

Program Documentation

FSUTMS Standard Reports

Prepared by:

Tao Wang, Graduate Research Assistant
Albert Gan, Ph.D., Associate Professor
Meng Ma, Visiting Scholar

Lehman Center for Transportation Research
Department of Civil and Environmental Engineering
Florida International University
10555 West Flagler Street, EC 3680
Miami, Florida 33174
Phone: (305) 348-3116
Fax: (305) 348-2802
E-mail: gana@fiu.edu

for

Systems Planning Office
State of Florida Department of Transportation
605 Suwannee Street, MS 30
Tallahassee, FL 32399-0450

April 2009
Revised June 2010

ACKNOWLEDGEMENTS

The authors are grateful to the following individuals for their review of various versions of the FSUTMS Standard Reports program:

- Mr. Vidya Mysore, Systems Planning Office, Florida Department of Transportation (FDOT).
- Ms. Lina Kulikowski, Broward Metropolitan Planning Organization Division.
- Mr. Yongqiang Wu, Gannett Fleming, Inc.
- Mr. Robert Schiffer, Cambridge Systematics, Inc.

Special thanks to Mr. Atiosis Blanco of the Lehman Center for Transportation Research (LCTR) at Florida International University for providing graphic support.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS.....	ii
TABLE OF CONTENTS.....	iii
1. INTRODUCTION	1
2. SYSTEM ARCHITECTURE AND REQUIREMENTS.....	2
2.1. System Architecture	2
2.2. System Installation	3
2.3. System Requirements	5
3. FUNCTIONALITIES AND USER INTERFACE	6
3.1. Interface Operations	6
3.1.1. Opening a Window.....	6
3.1.2. Floating a Window	8
3.1.3. Resizing a Window	9
3.1.4. Moving and Docking a Window	9
3.1.5. Closing a Window	13
3.1.6. Auto-Hiding a Window	13
3.2. Project Management.....	14
3.2.1. Creating a Project.....	15
3.2.2. Changing Project Settings	18
3.2.3. Saving a Project.....	20
3.2.4. Opening a Project.....	21
3.2.5. Closing a Project	23
3.3. Report Operations.....	23
3.4. GIS Map Settings	24
3.4.1. Selecting Additional Map Areas	24
3.4.2. Selecting Map Quality.....	28
3.4.3. Generating MXD and Geodatabase Files.....	28
4. REPORT DETAILS.....	30
4.1. Socioeconomic Ratio.....	30
4.2. Trip Rates	31
4.3. Subarea Summary.....	33
4.4. Trip Productions/Attractions	34
4.5. Total Production by Purpose	35
4.6. Total Attractions by Purpose	36
4.7. Bidirectional Number of Lanes	37
4.8. Area Type	39
4.9. Facility Type.....	40
4.10 Subarea	41
4.11. Traffic Count Locations	42
4.12. Screenline Locations	43

4.13. Transit Service.....	44
4.14. Speed Curve Lookup	46
4.15. Speed Curve Graph.....	47
4.16. Input Vs. Output Attractions	48
4.17. Intrazonal Vs. Total Trips.....	50
4.18. Average Trip Length	51
4.19. Trip Length Frequency Curve	53
4.20. Trip Length Frequency	55
4.21. Congested Travel Time to One Zone	57
4.22. Flow Chart.....	58
4.23. Coefficients.....	59
4.24. Bias Constants	60
4.25. Split by Purpose.....	61
4.26. Split by Zone	62
4.27. Free Flow Speed By Area Type and Facility Type	64
4.28. Congested Speed by Area Type and Facility Type	65
4.29. Volume-to-Count Ratio by Area Type and Facility Type.....	66
4.30. Volume to Capacity Ratio by Screenline	67
4.31. Bidirectional Capacity	68
4.32. Bidirectional Volume	69
4.33. Bidirectional Volume-to-Capacity Ratio.....	70
4.34. Volume-to-Capacity by Area Type and Facility Type	71
4.35. VMT by Area Type and Facility Type	72
4.36. VHT by Area Type and Facility Type.....	73
4.37. RMSE	74
4.38. Estimated Patronage	75
4.39. Transfer by Mode	77
4.40. Station Load.....	78
4.41. Riders by Mode and Route	79
4.42. Peak Period Vehicle Requirement.....	80
5. SYSTEM ENHANCEMENTS	81

1. INTRODUCTION

The purpose of this document is to describe a prototype desktop software system developed to automatically generate a set of 42 standard reports for the Cube Voyager version of FSUTMS. Developed by Gannett Fleming, Inc.¹, the standard reports were designed to display select model input and output data from each of the five major FSUTMS modeling steps: generation, distribution, modal split, highway assignment, and transit assignment. The standard reports include a combination of tables, charts, and GIS maps.

The current version of the prototype system includes the following major features and capabilities:

- It provides quick access to specific reports through a report list menu.
- It allows the simultaneous opening of multiple reports for multiple project scenarios to facilitate easy side-by-side comparisons.
- It provides a comprehensive set of report functions that includes export, print, search, zoom, etc.
- It provides a standalone GIS interface for the user to freely select and define multiple map areas to then and automatically generate map reports for the selected areas.
- It allows the export of GIS layers in all standard map reports to be exported with an accompanied MXD file that defines all the map settings.

Section 2 of this document describes the system architecture, system installation, and system requirements. Section 3 describes the user interface for project management, report operations, and map settings. Section 4 documents the input and output files used to generate each report and the process used to extract and calculate the data going into each report. Finally, Section 5 provides two specific recommendations to develop the prototype system into a production system that includes more standard reports and provides capabilities to customize input from various FSUTMS models.

¹. *Development of Standard FSUTMS Reports*, Submitted to Florida Department of Transportation, Central Office, Submitted by Gannett Fleming, Inc., April 2006. Available at:
http://www.fsutmsonline.net/images/uploads/mtf-files/FSUTMS_Templates-Draft-Final.pdf

2. SYSTEM ARCHITECTURE AND REQUIREMENTS

This section describes the structure of the system. It includes the individual system components, the data flow, and the associated input and output files. Also described are the system installation and requirements.

2.1. System Architecture

The system was developed as a standalone desktop application using the C# language in Microsoft Visual Studio 2005. Figure 2-1 shows the structure of system. The system consists of the following five components: project management, data conversion, data retrieval, GIS, and user interface.

As shown in the figure, the Project Management component provides functions to manage projects through the use of project files. Details on this component are provided in subsection 3.2.

When a specific report is selected by the user for data display, the system will first check to determine if the data needed for the report have been retrieved before. If so, the system will further check to determine if the data have become outdated because the input data have been changed. This is done by comparing the time stamps of when the data were last retrieved with those of the current input data files. If no data have been retrieved before or if the retrieved data have become outdated, the system will utilize the following two components, as needed, to generate the required data: the Data Conversion component and the GIS component (only used if the selected report is a GIS report).

When called, the Data Conversion component will convert some required scenario input and output data files into some temporary project data files. This component may also call upon the Cube Voyager engine to help convert some of the files that cannot be read directly. This includes the Cube native matrix (.mat) and network (.net) files. The steps to convert these files are as follows:

1. The Data Conversion component creates Cube Voyager script files.
2. The Data Conversion component starts the Cube Voyager engine.
3. The Cube Voyager engine executes the script files.
4. The Cube Voyager engine reads data from the .mat or .net files.
5. The Cube Voyager engine generates the .dbf version of the files.

If the selected report involves GIS maps, the system will call the GIS component. Developed using ArcGIS Engine, the component serves two purposes: (1) generate JPEG image files for input to reports involving maps, and (2) generate MXD (.mxd) ArcMap file and Geodatabase (.mdb) files for all available map layers for users who want to use the GIS layers with an external GIS system like ArcMap. In both cases, the GIS component reads the ArcGIS shape files (.shp, .shx, and .dbf files) and then applies functions from ArcGIS Engine to automatically generate the JPEG image files, the MXD file (storing layer properties), and the Geodatabase file (storing layer data).

