FREQ with new ramp counts for each time slice and new occupancy values for each origin. These values will replace those that are in the calibrated input file. FREQ then produces new average speed and ramp delay tables for each time slice, and this iterative process continues until convergence is achieved.

For calibration purposes the model requires information on freeway section speeds during all analysis time periods. Speed data are usually obtained via tachometer runs and are then summarized in a speed-contour diagram for each analysis period. Information on observed bottleneck capacities in the corridor are also required for model calibration.

4.2 Modification of Existing Data

Rather than create a new data set, the user has the option to retrieve and use an existing input data set by selecting FREQ main menu option 3 (Retrieve ***data set from disk file).*** The user may select a data set by either specifying the file name, or by obtaining a listing of existing data files by pressing the ***"Enter"*** key. To select a file from the list, the user may scroll down the list of files and strike the ***"Enter"*** key when the desired file is highlighted.

To modify an existing data set, the user must select option 4 from the FREQ Main Menu. If selected, a second sub-menu with a listing of various input/modification options will appear. The most commonly used modification choices are either geometric data or speed curves. To alter geometric data, the user should choose (2)-***Modify Freeway*** Design. Once in this template, the user may alter several variables, including number of lanes, ramp and mainline capacities, segment lengths, and individual segment speed curves.

The speed curves as well as the geometric characteristics of the freeway, must be consistent between the simulation and planning models. To alter the default speed curves for freeway mixed-flow lanes, the user should choose sub-menu option (5)-***Input/Modify Arterial and Other Optional Data.*** To alter the speed curves for HOV lanes, the user should choose sub-menu option (6)-***Input/Modify Data for HOV Operational Design,*** followed by (Function Key 3)-***Optional HOV Design Data.***

FREQ operates on the basis of speed/volume and demand/capacity relationships. The classic speed/volume relationship is used to compute speed for all subsections. Several curves are resident in the program, one of which can be selected for any simulation, or a user-developed curve can be specified. The curves resident in the program were derived from multiple sources (mainly Caltrans and FHWA) and can be selected by choosing the appropriate free flow speed. For most freeways the 55 mph curve is appropriate.

4.3 Input Parameters

Several parameters are selected by the user using the FREQ menu interface and these parameters are saved in the *.lNP files. During simulation the FREQ routines will access *.INP files and use input parameters to direct the freeway simulation process.

Before conducting a FREQ run, the user is asked to define the specific report options for the print screens. The information produced by the minimum run options are the only data that must be in an *.OUT file for the interface to perform properly. Therefore, since additional report options will slow the FREQ run processing, the user should only select subsequent run options to provide necessary detail about a particular run.

HOV demand is input as a percentage of total demand at each entry point. The user specifies percentage of SOVs, HOV2, and HOV3+ vehicles. FREQ does not use a separate trip table for each vehicle type, rather it uses proportional relationships to determine HOV volumes.

Within the FREQ interface the user is asked if “tilting” of the data will be necessary. In the context of the interface, although travel through a freeway segment may take longer than one time slice, the user should not employ “tilting” of the data. The reason for turning off tilting is primarily to maintain consistency of evaluation between the planning and simulation models. By employing tilting, the simulation model would evaluate the entire freeway corridor as one element rather than as individual freeway links and ramps at a single moment in time. Since the planning model determines trips and volumes on single links throughout the entire network area at any one time, FREQ should be coded to attempt a similar evaluation. This is best approximated by not employing “tilting”.

5. Arterial Initialization and Setup

This section describes how to define parameters and options for setting up the arterial traffic simulation. Prior to running the framework, data files describing peak spreading factors, link-node equivalence and traffic control strategies are required. The SYSEDIT user interface utility was created to allow the user to define these parameters and select various options. This menu driven program is executed by entering the module name "*SYSEDIT*" at the DOS prompt and hitting "*Enter*".

Table C-2: Master Control File (DEMO.MCF) Format

Record Number	Description
1*	Number of Networks, Traffic Control Type, Volume File Type, Peak Hour Number
2	No. of Intersections in Network 1
3	No. of Intersections in Network 2
...	...
26	No. of Intersections in Network 25
27*	TRANSYT-7F Northbound Link Codes (Left, Through, Right)
28*	TRANSYT-7F Southbound Link Codes (Left, Through, Right)
29*	TRANSYT-7F Eastbound Link Codes (Left, Through, Right)
30*	TRANSYT-7F Westbound Link Codes (Left, Through, Right)

* (comma delimited data in this record)

The SYSEDIT module is a user interface tool for editing and storing peak spreading percentages, link-node association data and turning movement link orientation codes. SYSEDIT is also used to describe parameters necessary to control the iteration process and to select the type of traffic control simulation performed on arterial streets of the study area.

A master control file (DEMO.MCF) is used to define the number of signalized arterial sub-networks, the type of traffic control system to be simulated, the planning model turning movement database file format, the planning model time period number that represents the peak period, the number of intersections per sub-network, and the TRANSYT-7F link turning movement direction codes. Table C-2 indicates the relationship between the various fields described above that together form the DEMO.MCF data file.

The SYSEDIT tool also creates a peak spreading data file (DEMO.PSD) that contains peak spreading percentages which are used to generate turning movement volumes for shorter time intervals than those produced by the planning model which represents a larger period of time. The peak spreading data file DEMO.PSD consists of comma delimited ASCII alphanumeric data that is used to divide planning model peak hour volumes into smaller fifteen minute volumes. The DEMO.PSD file, therefore, provides the conversion link between planning model time interval turning movements and the smaller time interval volumes used for traffic simulation. Table C-3 illustrates the format of the DEMO.PSD file.

Table C-3: Peak Spreading Data (DEMO.PSD) File Format

Record Number*	Description
1	Length of Time Period (min.), No. of Time Periods, No. of Intervals per Period, No. of Hours of Volume Data, Length of Simulation Time Interval (min.), No. of Time Intervals
2	Peak Spreading Factors for Intervals 1 through 5
3	Peak Spreading Factors for Intervals 6 through 10
4	Peak Spreading Factors for Intervals 11 through 15
5	Peak Spreading Factors for Intervals 16 through 20
6	Peak Spreading Factors for Intervals 21 through 24

* (comma delimited data in this record)

The link-node association file (DEMO.LNA), created by SYSEDIT, describes which TRANSYT-7F sub-network a planning model node will be assigned to, and what adjacent nodes will feed traffic volumes into each intersection. With this information, it is possible to associate intersection nodes between the planning model and TRANSYT-

7F, and also to determine the origin of traffic feeding each TRANSYT-7F link. This latter information is required by TRANSYT-7F to properly model progression. The file format for link-node association is represented in table C-4. The link-node association file consists of a list of node numbers that associate the signalized intersections, approaches and adjacent signalized intersections of each TRANSYT-7F sub-network with the associated planning model intersections. All data in the DEMO.LNA file is comma delimited ASCII alphanumeric information.

Table C-4: Link-Node Association (DEMO.LNA) File Format

Record, Field Position	Description
1, 1	**TRANSYT-7F Signal System Number**
1,2	TRANSYT-7F Node Number
1, 3	SYSTEM II Intersection Node Number
2, 1	Northbound Approach Origin Node Number
2, 2	Southbound Approach Origin Node Number
2, 3	Eastbound Approach Origin Node Number
2, 4	Westbound Approach Origin Node Number
3, 1	Northbound Adjacent Signalized Intersection Node Number
3, 2	Southbound Adjacent Signalized Intersection Node Number
3, 3	Eastbound Adjacent Signalized Intersection Node Number
3, 4	Westbound Adjacent Signalized Intersection Node Number

5.1 Peak Spreading Factors

The peak spreading factors are used to divide volumes provided by the planning model assignments into shorter time intervals for a more detailed simulation. The turning movement volumes from each planning model time period are multiplied by the peak spreading factors to model the increase or decrease in 15-minute flow rates that occur during the peak period time span. Normally, turning movement volumes from the planning model assignment will represent a one-hour time period. To divide this assignment up into flow rates for 15-minute periods, peak spreading factors are used to compute hourly flow rates for each 15minute period. The peak spreading factors should be normalized such that the average of the factors over each one-hour period is 1 .00. This is necessary so that 15-minute volumes are expressed in terms of a hourly flow rate, which is required by TRANSYT-7F. Table C-5 below contains a matrix of peak spreading factors, used for the demonstration study area, which is based on twelve simulation time intervals. Data in table C-5 also shows the percentage of the daily trips assigned during each of the planning model time periods, and the non-normalized peak spreading percentages.

Table C-5: Peak Spreading Factors for
Arterial Streets in Demonstration Study Area

Planning Model Time Period	Percent of Trip Table Assigned	Simulation Time Interval	Peak Spreading Percentages	Normalized Peak Spreading Factors
1	4.32%	1	23%	0.92
		2	24%	0.96
		3	26%	1.04
		4	27%	1.08
2	5.27%	5	24%	0.96
		6	26%	1.04
		7	26%	1.04
		8	24%	0.96
3	5.01%	9	26%	1.04
		10	25%	1.00
		11	25%	1.00
		12	24%	0.96

Peak spreading factors are edited under the ***SYSEDIT*** program main menu option 1 ***(View/Edit Peak Spreading Data),*** submenu option *2 **(Edit Interval Peak Spreading Percentages).*** Note that when properly normalized, the peak spreading percentages should add up to equal the number of simulation time intervals, and the average peak spreading percentage should be exactly one.

5.2 Link-Node Association

In order to associate planning model link and node numbers with TRANSIT-7F link and node numbers, an equivalence database is needed. The link node associations also establish the primary direction of each intersection approach, since the user must specify whether an approach is northbound, southbound, eastbound or westbound. In order to identify which planning model links make up the approaches to each intersection, the nearest adjacent nodes on each approach must be specified. In order to describe the spatial relationship among signalized intersections, the nearest adjacent signalized intersection node on each approach must also be specified. Note that the nearest adjacent node may or may not be the same as the nearest adjacent signalized intersection node. In many cases the nearest adjacent node will be a centroid connector node, while the nearest signalized intersection may be several nodes beyond the nearest adjacent node. Node numbers for adjacent signalized intersections that are not part of the coordinated signal network should not be specified. In general, the matrix of adjacent approach nodes includes all adjacent nodes inside and outside the signal network while the matrix of adjacent signalized intersections includes only those nodes within the signal network boundary.

Link-node association data is edited under ***SYSEDIT*** main menu option ***2 (View/Edit Link-Node Association Data).*** The resulting screen display will consist of a three dimensional array for association data. The ***Home*** and ***End*** keys are used to view the link-node association data of up to 25 different traffic signal networks. The ***Page-Up*** and ***Page-Down*** keys are used to view the next or previous ten intersections that are part of the current signal network. ***Arrow*** keys are used to edit planning model node numbers. The ***Escape*** key is used to exit the menu. Note that all intersections that are part of a particular network must be numbered consecutively starting with the number one.

5.3 Traffic Control Options

The arterial street simulation modeling framework is designed to accommodate four different traffic control implementation strategies. The first option is to simulate all signal networks for all time periods. This strategy assumes that the appropriate signal timing, phasing, cycle length and offsets have been previously set in the base TRANSYT-7F networks outside of the automated modeling process. The second option is to optimize the cycle length and timing for the peak interval of the peak period of planning model traffic volume data, then simulate traffic operations during all other time intervals assuming that the timing and offsets do not change. This option simulates the impacts of a fixed time coordinated traffic signal network designed to accommodate the peak interval. The third option is to optimize cycle length and timing for the peak interval of every period of planning model traffic volume data, then simulate traffic operations during all other time intervals within each period using the same timing and offsets. This option may be useful for adjusting the traffic signal timing plan to accommodate the different traffic patterns produced by each planning model assignment. The fourth option is to optimize cycle length, offsets and timing during every interval of every period of planning model traffic volume data. This option is designed to simulate real time traffic control. Cycle lengths, offsets and splits are optimized for each time interval of the overall simulation time span.

The traffic control strategy is selected from the ***SYSEDIT*** main menu option ***3 (View/Edit TRAZVSYT-7F Optimization Control Parameters). This*** option is also used to define the format of the traffic turning movement volume data coming from the planning model, and to identify the planning model time period representing the peak hour.

5.4 TRANSYT-7F Link Orientation

The TRANSYT-7F program was designed to use a link and node numbering scheme that does not force individual turning movements into a standard cardinal direction turning movement scheme used by most intersection analysis software. For example, with most other applications, turning movements are entered as eastbound left, or southbound through, etc. With TRANSYT-7F, the user can define the relationship between links in a generic fashion so that it is possible to have north-westbound left, south-westbound through, northbound U-turn, or any other unusual combination of cardinal directions or turn destinations. In fact, all platoon travel paths in TRANSYT-7F are defined in such a way that it is not even necessary to recognize the turning movement cardinal direction or turn direction.

Another attribute of TRANSIT-7F is that a typical intersection approach may consist of up to five approach links, as opposed to only one for planning model links. In TRANSYT-7F, approach links are divided up in much the same way that intersection approach lane groups are defined when performing an intersection capacity analysis based on Highway Capacity Manual procedures. Most intersection approaches will include at least two approach lane groups; one for left turns and one for through and right turns. Where exclusive right turn lanes and/or exclusive right turn phasing are present, an additional lane group may be added for right turns only. The arterial street modeling framework was designed to recognize up to three TRANSYT-7F links per approach and up to four approaches. These links are used to represent left, through, and right turn movements for the four major cardinal directions.

Since intersection turning movement volumes provided by planning models are generally organized using the standard labeling scheme (northbound left, through, and right, etc.), a link orientation standard for TRANSYT-7F link numbers is needed so that turning movements from the planning model, which are identified by approach direction and turn direction, may be associated with the last two digits of TRANSYT-7F link number. The primary purpose of the link orientation convention is to standardize the link numbering scheme used to prepare base TRANSYT-7F networks so that individual link volumes may be replaced by the appropriate turning movement volumes.

The table C-6 below shows two common link numbering conventions. The National Electrical Manufacturers Association (NEMA) style convention is typical of traffic signal controller phase movement numbers and is also the convention recommended in the

TRANSYT-7F documentation. A second option, the orientation convention, is generally easier to decipher, since the first digit represents the cardinal direction and the second digit represents the turning movement. Any link orientation convention may be used, however, all link numbers the TRANSYT-7F networks must conform to this orientation convention.

The link orientation codes can be edited under the ***SYSEDIT*** main menu option 4 ***(View/Edit Intersection Approach Lane Groups).*** The codes are modified in a sequential fashion.

Table C-6: Typical Conventions for
Turnina Movement Direction Link Codes

Turning Movement Direction	NEMA Style Convention	Orientation Convention
Northbound Left	2	10
Northbound Through	1	11
Northbound Right	9	12
Southbound Left	4	20
Southbound Through	3	21
Southbound Right	10	22
Eastbound Left	6	30
Eastbound Through	5	31
Eastbound Right	11	32
Westbound Left	8	40
Westbound Through	7	41
Westbound Right	12	42

5.5 File Operations

All data entered or edited by the ***SYSEDIT*** tool can be saved or loaded from main menu option 5 ***(File Operations).*** This menu option produces a submenu that allows loading or saving of the DEMO.PSD peak spreading data file, the DEMO.LNA link-node association data file and the DEMO.MCF master control file. A file list option is also available to view current files on the disk.

5.6 Calibration of TRANSYT-7F

TRANSYT-7F is normally used by traffic engineering professionals to design traffic signal system timing plans. Since the framework is using the TRANSYT-7F program as a traffic simulator, there are different calibration requirements than those necessary for using the program to only determine signal timing.

Two of the most important TRANSYT-7F parameters that need calibration for realistic results include the default saturation flow rates for intersection approach lane groups and the average running speed of vehicles between intersections. The saturation flow rates that are common for a particular study area may be obtained through a random survey of selected intersection approach lane groups. At least six different approaches should be surveyed. Saturation flow rates should also be measured for different lane group types. These include left turn lanes, multiple left turn lanes, through lanes, shared through and right turn lanes, and exclusive right turn lanes. The results of the saturation flow rate study are used as a guideline to define the saturation flow rates in TRANSYT-7F input files. Special care should be taken when coding intersection approaches that serve high pedestrian volumes. Pedestrians can have a significant effect on the saturation flow rates of permitted turning movements, especially right turns.

Average running speed measurements were taken on all major arterials and expressways that are part of the study area signalized sub-networks. At least three runs were obtained in each direction. The average running speed should not include signalized intersection delays, but should incorporate delays associated with speed zones, on-street parking maneuvers, bus operations, vehicles slowing down to access driveways and minor side streets, and other common mid-block factors that affect the travel time of vehicles between signalized intersections.