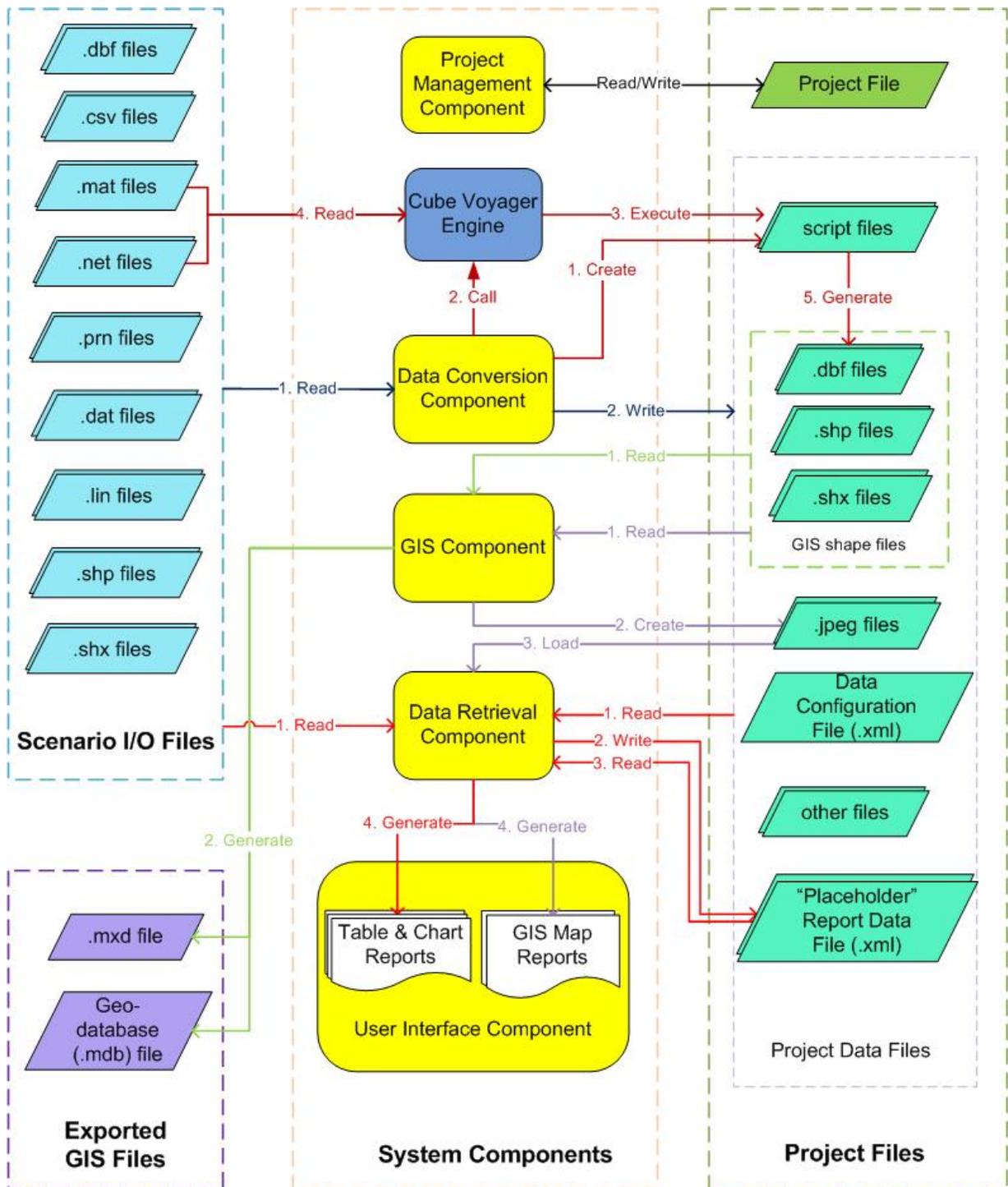


Figure 2-1: System Architecture

After the Data Conversion component and the GIS component (if applicable) complete their tasks, the system will call the Data Retrieval component to perform the following tasks: (1) to retrieve all data needed for the selected report from the applicable scenario I/O and project data files according to the settings in a Data Configuration File (.xml), (2) write the retrieved data to a Report Data File (.xml file) named by the report name that serves as a “placeholder” for the data, (3) read the

data for the selected report from the placeholder XML file, and (4) call the User Interface component to insert the data into the report template and display the report. Details for the Project Management and User Interface components are given in Section 3 and the detailed steps used to process the data going into each report are given in Section 4.

2.2. System Installation

The system was packaged using InstallShield 2009. To install the program, simply double-click on the installation executable file. After a few seconds, the setup program will display a welcome screen as shown in Figure 2-2. Follow the instructions on the screen to complete the installation. If an older version of the program is detected on the user's computer, the setup program will prompt the screen shown in Figure 2-3. It allows the user to modify the current installation. It is highly recommended that the Remove option be selected to completely remove the existing install and then re-run the setup program.

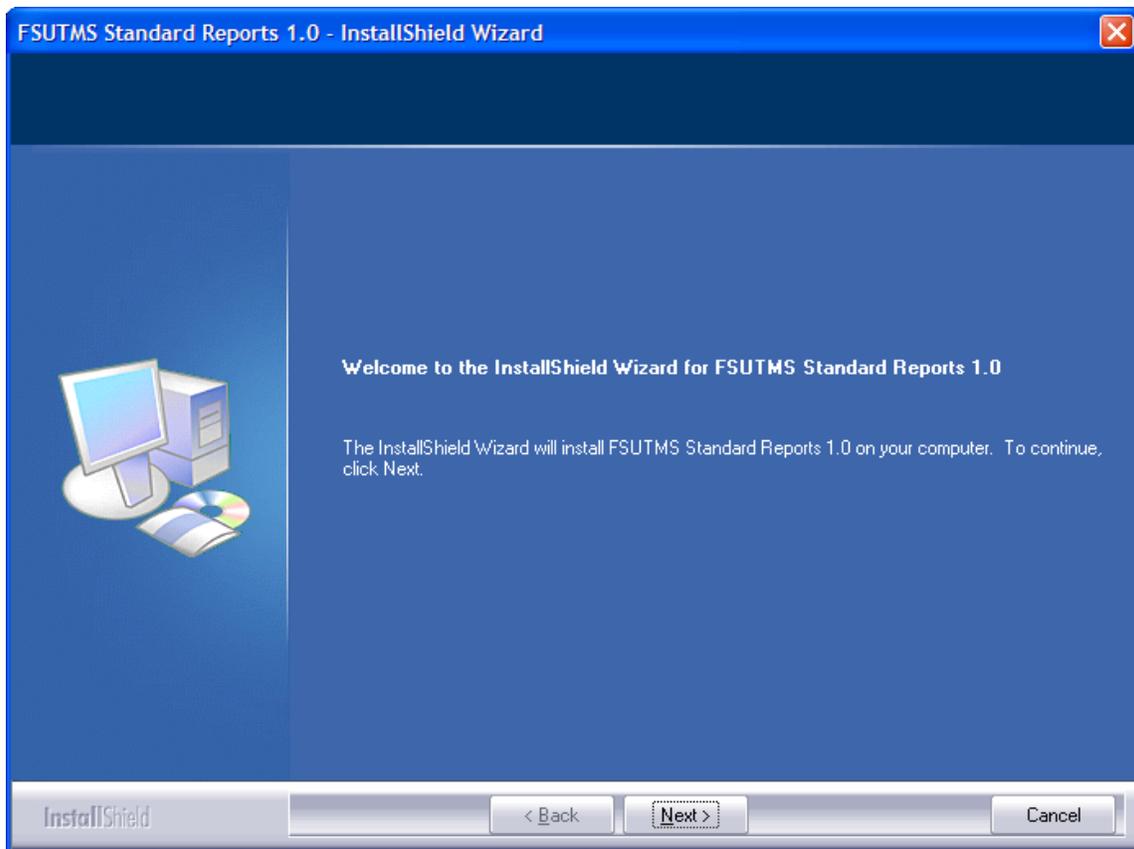


Figure 2-2: Welcome Dialog of Setup Program

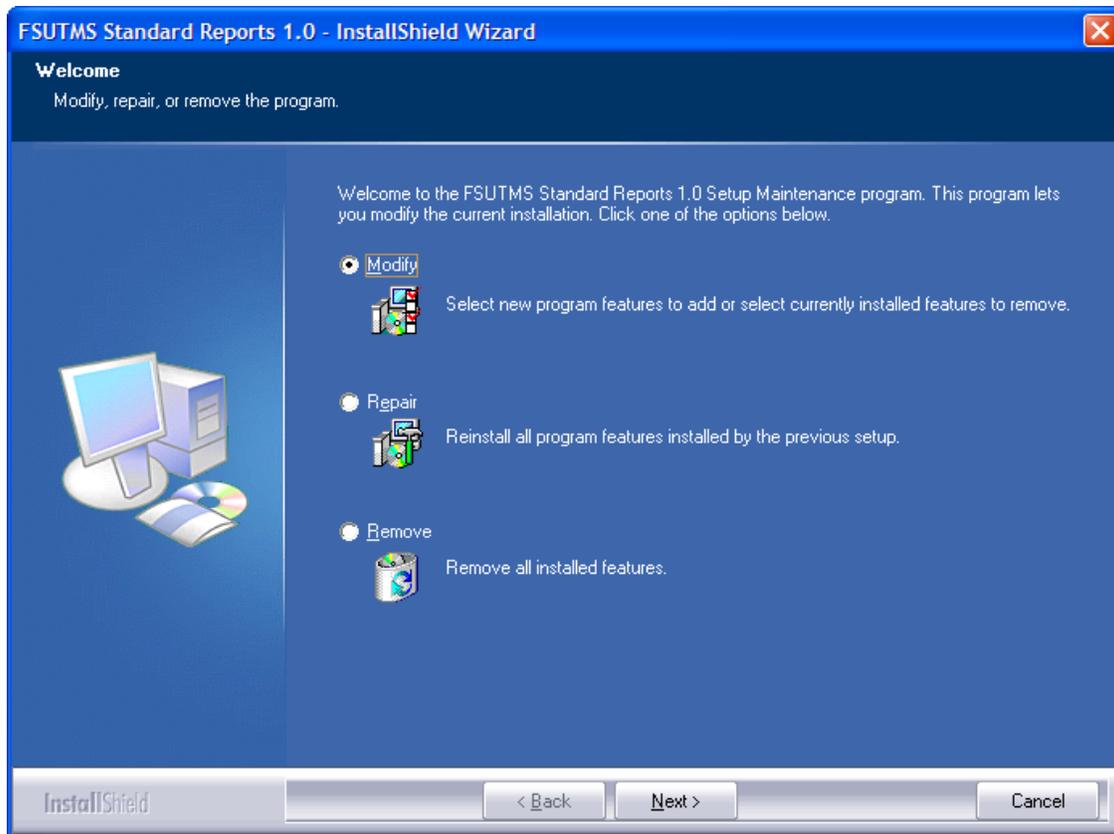


Figure 2-3: Welcome Dialog for Modifying Existing Install

2.3. System Requirements

To run the system successfully, the computer must be installed with the following additional software:

1. Microsoft .NET Framework 2.0 (or above) Runtime.
2. Cube 5.0 (or above).
3. ArcGIS Engine version 9.2 (or above) Runtime or ArcGIS Desktop version 9.2 (or above), licenses authorized.
4. Service Pack 5 for ArcGIS Engine Runtime, or Desktop version 9.2 and Service Pack 1 for version 9.3.

Microsoft .NET Framework 2.0 Runtime is included in the install file. During the install, the .NET Framework 2.0 Runtime will be automatically installed if one (version 2.0 or above) is not detected. The last two items are optional if the user does not want any of the GIS reports. The install package does not come with the install program for ArcGIS Engine Runtime.

3. FUNCTIONALITIES AND USER INTERFACE

This section focuses on the functionalities of the system and the user interface designed to execute these functionalities. The functionalities of the system can be categorized into the following four groups: interface operations, project management, report operations, and GIS map settings.

3.1. Interface Operations

Figure 3-1 shows the main screen of the system. The layout of the screen consists of three major windows: Report List, Project List, and Report Tab. The Report List window displays a list of available reports that are grouped by the standard modeling steps. The Project List window contains all the opened projects. Double-clicking a report in the Report List window will open and display the selected report in the Report Tab window. Each of the three windows can be customized using the following window operations: open, float, dock, move, close, close, auto-hide, and resize. Each of these operations is described in detail below.