If the study area network is to include intersections that are not currently coordinated and/or signalized, then an evaluation of phasing patterns should be performed to determine the most appropriate pattern based on the functional class of the intersecting roadways, the relative spacing of intersections, and expected demands for turning movements (especially left turns). Timing plan evaluations should consider whether the intersection should operate with fixed time, semi-actuated, or fully actuated control. Phasing scheme evaluations should compare leading versus lagging left turn phasing; protected, permitted, or protected-permitted turn phasing; turn prohibitions; minimum green requirements for pedestrians; and the need for protected right turn phasing. In situations where a particular arterial corridor may be used as a bypass around other congested routes, roadway restriping schemes or minor widening could also be considered to increase the capacity of travel paths.

6. TRANSYT-7F Network Representation

Since TRANSYT-7F requires a variety of geometry, phasing, timing and progression parameters not provided by a planning model, the base NET*.TlN input files represent all traffic signal networks in the study area and need to be coded before setting up the framework. TRANSYT-7F networks can be coded using either a text editor or a commercially available input editor. It is not necessary to include traffic volumes with the coded data sets, since the modeling framework automatically imports planning model volumes into each data set. However, it may be desirable to enter a set of volumes for the purpose of testing phasing parameters, and debugging data sets.

6.1 TRANSYT-7F Data Management Utilities

The TRANSYT-7F software program is an executable module that reads all input parameters from an input file and prints all results to one or more output files. Other than defining filenames and output destinations, the program does not provide interactive data entry. Therefore, TRANSYT-7F data sets must be coded off-line before running a simulation or optimization.

The TRANSYT-7F program comes with a data input manager called T7FDIM. Other available input managers include the Arterial Analysis Package (AAP) from the McTrans Center for Microcomputers in Transportation, or commercial packages such as EZ-TRANSYT (PC Trans), PRE-TRANSYT (Strong Concepts, Inc) and others. TRANSYT-7F input files could also be created using any ASCII text editor. However, the data input managers provide numerous benefits including interactive data entry, editing, help screens, and error checking.

6.2 Guidelines for Coding TRANSYT-7F Networks

Base TRANSYT-7F data input files NET*.TIN must be prepared prior to running the arterial street simulation modeling process. During the process of selecting a study area, for benefits assessment, key arterial streets and expressways should be identified for inclusion in the arterial street network of the study area. Once these streets and expressways are identified, the network must be broken down into a series of corridor and grid sub-networks.

Existing signal networks should generally be based on the networks that are currently in operation. Future signal networks should be based on the proximity of new and existing signal networks and other factors discussed later. In general, signal networks should be separated where there is more than a one mile spacing between adjacent signals. Also, networks must be separated if a certain group of signalized intersections are likely to require a longer or shorter cycle length than other nearby signals.

Each NET*.TIN network file can accommodate up to 50 signalized intersections. Since each network must operate on a common cycle length, the decision regarding how to break up the overall study area into smaller networks, based on cycle length requirements, must be made in advance. The arterial street modeling framework is designed to accommodate up to 25 networks, or up to 1,250 traffic signals.

Since TRANSYT-7F does not optimize phasing plans, the base NET*.TIN networks should be tested to determine whether the phasing plans are appropriate. This may not be necessary if all traffic signals are existing because phasing plans are already in use. However, if new signals are to be added, or if existing signals that are not currently tied to a signal system are to be included, then phasing plan options should be investigated to ensure that the patterns selected provide the best opportunities for good progression. Also, timing and phasing parameters must be provided with enough flexibility to allow for sizable variations in traffic volumes for different turning movements. An over-constrained timing plan could prevent the flexibility needed to model real time traffic control strategies with the overall framework.

Since cycle length optimizations are among the most time consuming parts of running TRANSYT-7F, it would be beneficial to run the arterial street modeling framework on a trial basis for the purpose of reducing the number of cycle lengths to be tested. This function could be performed after one iteration of the modeling framework. By summarizing the optimum cycle lengths selected by TRANSYT-7F, for each network and time interval, it is possible to determine whether the cycle length is sensitive to the variations in demand between intervals. If very little variation is present, then the range of cycle lengths tested by different networks can be reduced to lower the overall framework run time.

It should be noted that the framework cannot accommodate overlapped traffic signal networks. In other words, intersections that are part of one network can not be part of another adjacent network. Though overlapping is a necessary procedure when attempting to coordinate large signal networks of more than 50 intersections using TRANSYT-7F, there is not an automated method that can be used to perform this procedure within the framework without user interaction.

Before running the modeling framework, all NET*.TIN network files should be tested using a full cycle length-offset-split optimization to ensure that no errors are present in the data set. These errors include syntax and typographical errors and common errors that occur when setting up signal timing plans or selecting user defined options. The TRANSYT-7F User’s Manual includes a comprehensive list of common errors and how to resolve them. Avoid using any TRANSYT-7F options that could affect the results of a model framework run, or that cannot be addressed by the modeling framework. These options include flow multipliers, transit links, flow weighting, coding intersections with more than four legs or coding intersection approaches with more than three links. Otherwise, any other TRANSYT-7F parameters can be adjusted to suite the needs of a framework scenario run.

7. Framework File Formats

Planning Model Turning Movement Volume File Format
(TURNa.ASC)

Column Ranges	Description	Valid Values
1-8	Planning Model From Node Number	1 - 32,000
9-16	Planning Model Through Node Number	1 - 32,000
17-24	Planning Model To Node Number	1 - 32,000
25-48	Not Used	N/A
49-56	Turning Movement Volume	Integer value

[From-Through-To Node Format]

Turning Movement Volume File Format
(TURNb.TMV)

'Record Field Position	Description	Valid Value
1	TRANSIT-7F Node Number	1 - 1,250
2	Time Interval Number	1 - 12
3	Northbound Left Volume	integer value
4	Northbound Through Volume	integer value
5	Northbound Right Volume	integer value
6	Southbound Left Volume	integer value
7	Southbound Through Volume	integer value
8	Southbound Right Volume	integer value
9	Eastbound Left Volume	integer value
10	Eastbound Through Volume	integer value
11	Eastbound Right Volume	integer value
12	Westbound Left Volume	integer value
13	Westbound Through Volume	integer value
14	Westbound Right Volume	integer value

(Sequential File Type, record fields are comma delimited)

Planning Model Link Data Format
(LINKa.ASC)

Character	Link Attribute Name	Valid Values
l-8	Record Number	1 - 999,999 sequential integer
9-16	Planning Model From (A) Node Number	1 - 32,000
17-24	Planning Model To (B) Node Number	1 - 32,000
25-32	Planning Model Direction Flag	0 = Node A to node B 1 = Node B to node A
33-40	Functional Class ***Code***	0 = Zone Connector 1 = Freeway 2 = Loop Ramp 3 = Direct Ramp 4 = Bridge Toll 5 = Toll Paying Direction 6 = Expressway 7 = Divided Arterial 8 = Undivided Arterial 9 = Collector 10 = Freeway Outside Study Area 1 1 = Expressway outside Study Area 12 = Divided Arterial Outside Study Area 13 = Undivided Arterial outside Study Area 14 = HOV Carpool Facility 15 = HOV Meter By-pass Ramp

41-48	General Purpose HOV Facility Flag	o = s o v
		l = HOV

49-56	Length of Link in Miles	Miles value to 2 decimal places
57-64	Link Free Flow Speed	Speed value to 2 decimal places
65-72	Planning Model Through Lanes (One Direction)	integer value, by direction
73-80	Planning Model Capacity per Through Lane in VPH	integer value
81-88	Planning Model Loaded One-Way Volume	integer value

89-96	Planning Model Loaded Travel Speed in MPH	Speed value to 2 decimal places
97-104	Planning Model Area Type	1 = Core 2 = Central Business District 3 = Outlying Central Business District 4 = Residential 5=Rural 6 = Undefined

Peak Factor File
(PEAKFAC.ASC)

Record	Description	Value Used
1	Percent of Pre-Peak Hour	90
2	Percent of Peak Hour	100
3	Percent of Post-Peak Hour	80

Equivalence Data File
(EQUV*.PRE)

Character	Link Attribute Name	Valid Values
1-12	FREQ Link Segment Number	1 - 1,000
13-21	Ramp Type	1 = on-ramp 2 = off-ramp
22-30	Volume Type	0 = mixed flow 1 = carpool
3 l-39	A-NODE (Planning Model)	1 - 32,000
40-48	B-NODE (Planning Model)	1 - 32,000
49-57	Factor (* 100) used to divide the planning model volume into ‘ n’ FREQ segment volumes	0-100

Time Slice Distribution Factor File (FACTORS.DAT)

Record Number	Description	Field*	Factor Value
1	Freeway On-Ramp Time Slice Distribution Factors for 15-minute volumes	1	24
		2	25
		3	25
		4	26
		5	24
		6	26
		7	26
		8	24
		9	26
		10	25
		11	25
		12	24
2	Arterial Time Slice Distribution Factors for 15-minute volumes	1	23
		2	24
		3	26
		4	27
		5	24
		6	26
		7	26
		8	24
		9	26
		10	25
		11	25
		12	24

* Fields within each record are comma delimited.