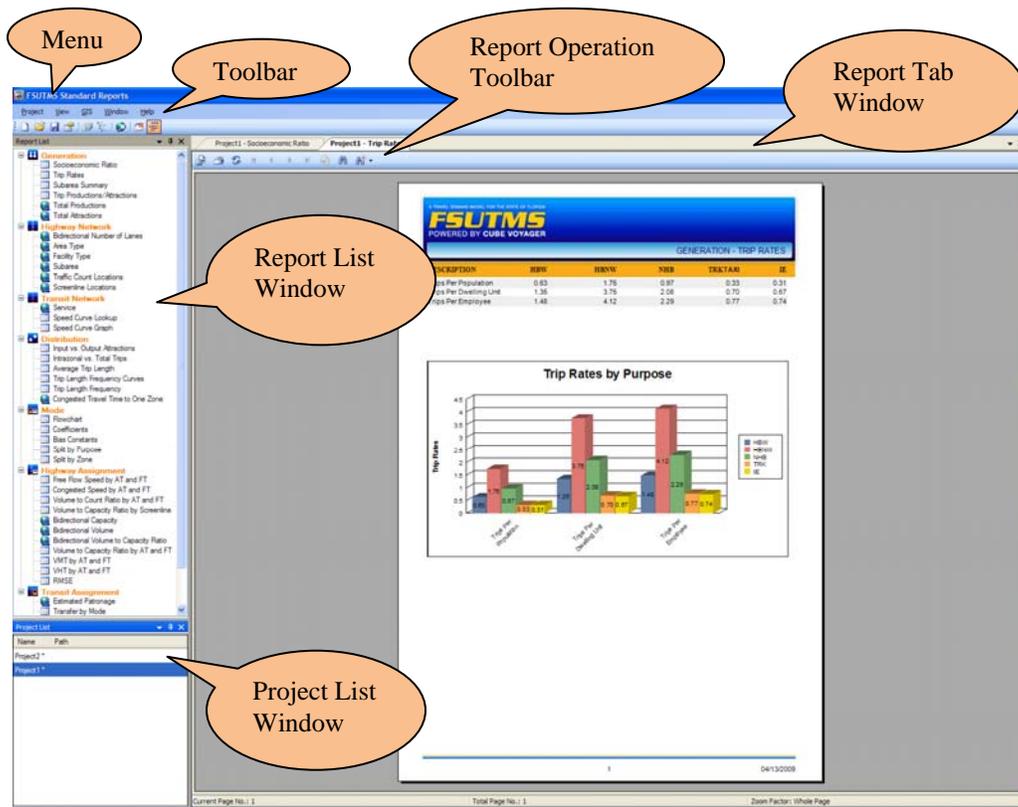


Figure 3-1: Layout of FSUTMS Standard Reports Main Window

3.1.1. Opening a Window

To open the Report List window:

- Select View|Report List, or
- Click the Report List icon button on the toolbar (see Figure 3-2).

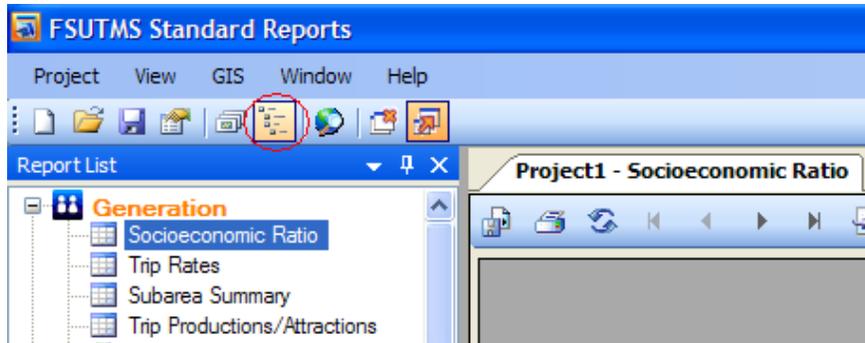


Figure 3-2: "Report List" Icon Button on Toolbar

Similarly, to open the Project List window:

- Select View|Project List, or
- Click the Project List icon button on the toolbar (see Figure 3-3).

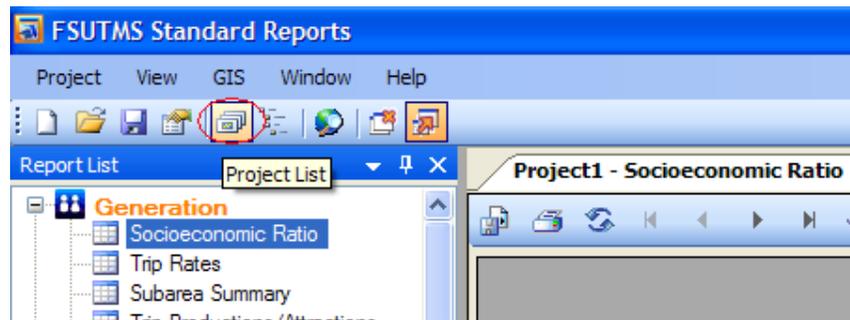


Figure 3-3: "Project List" Icon Button on Toolbar

To open a report, simply double-click the report name in the Report List window. If multiple projects are selected, double-clicking a report will generate multiple reports, one from each project (for details on projects, see subsection 3.2). By default, each report will open in a new tab. This default can be changed to always open a report in the same tab by:

- Selecting Window|Open Report in New Tab to deselect it, or
- Clicking the "Open Report in New Tab" icon button on the toolbar (see Figure 3-4).

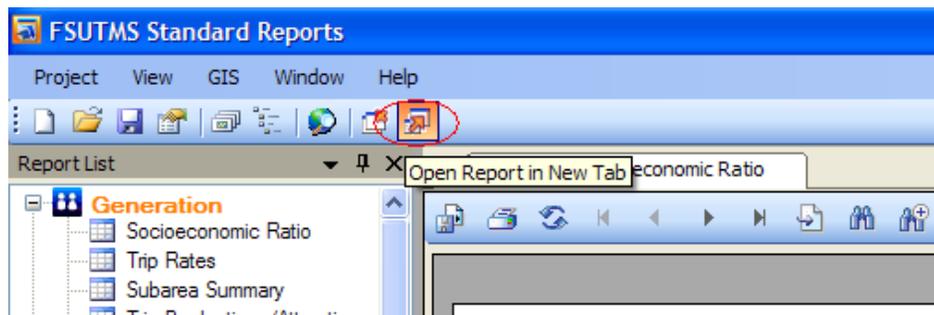


Figure 3-4: "Open Report in New Tab" Icon Button on Toolbar

To navigate among the report tabs:

- Click the tab showing the report name, or
- Select from the Window list as shown in Figure 3-5, or
- Click the down arrow (triangle icon) shown in Figure 3-6 and then select from the tab list.

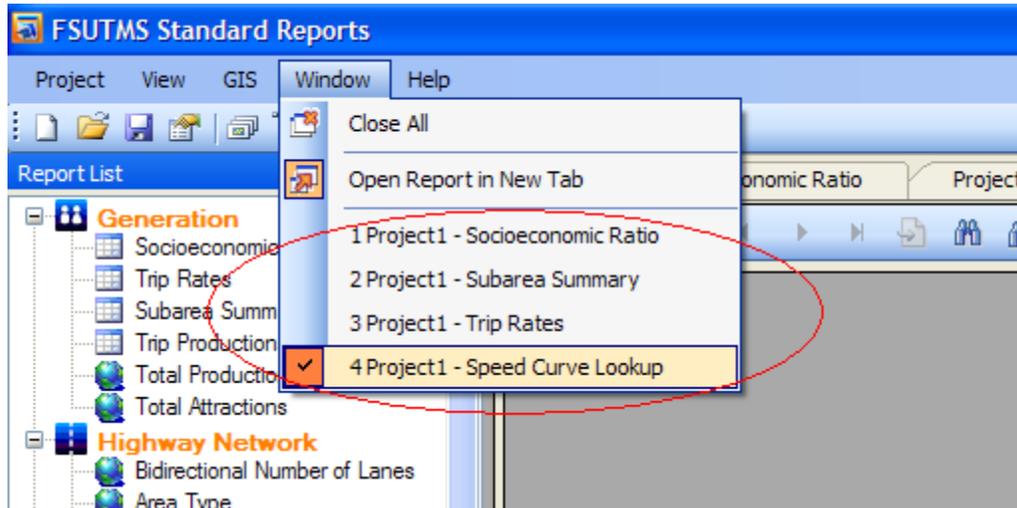


Figure 3-5: Report Window List under Window Menu

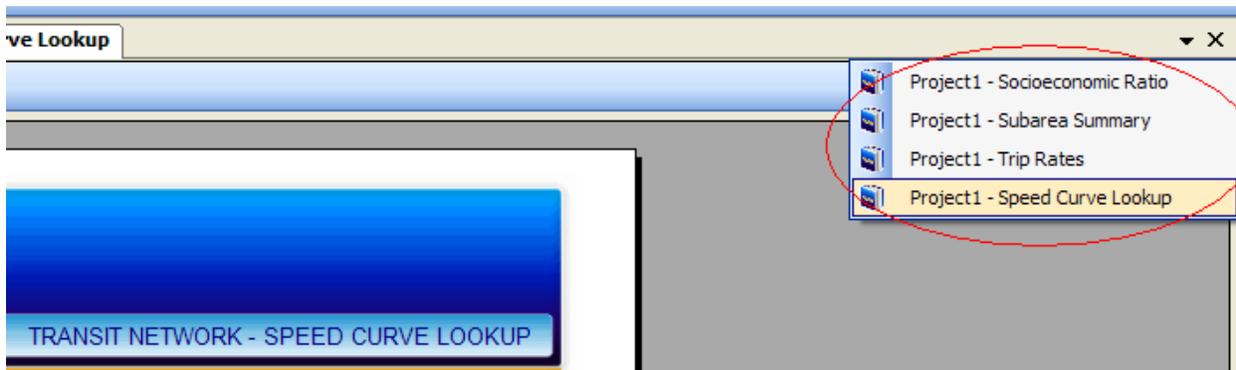


Figure 3-6: Report Window List under Triangle Icon Button

3.1.2. Floating a Window

The Project List and Report List windows can be made to float by double-clicking the title bar of each window. Similarly, a report tab can be floated by double-clicking the tab. A floating window or tab can be moved and resized outside the extent of the main window. Figure 3-7 shows an example in which all the windows are floating. A window can also be made to float by dragging and moving it. This capability is described in greater detail in subsection 3.1.4.