Arterial Street Speed File Format
A
(ARTSPDa.001)

Columns	Description	Valid Values
l-8	Planning Model Record Number	1 - 999,999 sequential integer
9-16	Planning Model From (A) Node Number	1 - 32,000
17-24	Planning Model To (B) Node Number	1 - 32,000
25-32	Planning Model Direction Flag	0 = Node A to node B 1 = Node B to node A
33-40	Planning Model Functional Class Code (facility type)	0 = Zone Connector 1 = Freeway 2 = Loop Ramp 3 = Direct Ramp 4 = Bridge Toll 5 = Toll Paying Direction 6 = Expressway 7 = Divided Arterial 8 = Undivided Arterial 9 = Collector 10 = Freeway Outside Study Area 1 1 = Expressway outside Study Area 12 = Divided Arterial Outside Study Area 13 = Undivided Arterial outside Study Area 14 = HOV Carpool Facility 15 = HOV Meter By-pass Ramp

41-48	Planning Model General Purpose/HOV Facility Flag	o = s o v
		l=HOV

49-56	Planning Model Length of Link in Miles	Miles value to 2 decimal places
57-64	Planning Model Free Flow Speed in MPH	Speed value to 2 decimal places
65-72	Planning Model Through Lanes per Direction	integer value

73-80	Planning Model Capacity per Through Lane in VPH	integer value
81-88	Planning Model Loaded One-Way Volume	integer value
89-96	Simulated Average Travel Speed in MPH (loaded speed)	Speed value to 2 decimal places
97-104	Planning Model Area Type	1 = Core 2 = Central Business District 3 = Outlying Central Business District 4 = Residential 5=Rural 6 = Undefined

Link Results File for Safety Data (SAFETYa.OUT)

Character	Description	Valid Values
1-8	Planning Model Record Number	1 - 999,999 sequential integer
9-16	Planning Model From (A) Node Number	1 - 32,000
17-24	Planning Model To (B) Node Number	1 - 32,000
25-32	Planning Model Direction Flag	0 = Node A to node B 1 = Node B to node A
33-42	Safety data for either Property Damage Only (PDO) Accidents, Injury Accidents, or Total Accidents.	(Accident value)* 1,000,000

Fuel & Air Quality Analysis File
FAQSPDa.OUT

Column	Description	Valid Values
l-8	Planning Model Record Number	1 - 999,999 sequential integer
9-16	Planning Model From (A) Node Number	1 - 32,000
17-24	Planning Model To (B) Node Number	1 - 32,000
25-32	Planning Model Direction Flag	0 = Node A to node B 1 = Node B to node A
33-40	Planning Model Length of Link	Miles value to 2 decimal places
41-48	TRANSYT-7F Simulated Average Travel Speed in MPH	Speed value to 2 decimal places
49-56	TRANSYT-7F Coded Free Flow Speed in MPH	Speed value to 2 decimal places
57-64	Planning Model Through Lanes per Direction	integer value
65-72	Planning Model Loaded One-Way Volume	integer value
73-80	TRANSYT-7F Intersection Stops	integer value
81-88	TRANSIT-7F Vehicle Miles of Travel	Miles value to 2 decimal places
89-96	TRANSYT-7F Simulated Vehicle Hours of Travel	Hours value to 2 decimal places
97-104	TRANSYT-7F Simulated Vehicle Hours of Delay	Hours value to 2 decimal places
105-l 12	Planning Model Area Type	1 = Core 2 = Central Business District 3 = Outlying Central Business District 4 = Residential 5=Rural 6 = Undefined
113-120	TRANSYT-7F Fuel Consumption	Fuel value in gallons
121-128	TRANSYT-7F Carbon Monoxide Emissions in Grams	Emission value in grams/hour

129-136	TRANSYT-7F Hydrocarbon Emissions in Grams	Emission value in grams/hour
137-144	TRANSYT-7F Nitrogen Oxide Emissions in Grams	Emission value in grams/hour

Link Results File for Emission, Fuel Consumption and Accident Data
(HOURa.EMS)

Character	Description	Valid Values
1-8	Planning Model Record Number	1 - 999,999 sequential integer
9-16	Planning Model From (A) Node Number	1 - 32,000
17-24	Planning Model To (B) Node Number	1 - 32,000
25-32	Planning Model Direction Flag	0 = Node A to node B 1 = Node B to node A
33-39	VEHSIME Carbon Monoxide	Emission value in grams/hour
40-46	VEHSIME Hydrocarbons	Emission value in grams/hour
47-53	VEHSIME Oxides of Nitrogen	Emission value in grams/hour
54-60	VEHSlME Fuel Consumption	Emission value in grams/hour
61-67	MOBILE 5a Carbon Monoxide	Emission value in grams/hour
68-74	MOBILE 5a Hydrocarbons	Emission value in grams/hour
75-8 1	MOBILE 5a Oxides of Nitrogen	Emission value in grams/hour
82-88	EMFAC 7F Carbon Monoxide	Emission value in grams/hour
89-95	EMFAC 7F Hydrocarbons	Emission value in grams/hour
96-102	EMFAC 7F Oxides of Nitrogen	Emission value in grams/hour
103-108	Conventional Fuel Consumption	Fuel value in eallons

Freeway Speed File Format
(HOURa.OUT)

Character	Description	Valid Values
1-8	Planning Model Record Number	1 - 999,999 sequential integer
9-16	Planning Model From (A) Node Number	1 - 32,000
17-24	Planning Model To (B) Node Number	I- 32,000
25-32	Planning Model Direction Flag	3 = Node A to node B 1 = Node B to node A
33-40	Planning Model Functional Class Code (facility type)	3 = Zone Connector 1 = Freeway 2 = Loop Ramp 3 = Direct Ramp 4 = Bridge Toll 5 = Toll Paying Direction 6 = Expressway 7 = Divided Arterial 8 = Undivided Arterial 9 = Collector 10 = Freeway Outside Study Area 11 = Expressway outside Study Area 12 = Divided Arterial Outside Study Area 13 = Undivided Arterial outside Study Area 14 = HOV Carpool Facility 15 = HOV Meter By-pass Ramp

41-48	Planning Model General Purpose/HOV Facility Flag	o = s o v
		l=HOV

49-56	Planning Model Length of Link	Miles value to 2 decimal places
57-64	Planning Model Free Flow Speed in MPH	Speed value to 2 decimal places
65-72	Planning Model Through Lanes per Direction	integer value
73-80	Planning Model Capacity per Through Lane in VPH	integer value

81-88	Planning Model Loaded One-Way Volume	integer value
89-96	Simulated Average Travel Speed in MPH (loaded speed)	Speed value to 2 decimal places
97-104	Planning Model Area Type	1 =core 2 = Central Business District **3** = Outlying Central Business District 4 = Residential 5=Rural 6 = Undefined

Vehicle Type Distribution File
(VEHTYPE%.PRN)

Characters	Description	Valid Values
1-3	Area Type 1	1 = Central business district 2 = Commercial 3 = Suburban 4 = Rural
4-6	Area Type 2	Same as Area Type 1
7-9	Facility Type 1	1 = Zone Connector 2 = Freeway Merge 3 = Freeway Straight 4 = Freeway Weave 5 = Off Ramp 6 = On Ramp 7 = Arterial Collector
10-12	Facility Type 2	Same as Facility Type 1
13-16	Time Period	1 = Pre-peak 2 = Peak 3 = Post-peak
17-24	Light Duty Gasoline Auto	Vehicle type factor values
25-32	Light Duty Diesel Auto	Vehicle type factor values
33-40	Light Duty Truck	Vehicle type factor values
41-48	Medium Duty Truck	Vehicle type factor values
49-56	Heavy Duty Truck	Vehicle type factor values
57-64	Motorcycle	Vehicle type factor values