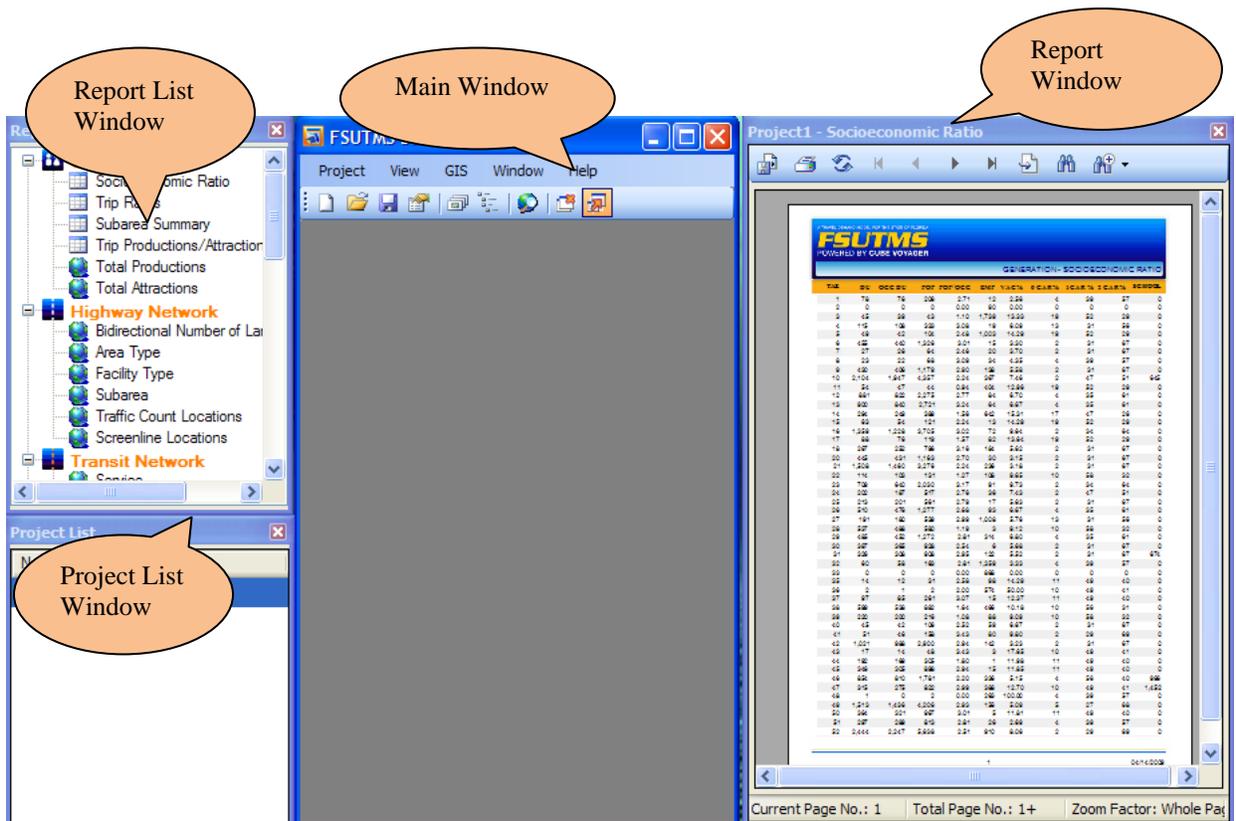


Figure 3-7: Sample Screen with All Windows Floating

3.1.3. Resizing a Window

A floating window can be easily resized by dragging any corner of the window to the desired dimension.

3.1.4. Moving and Docking a Window

Any window (Project List, Report List, or Report) can be moved by dragging the title bar. A window being moved can then be docked to another window and become one of the sub-windows within a larger window. An exception is the Report window, which cannot be docked in the Project List window or the Report List window.

A floating window can be docked to the main window by simply by double-clicking its title bar. For example, double-clicking the Report window to the right in Figure 3-7 will change the layout to Figure 3-8.

Another way of docking a window is by simply dragging and moving the window and then placing it on top of another window. This allows multiple windows to be displayed simultaneously, thereby facilitating easy side-by-side report comparisons. Figure 3-9 shows an example that compares two reports horizontally. Similarly, Figure 3-10 shows one that compares two reports vertically.