Emission and Fuel Consumption Rate File
(EMRATE95.PRN)

Characters	Description	Valid Values
1-2	Emission Rate Model	1 = VEHSIME 2 = MOBILE 3 = EMFAC 4 = CALTRANS
3-5	Vehicle Type	1 = Light Duty Gasoline Auto 2 = Light Duty Diesel Auto 3 = Light Duty Truck 4 = Medium Duty Truck 5 = Heavy Duty Truck 6 = Motorcycle
6-8	Facility Type 1 (Source)	1 = Zone Connector 2 = Freeway Merge 3 = Freeway Straight 4 = Freeway Weave 5=OffRamp 6=OnRamp 7 = Arterial Collector
9-11	Facility Type 2 (Destination)	See Facility Type 1 above
12-13	Number of lanes on facility type 1	1 - 5
14-15	Number of lanes on facility type 2	1 - 5
16-17	Pollutant Type or Fuel Consumption Record	1 = Carbon Monoxide 2 = Hydrocarbon 3 = Nitrogen Oxides 4 = Fuel
18-24	Stop Rate on Arterial	rate data
25-3 1	Idle Time Rate on Arterial	rate data
32-38	Speed Rate for O-2.5 mph	rate data
39-45	Speed Rate for 2.5-5.0 mph	rate data
46-52	Speed Rate for 5-O-7.5 mph	rate data
		rate data
207-213	Speed Rate for 62.5-65 mph	rate data

8. Batch files

All of the batch files used to control execution of the framework are listed in this section of Appendix C. Some of these batch files are called from the high level control batch file referred to as FRAME_WKBAT. Many of the lower level batch files will subsequently invoke other lower level files that together enable a controlled sequential execution of the framework.

8.1 Batch File TL.BAT

```
cd d:\tl\plan
set dpath=d:\tl\plan
set SETUP=d:\tl\plan\setup.set
set CONTROL=d:\tl\plan\control.set
set CALIB=d:\tl\plan\calib.set
set help=c:\sysnew\help.set
set system2=c:\sysnew\
set gmode=6466
rem set menufont=c:\sysnew\system96.fnt
set user=NSR
set project=1 111
call startbt.bat
```

8.2 Batch File STARTBT.BAT

```
btrieve /P:2560 /M:24 /C /e
```

8.3 Batch File COMl_RUN.BAT

```
call dumplink.bat
call dumpturnbat
call  clintrfc.bat
call readspd0.bat
REM
call reassgn 1 .bat
call  clintrfc.bat
call readspdl .bat
REM
call reassgn2.bat
call c lintrfc.bat
call readspd2.bat
REM
call reassgn3.bat
call  clintrfc.bat
call readspd3.bat
```

```
REM
call reassgn4.bat
call clintrfc.bat
call readspd4.bat
REM
call reassgn5.bat
call clintrfc.bat
call readspd5.bat
REM
call reassgn6.bat
call  clintrfc.bat
call readspd6.bat
REM
call reassgn7,bat
call clintrfc.bat
call readspd7.bat
REM
call reassgn8.bat
call c lintrfc.bat
call readspd8.bat
```

8.4 Batch File DUMPLINKBAT

```
dataedit dumphrl
dataedit dumphr2
dataedit dumphr3
```

8.5 Batch File DUMPTURN.BAT

```
turndump turn1 .dat turn1 .asc 2
turndump turn2,dat turn2.asc 2
turndump turn3dat turn3.asc 2
```

8.6 Batch File ClINTRFC.BAT

```
cd d:\tl\fsim
copy d:\tl\plan\link?.asc
copy plsbam.inp freql lpl.inp
copy f2sbam.inp freql 1pe.inp
copy plpeeqsb.asc plpeequv.asc
presim < copresb.ctl
copy freqpe.out pesbam.out
copy freqpl.out plsbam.out
copy plnbam.inp freql lpl.inp
copy f2nbam.inp freq 1 lpe.inp
copy plpeeqnb.asc plpeequv.asc
```

```
presim < coprenb.ctl
copy freqpl.out phrbam.out
copy freqpe.out penbam.out
freql 11x <plsbam.out
freql lex < pesbam.out
copy frqplaux.out plsbam.aux
copy frqpeaux.out pesbam.aux
postsim link1 .asc link2.asc link3.asc equvcosb.pst 3
freq 11 lx <plnbam.out
freq 1 lex < penbam.out
copy frqplaux.out plnbam.aux
copy frqpeaux.out penbam.aux
postsim linklasc link2.asc link3.asc equvconb.pst 3
copy hour?.out d:\tl
cd d:\tl\asim
copy d:\tl\plan\link?.asc
copy d:\tl\plan\turn?.asc
call runsys.bat
cd..
copy d:\tl\asim\artspd*.out
copy hour 1 .out+artspd 1 .out
copy hour2.out+artspd2.out
copy hour3.out+artspd3.out
copy hour*.out d:\tl\plan
cd d:\tl\plan
```