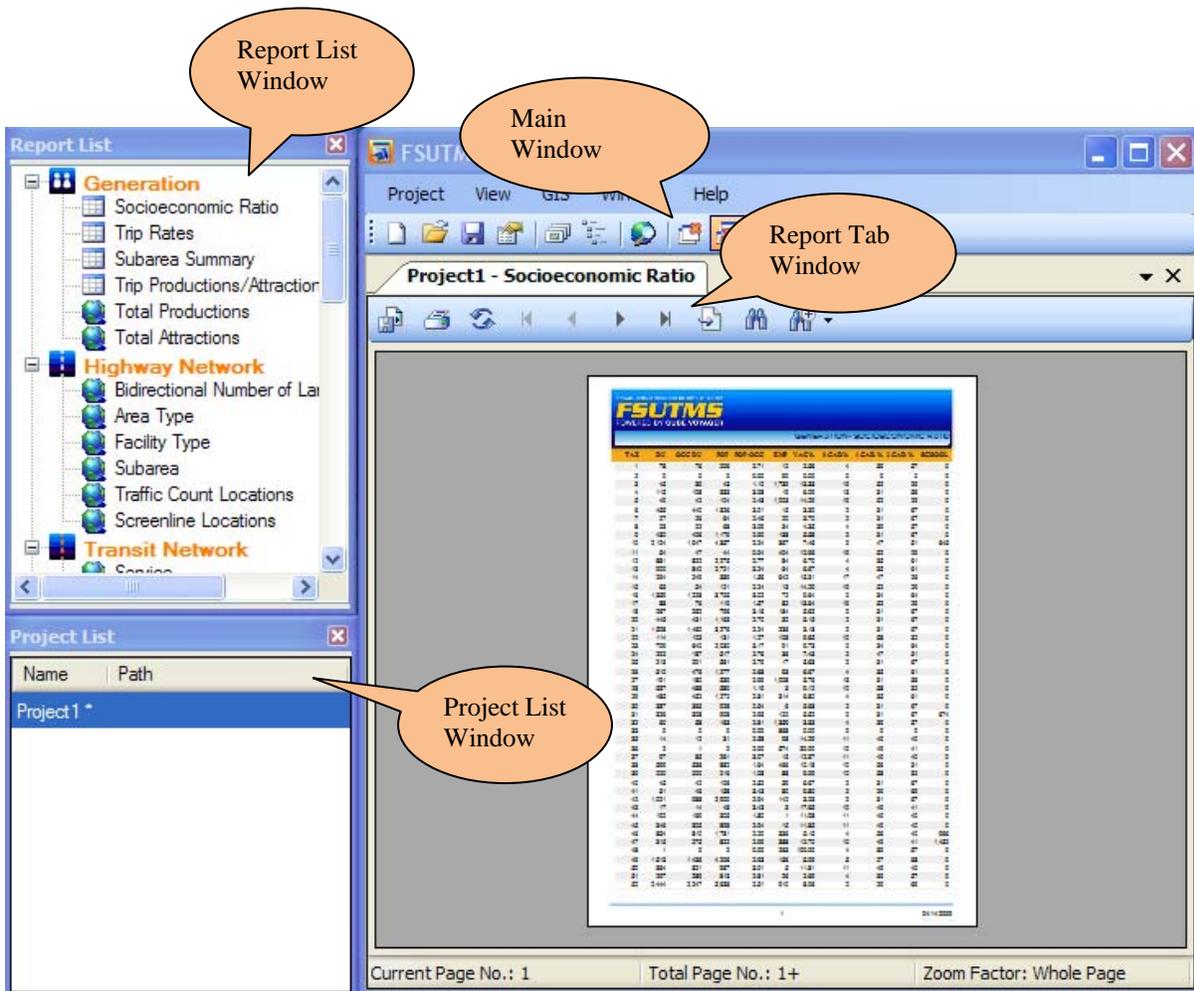


Figure 3-8: Report Window Docked after Double-Clicking

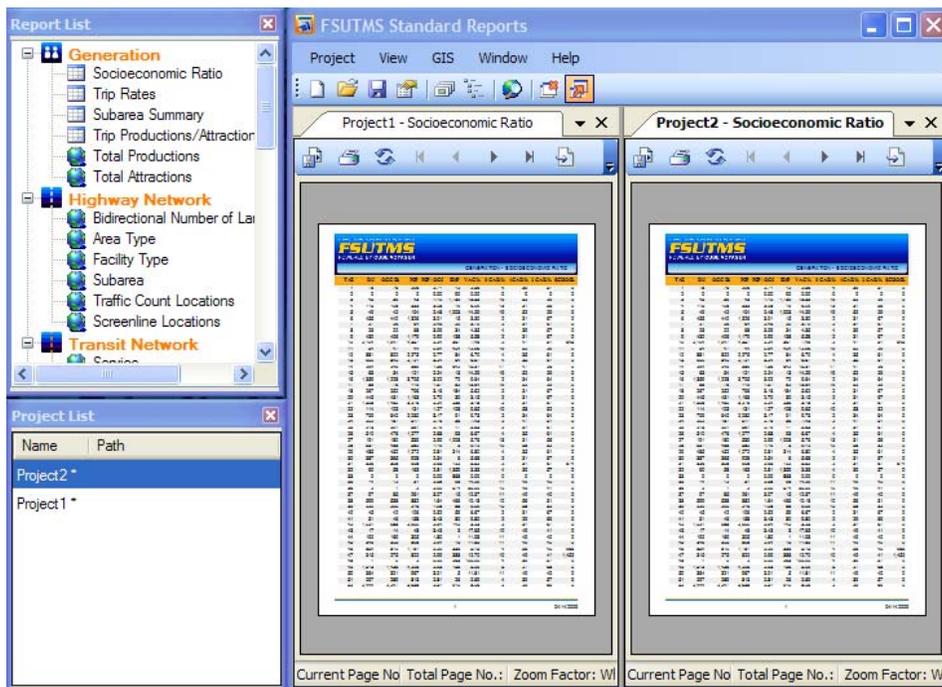


Figure 3-9: Horizontal Report Comparison

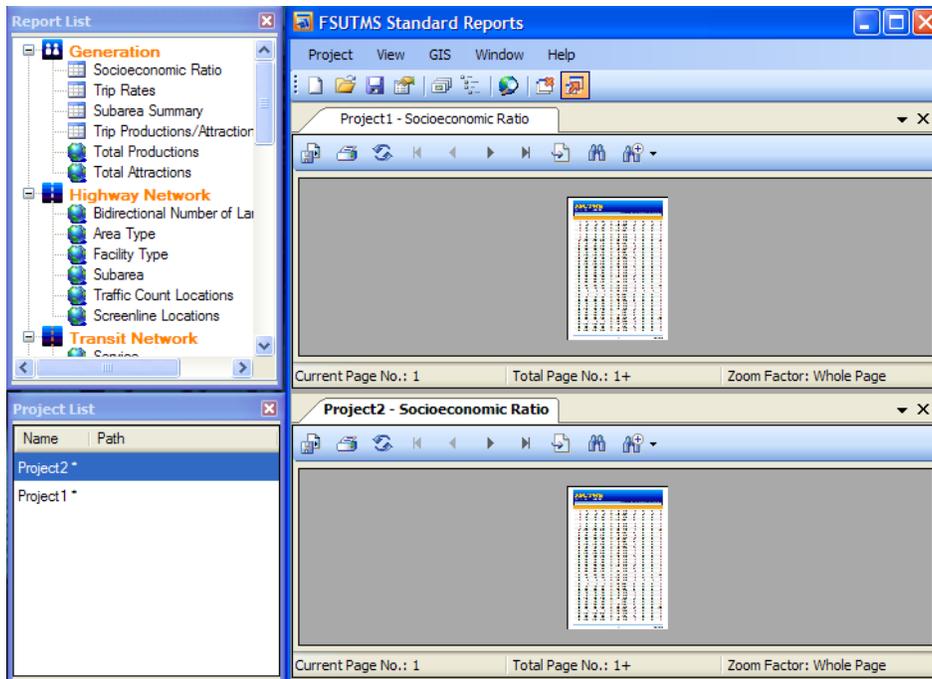


Figure 3-10: Vertical Report Comparison

When a window is dragged into another window, a shadow appears to allow for tracing of the window position. At the same time, four directional icons, each with an arrow head, will appear in the target window. For example, if the Project List window in Figure 3-8 is dragged into the main window, both the shadow and the directional icons will appear as shown in Figure 3-11.

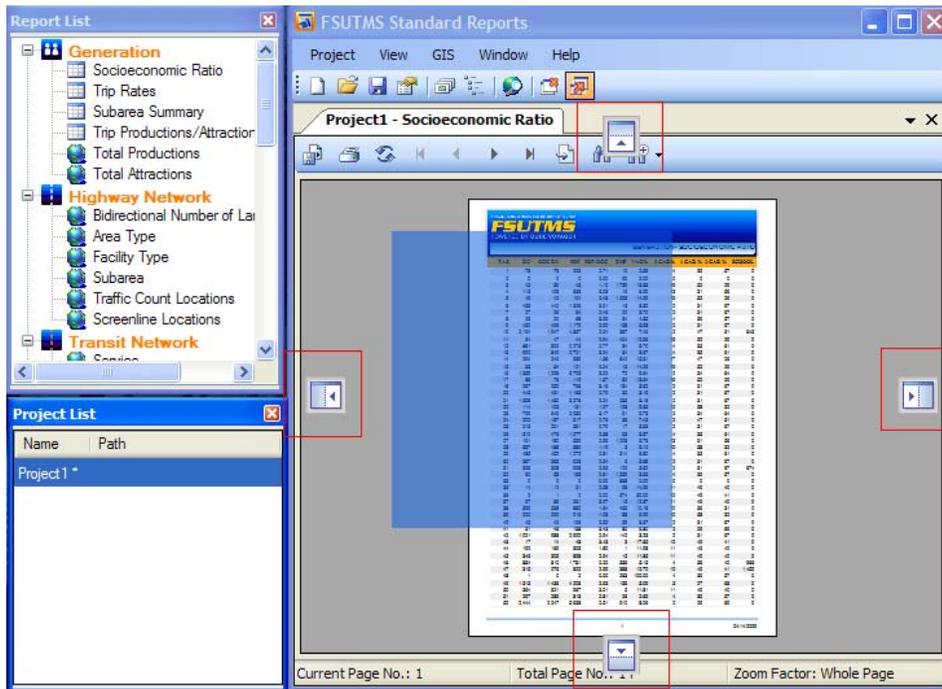


Figure 3-11: Shadow and Directional Icons

By dragging the window and then placing the cursor on one of the four icons, a new shadow will appear in the selected location, as shown in Figure 3-12. When the mouse button is released, the Project List window will be docked to the right side of the main window, as shown in Figure 3-13.

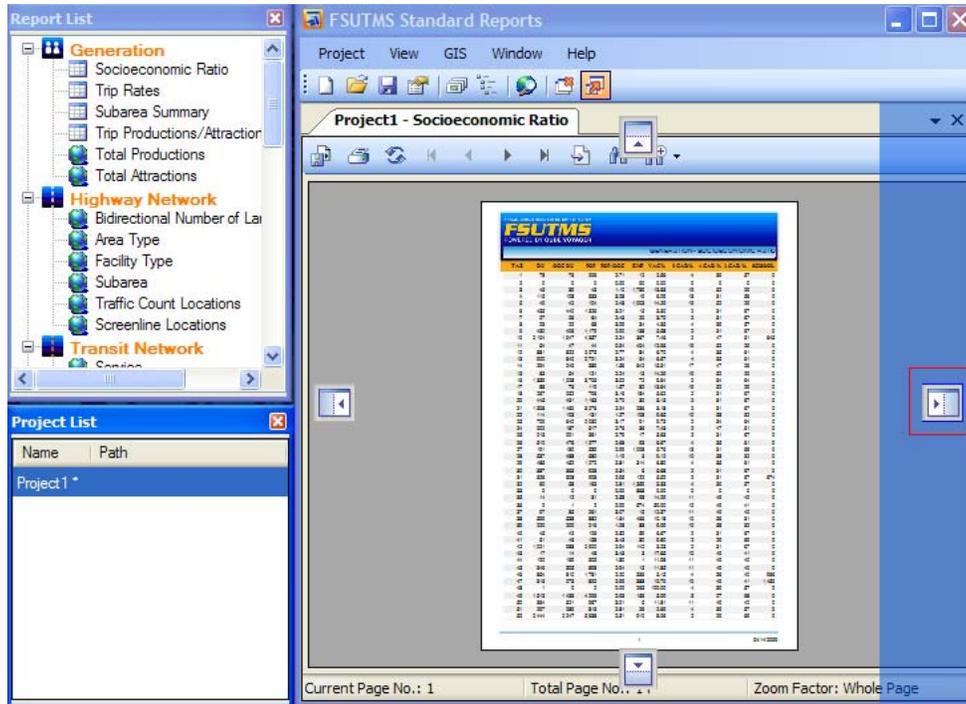


Figure 3-12: Cursor Showing Shadow of Dock Position

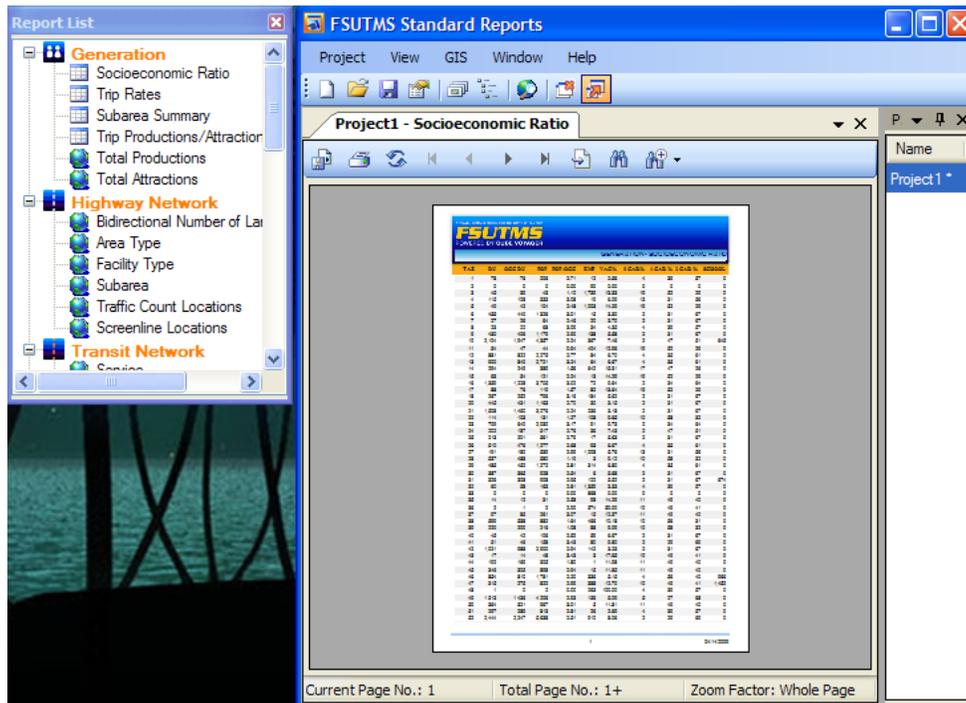


Figure 3-13: Project List Window Docked to Main Window

3.1.5. Closing a Window

A window can be easily closed by simply clicking the “x” button on the top-right corner. To close all windows at once:

- Select Window|Close All, or
- Click the “Close All” icon on the toolbar as shown in Figure 3-14.

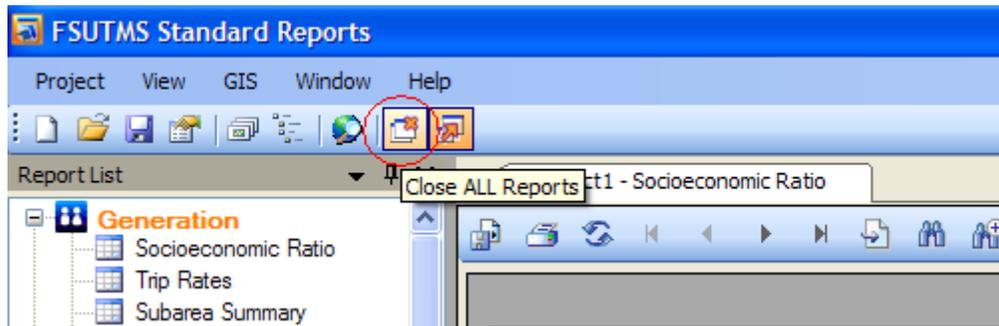


Figure 3-14: Toolbar Icon Button for "Close All"

3.1.6. Auto-Hiding a Window

The Project List and Report List windows can be changed to the auto-hide mode by clicking the Auto Hide icon at the top-right corner next to the Close button shown in Figure 3-15. After clicking this button, a vertical button will appear by the side of the window where it is docked. When the window loses its focus, it will be hidden automatically, as shown in Figure 3-16. However, once the user places the cursor over the vertical button, the window will reappear. The auto-hide status can be canceled by clicking the Auto Hide button again.