3.7 Batch File RUNSYS.BAT

```
tmcgen.exe
netmake.exe
updatvol.exe
batmake.exe
call runt7f.bat
```

8.8 Batch File RUNT7F.BAT

```
echo off
echo TRANSYT-7F Execution Batch File
echo Run Type 4 - Optimize all Time Periods
de1 iospec
ren net 1 - 1 .ios iospec
t7f
de1 netl-l.tin
de1 iospec
de1 null
de1 nu12
```

```
ren net 1-2.ios iospec
t7f
de1 netl-2.tin
de1 iospec
de1 null
de1 nu12
ren net l-3 .ios iospec
t7f
de1 netl-3.tin
de1 iospec
de1 null
de1 nu12
ren netl-4.ios iospec
t7f
de1 netl-4.tin
de1 iospec
de1 null
de1 nu12
ren net l-5 .ios iospec
t7f
de1 netl-5.tin
de1 iospec
de1 null
de1 nu12
ren netl-6.ios iospec
t7f
de1 netl-6.tin
de1 iospec
de1 null
de1 nu12
ren net 1-7.ios iospec
t7f
de1 netl-7.tin
de1 iospec
de1 null
de1 nu12
ren net 1-8.ios iospec
t7f
de1 netl-8.tin
de1 iospec
de1 null
de1 nu12
ren net l -9.ios iospec
t7f
de1 netl-9.tin
```

```
de1 iospec
de1 null
de1 nu12
ren net 1 - 1 O.ios iospec
t7f
de1 netl-lO.tin
de1 iospec
de1 null
de1 nu12
ren net 1 - 11 .ios iospec
t7f
de1 netl-1 l.tin
de1 iospec
de1 null
de1 nu12
ren netl-12.ios iospec
t7f
de1 netl-12.tin
de1 iospec
de1 null
de1 nu12
ren net2- 1 .ios iospec
t7f
de1 net2- 1 .tin
de1 iospec
de1 null
de1 nu12
ren net2-2.ios iospec
t7f
de1 net2-2. tin
de1 iospec
de1 null
de1 nu12
ren net2-3.ios iospec
t7f
de1 net2-3. tin
de1 iospec
de1 null
de1 nu12
ren net2-4.ios iospec
t7f
de1 net2ftin
de1 iospec
de1 null
de1 nu12
```

```
ren net2-5.ios iospec
t7f
de1 net2-5.tin
de1 iospec
de1 null
de1 nu12
ren net2-6.ios iospec
t7f
de1 net2-6.tin
de1 iospec
de1 null
de1 nu12
ren net2-7.ios iospec
t7f
de1 net2-7. tin
de1 iospec
de1 null
de1 nu12
ren net2-8.ios iospec
t7f
de1 net2-8.tin
de1 iospec
de1 null
de1 nu12
ren net2-9.ios iospec
t7f
de1 net2-9.tin
de1 iospec
de1 null
de1 nu12
ren net2- 1 O.ios iospec
t7f
de1 net2- lO.tin
de1 iospec
de1 null
de1 nu12
ren net2- 11 .ios iospec
t7f
de1 net2- 11 .tin
de1 iospec
de1 null
de1 nu12
ren net2-12.ios iospec
t7f
de1 net2- 12. tin
```

```
de1 iospec
de1 null
de1 nu12
ren net3- 1 .ios iospec
t7f
de1 net3- 1 .tin
de1 iospec
de1 null
de1 nu12
ren net3-2.ios iospec
t7f
de1 net3-2.tin
de1 iospec
de1 null
de1 nu12
ren net3-3.ios iospec
t7f
de1 net3-3.tin
de1 iospec
de1 null
de1 nu12
ren net3-4.ios iospec
t7f
de1 net3-4.tin
de1 iospec
de1 null
de1 nu12
ren net3-5.ios iospec
t7f
de1 net3-5.tin
de1 iospec
de1 null
de1 nu12
ren net3-6.ios iospec
t7f
de1 net3-6.tin
de1 iospec
de1 null
de1 nu12
ren net3-7.ios iospec
t7f
de1 net3-7.tin
de1 iospec
de1 null
de1 nu12
```

```
ren net3-8.ios iospec
t7f
de1 net3-8.tin
de1 iospec
de1 null
de1 nu12
ren net3-9.ios iospec
t7f
de1 net3-9.tin
de1 iospec
de1 null
de1 ml2
ren net3- lO.ios iospec
t7f
de1 net3- lO.tin
de1 iospec
de1 null
de1 nu12
ren net3- 11 .ios iospec
t7f
de1 net3- 11 .tin
de1 iospec
de1 null
de1 nu12
ren net3-12.ios iospec
t7f
de1 net3-12.tin
de1 iospec
de1 null
de1 nu12
ren net4- 1 .ios iospec
t7f
de1 net4- 1 .tin
de1 iospec
de1 null
de1 nu12
ren net4-2.ios iospec
t7f
de1 net4-2.tin
de1 iospec
de1 null
de1 nu12
ren net4-3.ios iosflec
t7f
de1 net4-3.tin
```

```
de1 iospec
de1 null
de1 nu12
ren net4-4.ios iospec
t7f
de1 net4-4.tin
de1 iospec
de1 null
de1 nu12
ren net4-5.ios iospec
t7f
de1 net4-5 .tin
de1 iospec
de1 null
de1 nu12
ren net4-6.ios iospec
t7f
de1 net4-6.tin
de1 iospec
de1 null
de1 nu12
ren net4-7.ios iospec
t7f
de1 net4-7. tin
de1 iospec
de1 null
de1 nu12
ren net4-8.ios iospec
t7f
de1 net4-8tin
de1 iospec
de1 null
de1 nu12
ren net4-9.ios iospec
t7f
de1 net4-9.tin
de1 iospec
de1 null
de1 nu12
ren net4- lO.ios iospec
t7f
de1 net4- 10.tin
de1 iospec
de1 null
de1 nu12
```

```
ren net4- 11 .ios iospec
t7f
de1 net4- 11 .tin
de1 iospec
de1 null
de1 nu12
ren net4- 12.ios iospec
t7f
de1 net4-12.tin
de1 iospec
de1 qull
de1 nu12
ren net51 .ios iospec
t7f
de1 net5-l.tin
de1 iospec
de1 null
de1 nu12
ren net52.ios iospec
t7f
de1 net5-2.tin
de1 iospec
de1 null
de1 nu12
ren net53.ios iospec
t7f
de1 net5-3.tin
de1 iospec
de1 null
de1 ml2
ren net5-4.ios iospec
t7f
de1 net5-4.tin
de1 iospec
de1 null
de1 nu12
ren net55.ios iospec
t7f
de1 net5-5.tin
de1 iospec
de1 null
de1 nu12
ren net5-6.ios iospec
t7f
de1 net5-6.tin
```

```
de1 iospec
de1 null
de1 nu12
ren net57.ios iospec
t7f
de1 ned-7.tin
de1 iospec
de1 null
de1 nu12
ren net58.ios iospec
t7f
de1 net5-8tin
de1 iospec
de1 null
de1 nu12
ren net5-9.ios iospec
t7f
de1 net5-9.tin
de1 iospec
de1 null
de1 nu12
ren net5 lO.ios iospec
t7f
de1 net5- lO.tin
de1 iospec
de1 null
de1 nu12
ren net5- 11 .ios iospec
t7f
de1 net5- 11 .tin
de1 iospec
de1 null
de1 nu12
ren net5 12.ios iospec
t7f
de1 net5- 12.tin
de1 iospec
de1 null
de1 ml2
ren net6- 1 .ios iospec
t7f
de1 net6-l.tin
de1 iospec
de1 null
de1 nu12
```

```
ren net6-2.ios iospec
t7f
de1 net6-2.tin
de1 iospec
de1 null
de1 nu12
ren net6-3.ios iospec
t7f
de1 net6-3.tin
de1 iospec
de1 null
de1 nu12
ren net6-4.ios iospec
t7f
de1 net6-4.tin
de1 iospec
de1 null
de1 nu12
ren net6-5.ios iospec
t7f
de1 net6-5.tin
de1 iospec
de1 null
de1 nu12
ren net6-6.ios iospec
t7f
de1 net6-6.tin
de1 iospec
de1 null
de1 nu12
ren netcj-7.ios iospec
t7f
de1 net6-7.tin
de1 iospec
de1 null
de1 nu12
ren net6-8.ios iospec
t7f
de1 net6-8.tin
de1 iospec
de1 null
de1 nu12
ren net6-9.ios iospec
t7f
de1 net6-9. tin
```

```
de1 iospec
de1 null
de1 nu12
ren net6- 10.ios iospec
t7f
de1 net6- 10.tin
de1 iospec
de1 null
de1 nu12
ren net6- 11 .ios iospec
t7f
de1 net6- 11. tin
de1 iospec
de1 null
de1 nu12
ren net6-12.ios iospec
t7f
de1 net6- 12.tin
de1 iospec
de1 null
de1 nu12
ren net7- 1 .ios iospec
t7f
de1 net7- 1. tin
de1 iospec
de1 null
de1 nu12
ren net7-2.ios iospec
t7f
de1 net7-2.tin
de1 iospec
de1 null
de1 nu12
ren net7-3.ios iospec
t7f
de1 net7-3.tin
de1 iospec
de1 null
de1 nu12
ren net7-4.ios iospec
t7f
de1 net7-4.tin
de1 iospec
de1 null
de1 nu12
```

```
ren net7-5.ios iospec
t7f
de1 net7-5.tin
de1 iospec
de1 null
de1 nu12
ren net7-6.ios iospec
t7f
de1 net7-6.tin
de1 iospec
de1 null
de1 nu12
ren net7-7.ios iospec
t7f
de1 net7-7. tin
de1 iospec
de1 null
de1 nu12
ren net7-8.ios iospec
t7f
de1 net7-8.tin
de1 iospec
de1 null
de1 nu12
ren net7-9.ios iospec
t7f
de1 net7-9.tin
de1 iospec
de1 null
de1 nu12
ren net7-10.ios iospec
t7f
de1 net7- 10. tin
de1 iospec
de1 null
de1 nu12
ren net7- 11 .ios iospec
t7f
de1 net7- 11 .tin
de1 iospec
de1 null
de1 nu12
ren net7- 12.ios iospec
t7f
de1 net7- 12.tin
```

```
de1 iospec
de1 null
de1 nu12
ren net8- 1 .ios iospec
t7f
de1 net8-l.tin
de1 iospec
de1 null
de1 nu12
ren net8-2.ios iospec
t7f
de1 net8-2.tin
de1 iospec
de1 null
de1 nu12
ren net8-3.ios iospec
t7f
de1 net8-3.tin
de1 iospec
de1 null
de1 nu12
ren nets-4.ios iospec
t7f
de1 net8-4.tin
de1 iospec
de1 null
de1 nu12
ren net8-5.ios iospec
t7f
de1 net8-5.tin
de1 iospec
de1 null
de1 nu12
ren nets-6.ios iospec
t7f
de1 net8-6.tin
de1 iospec
de1 null
de1 nu12
ren net8-7.ios iospec
t7f
de1 net8-7.tin
de1 iospec
de1 null
de1 nu12
```

```
ren nets-8.ios iospec
t7f
de1 net8-8.tin
de1 iospec
de1 null
de1 nu12
ren net8-9.ios iospec
t7f
de1 net8-9.tin
de1 iospec
de1 null
de1 nu12
ren net8- lO.ios iospec
t7f
de1 net8-lO.tin
de1 iospec
de1 null
de1 nu12
ren net8- 11 .ios iospec
t7f
de1 net8- 11 .tin
de1 iospec
de1 null
de1 nu12
ren net8-12.ios iospec
t7f
de1 net8- 12.tin
de1 iospec
de1 null
de1 nu12
echo End of TRANSIT-7F Runs
speedcmp.exe
```