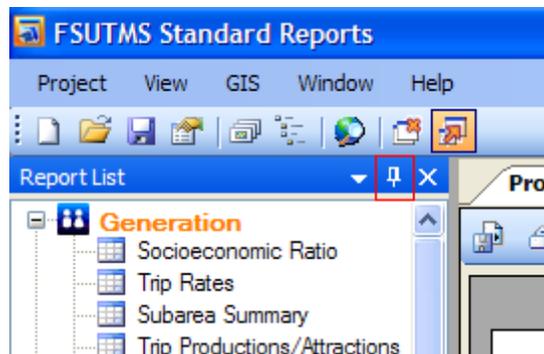


Figure 3-15: Icon Button for "Auto Hide" on Report List Window

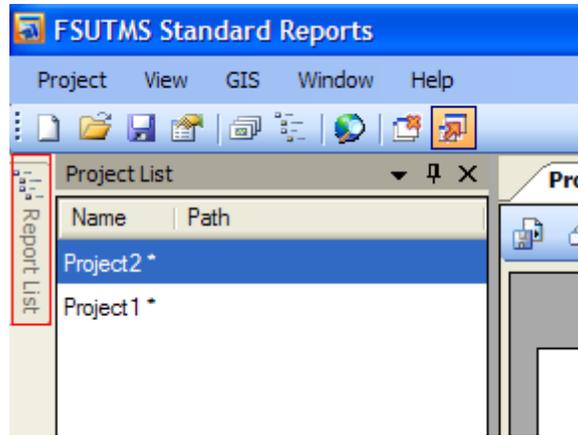


Figure 3-16: Auto Hidden Report List Window

The Auto-Hide status can also be changed by using the context menu of the Project List or Report List window. To access this menu, click the down arrow button on the top-right corner of the window next to the Auto Hide button. This menu also allows the floating, docking, or hiding of the current window. An example context menu for the Report List window is shown in Figure 3-17.

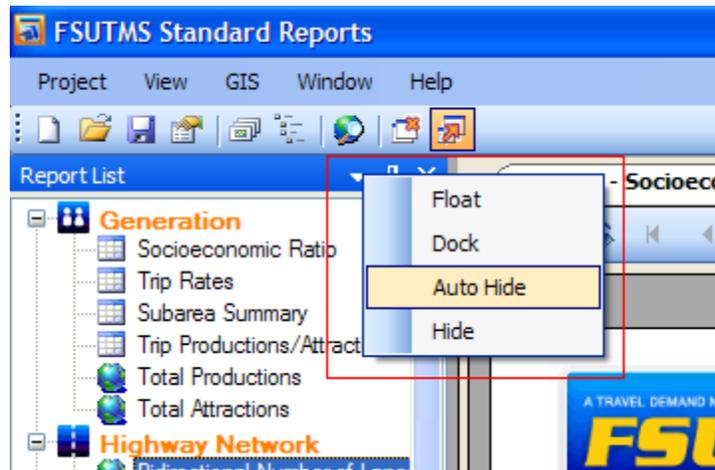


Figure 3-17: Context Menu for Report List Window

3.2. Project Management

The system manages the reports for different project scenarios using project files. A project consists of a project file and the associated project data files. The project file has the “.fsr” file extension. The “.fsr” extension is made up of the first characters of “FSUTMS Standard Reports”. A “.fsr” project file works like the catalog file (.cat) in Cube Voyager. It currently saves the project setting information, including the scenario path, the model name, and the report footer. Supplemental information can be added in future versions of the system as needed.

It is noted here that, to improve the speed of generating the reports, the data used to generate the reports are stored in the Report Data Files in XML format that serve as the placeholders for the data. In addition, some temporary data files are also created and saved for later use. The project data files are saved under a folder named “### data”, where ### is the project name. For example, a project named “Olympus” is saved with the structure shown in Figure 3-18.

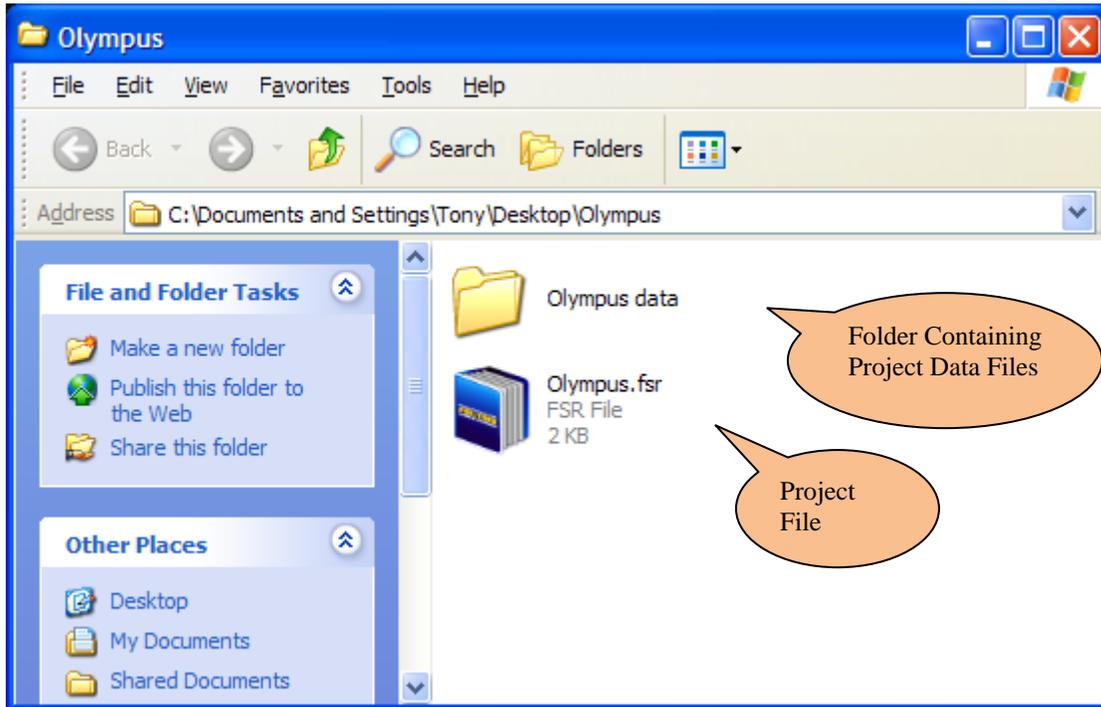


Figure 3-18: Project File and Data Files for “Olympus” Project

3.2.1. Creating a Project

To create a project:

- Select the “Create a new project” radio button in Figure 3-19 when the program is first started, or
- Select New|Project, or
- Press “Ctrl + N” shortcut keys, or
- Click the New Project icon button on the toolbar (see Figure 3-20).

Executing any one of the above will bring up the Project Settings dialog shown in Figure 3-21.

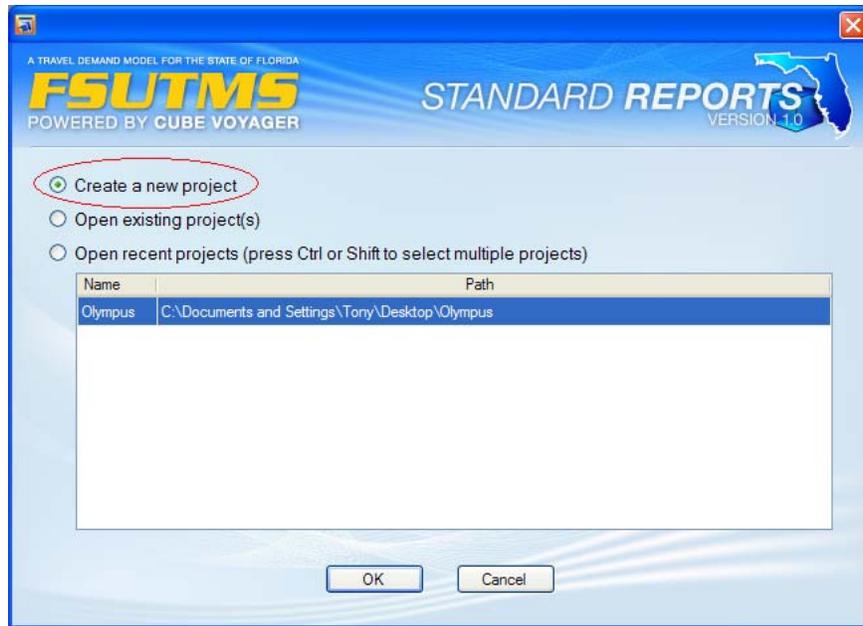


Figure 3-19: "Create a new project" Radio Button in Welcome Window

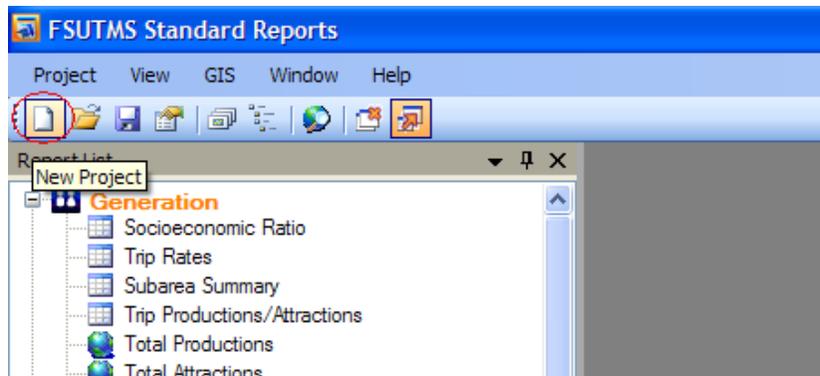


Figure 3-20: "New Project" Icon Button on Toolbar

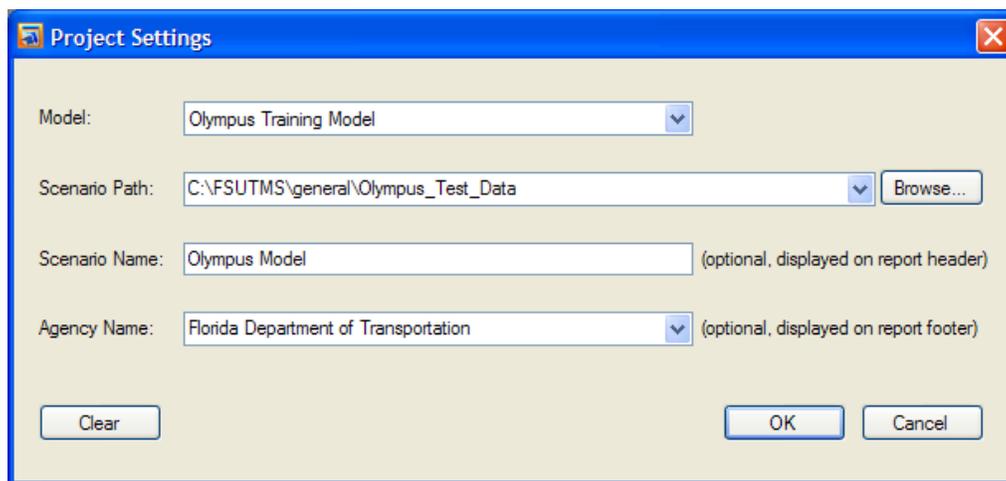


Figure 3-21: Project Setting Dialog

The Project Settings dialog box allows the following items to be set:

1. Model: the name of the model. Olympus Training model and FSUTMS models supported by this software, such as NERPM and SERPM, are listed here; this item is required for a project.
2. Scenario Path: the path to store I/O data files for a FSUTMS-CUBE scenario. It should be an existing absolute path. All the data files required to generate the reports should be in this folder or subfolders under it; this item is required for a project.
3. Scenario Name: the name of the scenario. This item is optional. If inputted, the scenario name will appear on the header of each report (see Figure 3-22).

A TRAVEL DEMAND MODEL FOR THE STATE OF FLORIDA
FSUTMS
 POWERED BY CUBE VOYAGER

Olympus Model

GENERATION - SOCIOECONOMIC RATIO

T.AZ	DU	OCC DU	POP	POP/OCC	EMP	VAC %	0 CAR %	1 CAR %	2 CAR %	SCHOOL
1	78	76	206	2.71	12	2.56	4	39	57	0
2	0	0	0	0.00	90	0.00	0	0	0	0

Figure 3-22: Scenario Name Shown as Report Header

4. Agency Name: the name of the agency creating the reports. This item is also optional. If inputted, the agency name will appear on the footer of each report (see Figure 3-23).

49	1,513	1,436	4,206	2.93	156	5.09	5	27	68	0
50	364	321	967	3.01	5	11.81	11	49	40	0
51	297	289	813	2.81	26	2.69	4	39	57	0
52	2,444	2,247	5,638	2.51	910	8.06	2	29	69	0

Florida Department of Transportation 1 04/13/2009

Figure 3-23: Agency Name Shown as Report Footer

All of the above input items can be cleared by clicking the Clear button. After the settings of a project are configured, click the “OK” button. All the required input will be checked, if there are any invalid inputs, the warning message will be shown to the user. For example, if an entered scenario path is invalid, the warning dialog shown in Figure 3-24 will appear.



Figure 3-24: Warning Message for Invalid Scenario Path

After the “OK” button is clicked, the software will try to retrieve the Model Year and Alternative Letter of the scenario from the data file names automatically. If they cannot be retrieved successfully, the software will ask the user to input them manually with a dialog shown in Figure 3-25.

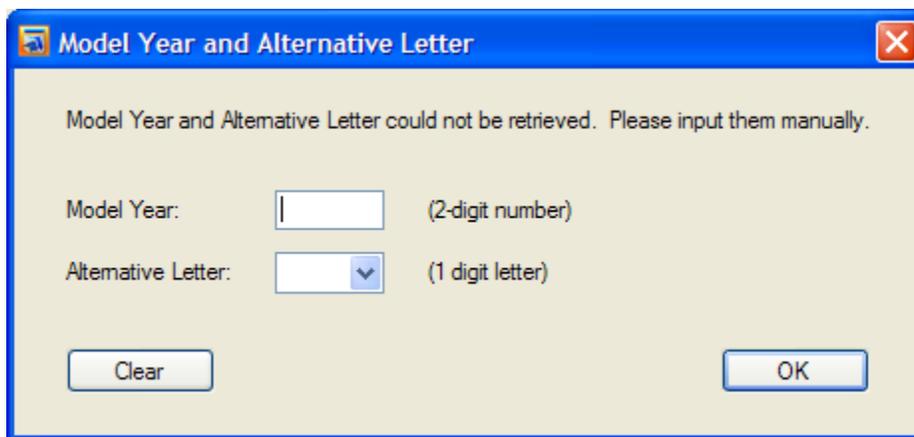


Figure 3-25: Input Model Year and Alternative Letter Dialog

After the Model Year and Alternative Letter are retrieved or inputted, the software will check if all the required data files for all the reports exist. A data file’s pathname is determined by combining the absolute scenario path defined in the Project Settings dialog with the relative path and name defined in the Data Configuration File. If data files cannot be found, the dialog shown in Figure 3-26 will appear to ask the user for verification. All the missing data files will be shown in red. For example, in Figure 3-26, there are three missing files named by “s65tazs_05.dbf”, “s65tazs_05.shp”, and “s65tazs_05.shx”, separately. To help the software to locate those files, the user can double-click each of the red nodes to select the target files. If all or most of the files are in red, then that means the scenario path may be inappropriate. It is suggested that the user should click the Back button to go back to the Project Setting dialog and set an appropriate scenario path.

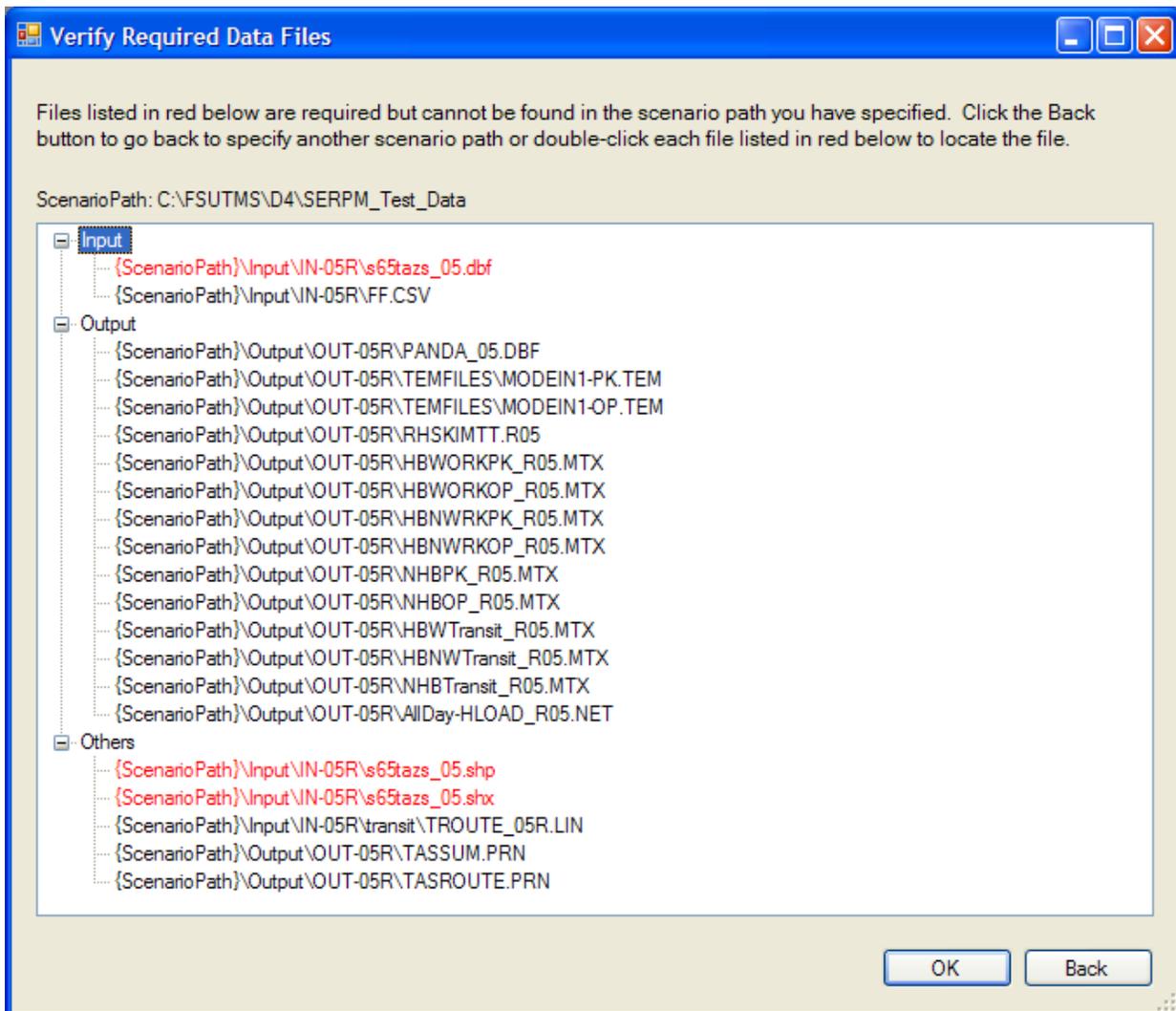


Figure 3-26: Verify Required Data Files Dialog

If the user ignores the missing data files and clicks the “OK” button to continue, a warning dialog will appear as shown in Figure 3-27. The user can then click the “No” button to go back and verify those missing files again. If the user clicks the “Yes” button to continue, the reports which use data from those missing files will be empty.