8.9 Batch File REASSGNl.BAT

```
dataedit maxhrl 1
assign arnhrl
dataedit amtotal 11
dataedit dumphrl 1
turndump turnldat turnlasc 2
dataedit maxhr2 1
assign amhr2
dataedit amtotal
dataedit dumphr2 1
turndump turn2.dat turn2.asc 2
```

```
dataedit maxhr3 1
assign amhr3
dataedit amtotal3 1
dataedit dumphr3 1
turndump turn3.dat turn3.asc 2
turnadj turn1 turn2 turn3 turnl.avg turn2.avg turn3.avg 1
```

8.10 Batch File READSPD1.BAT

```
dataedit readhr 11
perf area1 1 100
perfartll  100
dataedit readhr2 1
perf area21 100
perfart21 100
dataedit readhr3 1
perf area3 1 100
perfart31 100
```

8.11 Batch File COM2_RUN.BAT

```
call dumplinkbat
call dumpturn.bat
call c2intrfc.bat
call readspd0.bat
REM
call reassgn 1 .bat
call c2intrfc.bat
call readspd 1 .bat
REM
call reassgn2.bat
call c2intrfc.bat
call readspd2.bat
REM
call reassgn3.bat
call c2intrfc.bat
call readspd3.bat
REM
call reassgn4.bat
call c2intrfc.bat
call readspd4.bat
REM
call reassgn5.bat
call c2intrfc.bat
call readspd5.bat
```

```
REM
call reassgn6.bat
call c2intrfc.bat
call readspd6.bat
REM
call reassgn7.bat
call c2intrfc.bat
call readspd7.bat
REM
call reassgn8.bat
call c2intrfc.bat
call readspd8.bat
```

8.12 Batch File C2INTRFC.BAT

```
cd d:\tlVsim
copy  d:\tl\planUink?.asc
copy plsbam.inp freql lpl.inp
copy f3bam.inp freql  lpe.inp
copy plpeeqsbasc plpeequv.asc
presim < copresb.ctl
copy freqpe.out pesbarnout
copy freqpkout plsbarnout
copy phrbam.inp freql lpl.inp
copy f3nbam.inp freq 1 lpe.inp
copy plpeeqnbasc plpeequvasc
presim < coprenbctl
copy freqpl.out plnbam.out
copy freqpe.out penbam.out
freq 111x < plsbam.out
freq 1 lex < pesbam.out
copy frqplaux.out plsbam.aux
copy frqpeaux.out pesbam.aux
postsim linkl.asc link2.asc link3.asc equvcosb.pst 3
freq 111x < plnbam.out
freql lex < penbam.out
copy frqplaux.out plnbam.aux
copy frqpeaux.out penbam.aux
postsim linklasc link2.asc link3.asc equvconb.pst 3
copy hour?.out d:\tl
cd d:\tl\asim
copy d:\tl\plan\link?.asc
copy d:\tl\plan\turn?.asc
call runsys.bat
cd..
```

```
copy d:\tl\asim\artspd*.out
copy hour 1 .out+artspd 1 .out
copy hour2.out+artspd2.out
copy hour3 .out+artspd3 .out
copy hour*.out  d:\tl\plan
cd d:\tl\plan
```

8.13 Batch File PL_RUN.BAT

```
call dumplink.bat
call dumpturn.bat
call plintrfc.bat
call readspd0.bat
REM
call reassgn 1 .bat
call plintrfc.bat
call readspdl .bat
REM
call reassgn2.bat
call plintrfc.bat
call readspd2.bat
REM
call reassgn3.bat
call plintrfc.bat
call readspd3 .bat
REM
call reassgn4.bat
call plintrfc.bat
call readspd4.bat
REM
call reassgn5.bat
call plintrfc.bat
call readspd5.bat
REM
call reassgn6.bat
call plintrfc.bat
call readspd6,bat
REM
call reassgn7.bat
call plintrfcbat
call readspd7,bat
REM
call reassgn8.bat
call plintrfc.bat
call readspd8.bat
```

8.14 Batch File PE_D_RUN.BAT

```
call dumplink.bat
call dumpturn.bat
copy d:\tl\fsim\f3sbam.inp pesbam.inp
copy d:\tl\fsim\f3nbam.inp penbam.inp
call peintrfc.bat
call readspd0.bat
rem
call reassgn 1 .bat
call peintrfc.bat
call readspd 1 .bat
rem
call reassgn2.bat
call peintrfc.bat
call readspd2.bat
rem
call reassgn3 .bat
call peintrfc.bat
call readspd3.bat
rem
call reassgn4.bat
call peintrfc.bat
call readspd4.bat
rem
call reassgn5.bat
call peintrfc.bat
call readspd5.bat
rem
call reassgn6.bat
call peintrfc.bat
call readspd6.bat
rem
call reassgn7.bat
call peintrfcbat
call readspd7.bat
rem
call reassgn8.bat
call peintrfcbat
call readspd8.bat
rem
```

8.15 Batch File PE_F_RUN.BAT

```
call dumplink.bat
call dumpturn.bat
copy d:\tl\fsim\f2sbam.inp pesbam.inp
copy d:\tl\fsim\f2nbam.inp penbam.inp
call peintrfc.bat
call readspd0.bat
rem
call reassgn 1 .bat
call peintrfc.bat
call readspd 1 .bat
rem
call reassgn2.bat
call peintrfc.bat
call readspd2.bat
rem
call reassgn3.bat
call peintrfc.bat
call readspd3.bat
rem
call reassgn4.bat
call peintrfc.bat
call readspd4.bat
rem
call reassgn5.bat
call peintrfc.bat
call readspd5.bat
rem
call reassgn6.bat
call peintrfc.bat
call readspd6.bat
rem
call reassgn7.bat
call peintic.bat
call readspd7.bat
rem
call reassgn8.bat
call peintrfc.bat
call readspd8.bat
rem
```

8.16 Batch File PLINTRFC.BAT

```
cd d:\tl\fsim
copy d:\tl\plan\link.asc
copy plsbam.inp freql lpl.inp
presim < p1presb.ctl
copy freqpl.out plsbam.out
copy phrbam.inp freql lpl.inp
presim < plprenb.ctl
copy freqpl.out plnbam.out
freql 11x < plsbam.out
copy freql lpl.out sbplots.out
copy frqplaux.out plsbam.aux
postsim link1 .asc link2.asc link3.asc equvplsb.pst 1
freq 111x < phrbam.out
copy freql lpl.out nbplots.out
copy frqplaux.out plnbam.aux
postsim link1.asc link2.asc link3.asc equvplnb.pst 1
copy hour?.out d:\tl
cd d:\tl\asim
copy d:\tl\plan\link?.asc
copy d:\tl\plan\turn?.asc
call runsys.bat
cd..
copy d:\tl\asim\artspd*.out
copy hour 1 .out+artspd 1 .out
copy hour2.out+artspd2.out
copy hour3.out+artspd3.out
copy hour*.out d:\tl\plan
cd d:\tl\plan
```

8.17 Batch File PEINTRFC.BAT

```
cd d:\tl\fsim
copy d:\tl\planning\link?.asc
copy pesbam.inp freql lpe.inp
presim < pepresb.ctl
copy freqpe.out pesbam.out
copy penbam.inp freq 1 lpe.inp
presim < peprenbctl
copy freqpe.out penbam.out
freql lex < pesbam.out
copy frqpeaux.out pesbam.aux
postsim linkl.asc link2.asc link3.asc equvpesb.pst 2
freql lex < penbam.out
copy fiqpeaux.out penbam.aux
postsim link1 .asc link2.asc link3.asc equvpenb.pst 2
copy hour?.out d:\tl
cd d:\tl\asim
```

```
copy d:\tI\plan\link?.asc
copy d:\tl\plan\turn?.asc
call runsys.bat
cd..
copy d:\tl\asim\spd*.out
copy hour1 .out+artspdl .out
copy hour2.out+artspd2.out
copy hour3.out+artspd3.out
copy hour*.out d:\tl\plan
cd d:\tl\plan
```

8.18 Batch File IMPACTS.BAT

```
cd\tl\mp
set SETUP=d:\tl\imp\impsetup.set
set CONTROL==d:\tl\imp\impcntrl.set
set CALB=d:\tl\imp\impcalib.set
call startbt
```

8.19 Batch File EMISRUN.BAT

```
REM To create I880LINK.ems file
dataedit create-ems-file
dataedit zone-speed
REM    Create ASCII link files by hour
dataedit emisdump_hrl
dataedit emisdump_hr2
dataedit emisdump_hr3
REM   Emissions and fuel consumption estimation
emis < doemis.ctl
REM    Load results into I880LINK.EMS
dataedit readmobile_hrl
dataedit readmobile_hr2
dataedit readmobile_hr3
dataedit reademfac_hr 1
dataedit reademfac_hr2
dataedit reademfac_hr3
dataedit readfuel_hrl
dataedit readfuel_hr2
dataedit readfuel_hr3
volsum <mob-co.ctl
volsum < mob-hc.ctl
volsum < mob-nox.ctl
volsum < vol_fuel.ctl
perf mobile-co-tot < spdin.ctl
perf mobile-hc-tot c spdinctl
perf mobile-nox-tot c  spdin.ctl
perf fuel-tot < spdinctl
perf mobile-co-totart <  spdinctl
perf mobile-hctotart < spdin.ctl
perf mobile_nox_totart < spdin.ctl
perf fuel_totart < spdinctl
volsum < emf_co.ctl
volsum < emf_hc.ctl
volsum < emf_nox.ctl
perf emfac_co_tot < spdinctl
perf emfac_hc_tot < spdinctl
perf emfac_nox_tot < spdin.ctl
perf emfac_co_totart < spdinctl
perf emfac_hc_totart< spdinctl
perf emfac_nox_totart < spdin.ctl
```

8.20 Batch File SAFERUN.BAT

```
dataedit zone-speed
dataedit emisdump_hr1
dataedit emisdump_hr2
dataedit emisdump_hr3
safety < safe-pdo.ctl
dataedit readsafety_1
volsum < vol_pdo.ctl
perf pdo_fwytot c spdin.ctl
perf pdo_arttot < spdin.ctl
copy safety 1 .prn pdo_acc 1 .prn
copy safety2.prn pdo_acc2.prn
copy safety3.prn pdo_acc3.prn
safety < safe-inj.ctl
dataedit readsafety_2
volsum < vol_inj.ctl
perf inj_fwytot < spdinctl
perf inj_arttot < spdin.ctl
copy safety 1 .prn inj_acc 1 .prn
copy safety2.prn inj_acc2.prn
copy safety3.prn inj_acc3.prn
safety < safe-tot.ctl
dataedit readsafety-
volsum < vol_acc.ctl
perf acc_fwytot < spdin.ctl
perf acc_arttot  < spdin.ctl
copy safety 1 .prn tot_acc 1 .pm
copy safety2.prn tot_acc2.prn
copy safety3.prn tot_acc3.prn
```

8.21 Batch File STATEXP.BAT

```
dataedit dump_statistica_hrl
dataedit dump_statistica_hr2
dataedit dump_statistica_hr3
```

8.22 Batch File READSPDO.BAT

```
dataedit readhrl
dataedit readhr2
dataedit readhr3
```

8.23 Batch File READSPD2.BAT

```
dataedit readhr12
dataedit readhr22
dataedit readhr32
```

8.24 Batch File READSPD3.BAT

```
dataedit readhr13
dataedit readhr23
dataedit readhr33
```

8.25 Batch File READSPD$.BAT

```
dataedit readhr14
dataedit readhr24
dataedit readhr34
```

8.26 Batch File READSPD5.BAT

```
dataedit readhr 15
dataedit readhr25
dataedit readhr35
```

8.27 Batch File READSPD6.BAT

```
dataedit readhr 16
dataedit readhr26
dataedit readhr36
```

8.28 Batch File READSPD7.BAT

```
dataedit readhr17
dataedit readhr27
dataedit readhr37
```

8.29 Batch File READSPDS.BAT

```
dataedit readhrl8
dataedit readbr28
dataedit readhr38
```

8.30 Batch File REASSGN2.BAT

```
dataedit maxhr 12
assign amhrl
dataedit amtotall2
dataedit dumphr12
turndump turn 1 .dat turn 1 .asc 2
dataedit maxhr22
assign amhr2
dataedit amtotal22
dataedit dumpbr22
turndump turn2.dat turn2.asc 2
dataedit maxhr32
assign anhr3
dataedit amtotal32
dataedit dumphr32
turndump turn3.dat turn3.asc 2
turnadj turn1 turn2 turn3 turn1 .avg turn2.avg turn3.avg 2-
```

8.31 Batch File REASSGN3,BAT

```
dataedit maxhr13
assign amhrl
dataedit amtotal3
dataedit dumphr13
turndump turnl.dat turnl.asc 2
dataedit maxhr23
assign amhr2
dataedit amtotal23
dataedit dumphr23
turndump turn2.dat turn2.asc 2
dataedit maxhr33
assign amhr3
dataedit amtotal33
dataedit dumphr33
turndump turn3.dat turn3.asc 2
turnadj turn1 turn2 turn3 turnl.avg turn2.avg turn3.avg 3
```

8.32 Batch File REASSGN4.BAT

```
dataedit maxhr14
assign amhrl
dataedit amtotal 14
dataedit dumphrl4
turndump turn1 .dat turn 1 .asc 2
dataedit maxhr24
assign amh2
dataedit amtotal24
dataedit dumphr24
turndump turn2.dat turn2.as.c 2
dataedit maxhr34
assign amhr3
dataedit amtotal34
dataedit dumphr34
turndump turn3.dat turn3.asc 2
turnadj turn 1 turn2 turn3 turn1 .avg turn2.avg turn3.avg 4
```

8.33 Batch File REASSGN5.BAT

```
dataedit maxhr15
assign amhrl
dataedit amtotal 15
dataedit dumphr15
turndump turn 1 .dat turn 1 .asc 2
dataedit maxhr25
assign amhr2
dataedit amtotal25
dataedit dumphr25
turndump turn2.dat turn2.asc 2
dataedit maxhr35
assign amhr3
dataedit amtotal35
dataedit dumphr35
turndump turn3.dat turn3.asc 2
turnadj turn1 turn2 turn3 turnl.avg turn2.avg turn3.avg 5
```

8.34 Batch File REASSGN6.BAT

```
dataedit maxhr16
```

```
assign amhrl
dataedit amtotal
dataedit dumphr16
turndump turn 1 .dat turn 1 .asc 2
dataedit maxhr26
assign amhr2
dataedit amtotal26
dataedit dumphr26
turndump turn2.dat turn2.asc 2
dataedit rnaxbr36
assign amhr3
dataedit amtotal36
dataedit dumphr36
turndump turn3.dat turn3.asc 2
turnadj turn1 turn2 turn3 turn1 .avg turn2.avg turn3.avg 6
```

8.35 Batch File REASSGN7.BAT

```
dataedit maxhrl7
assign amhrl
dataedit amtotal 17
dataedit dumphr17
turndump turn 1 .dat turn 1 .asc 2
dataedit maxhr27
assign amhr2
dataedit amtotal27
dataedit dumphr27
turndump turn2.dat turn2.asc 2
dataedit maxhr37
assign amh3
dataedit amtotal37
dataedit dumphr-37
turndump turn3.dat turn3.asc 2
turnadj turnl turn2 turn3 turnl.avg turn2.avg turn3.avg 7
```

8.36 Batch File REASSGN8.BAT

```
dataedit maxhrl8
assign amhrl
dataedit amtotal 18
dataedit dumphrl8
turndump turnl.dat turn1 .asc 2
dataedit maxhr28
assign amh2
dataedit amtotal28
dataedit dumphr28
turndump turn2.dat turn2.asc 2
dataedit maxhr38
assign amhr3
dataedit amtotal38
dataedit dumpbr38
turndump turn3.dat turn3.asc 2
turnadj turn1 turn2 turn3 turnl.avg turn2.avg turn3.avg 8
```