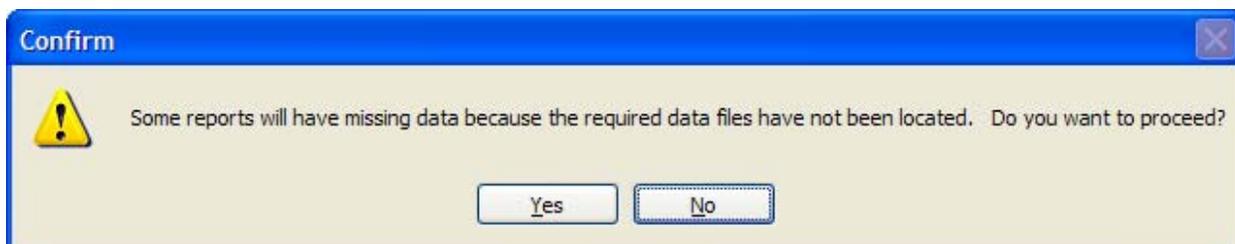


Figure 3-27: Missing Data Files Warning Dialog

After all of the above steps, a new created project with an automatically assigned name will be added to the Project List window. Symbol “*” will be put behind the project name to indicate an unsaved project. A meaningful name can be specified when the project is saved. Once the project is saved, the “*” will disappear.

3.2.2. Changing Project Settings

To change the settings of a project, first select the project in the Project List window and then:

- Select Project|Project Settings, or
- Double-click the project name in the Project List window, or
- Click the Project Settings icon button on the toolbar (see Figure 3-28).

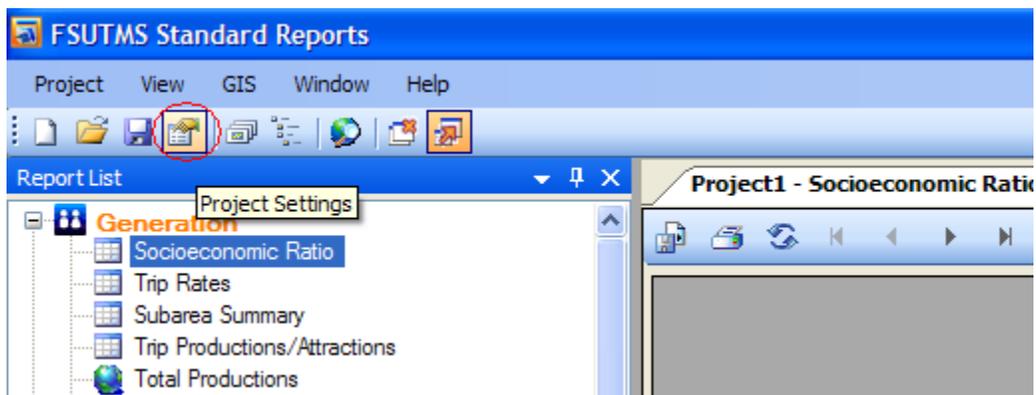


Figure 3-28: Icon Button for "Project Settings"

3.2.3. Saving a Project

To save a newly created or changed project(s), select the project(s) in the Project List window and then:

- Select Project|Save, or
- Press “Ctrl + S” shortcut keys, or
- Select Project|Save As... to save the project with a different path and/or name, or
- Click the “Save” icon button on the toolbar (see Figure 3-29).

Any of these options will bring up the Save Project dialog box shown in Figure 3-30. This dialog allows the project name and project path to be specified and saved. The dialog also includes a checkbox called “Create directory for project”. If this checkbox is checked, a folder with the same name as the project name will be created under the project path. The project file and all data files for this project will be saved under this folder. The checkbox is selected by default. This default option is recommended to separate multiple projects, allowing for better-organized files when they are saved to the same folder.

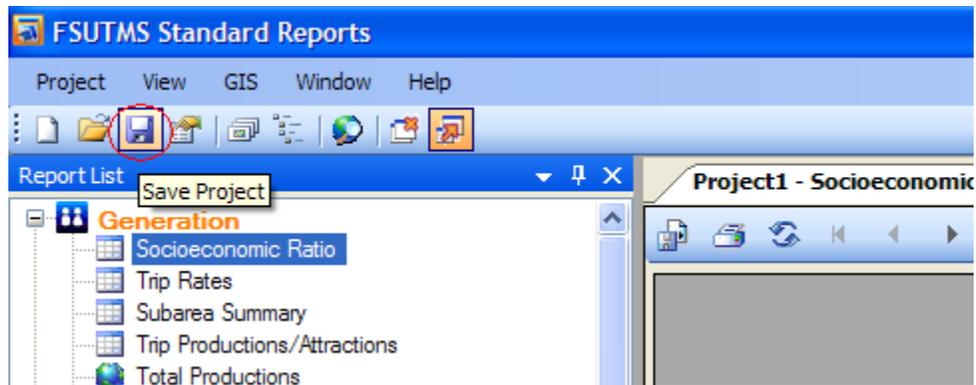


Figure 3-29: Icon for "Save Project"

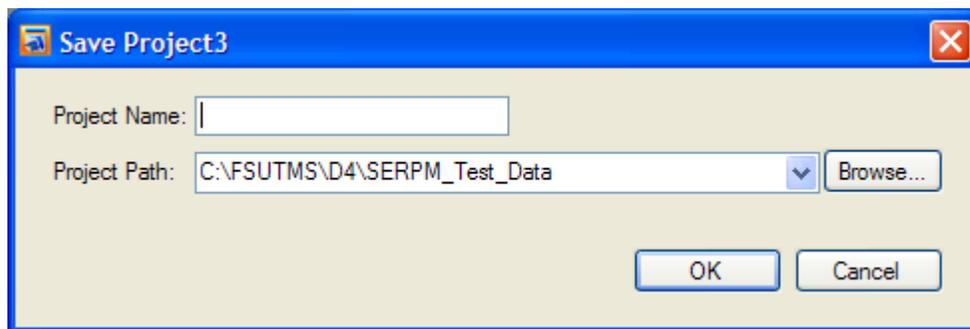


Figure 3-30: Save Project Dialog

After an unsaved project is saved, the star "*" behind the project name will disappear to indicate that the project has been saved.

3.2.4. Opening a Project

A project can be opened via one of the following options:

1. Selecting the "Open existing project(s)" radio button shown in Figure 3-31 when the program is first started.
2. Selecting the "Open recent projects" radio button as shown in Figure 3-32 when the program first started. Double-click a project in the project list to open the project directly. Multiple recent projects can be selected by first pressing the Ctrl or Shift key, and then clicking the "OK" button to open.
3. Select Open|Project.
4. Press "Ctrl + O" shortcut keys.
5. Click the Open Project icon on the toolbar (see Figure 3-33).

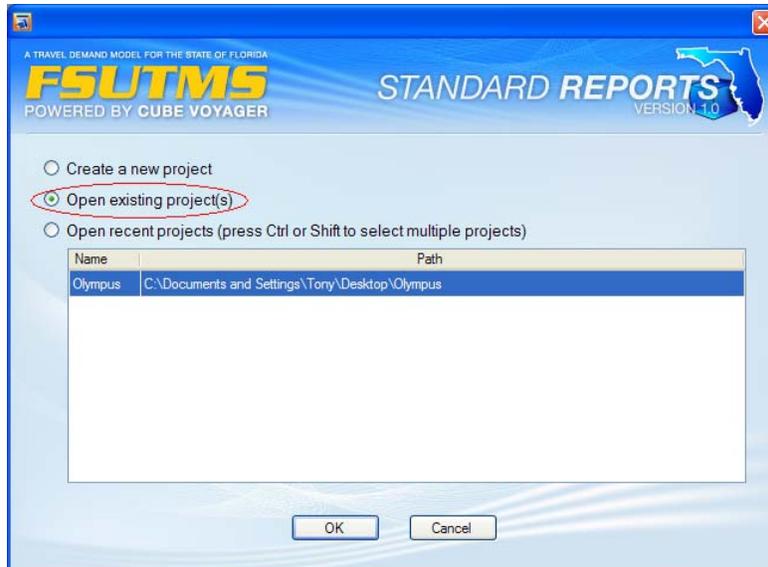


Figure 3-31: Radio Button for “Open existing project(s)”

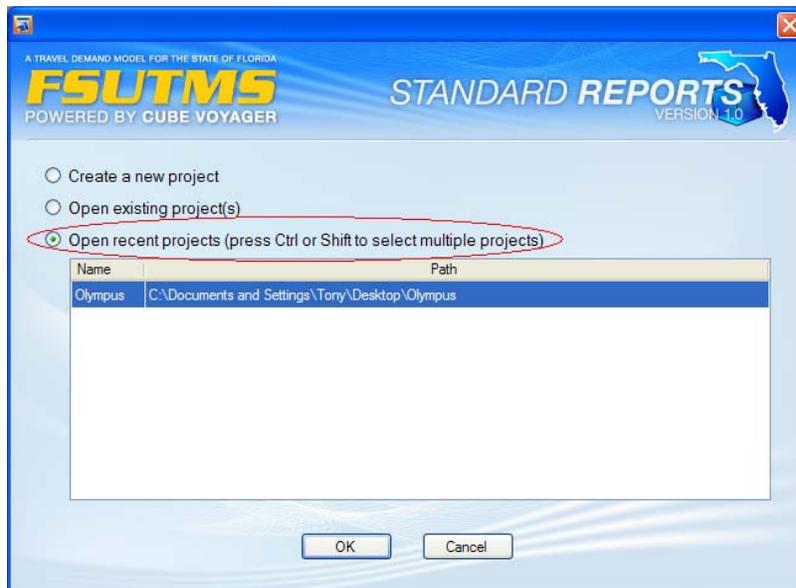


Figure 3-32: Radio Button for “Open recent projects”

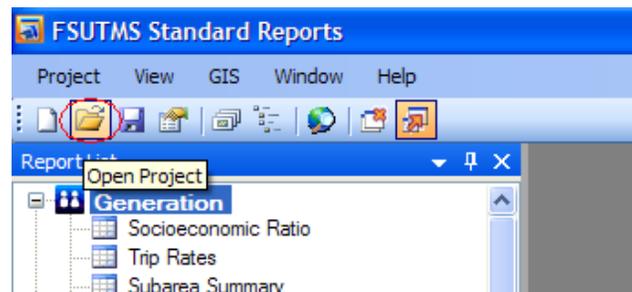


Figure 3-33: Tool Icon Button for Open Project

When no specific project(s) is selected, an Open dialog as shown in Figure 3-34 will pop up. Select a project file (*.fsr) from this dialog. Multiple project files in the same folder can be selected by pressing the Ctrl or Shift key. Once a project(s) is selected and opened, its name(s) and path(s) will be added to the Project List window.

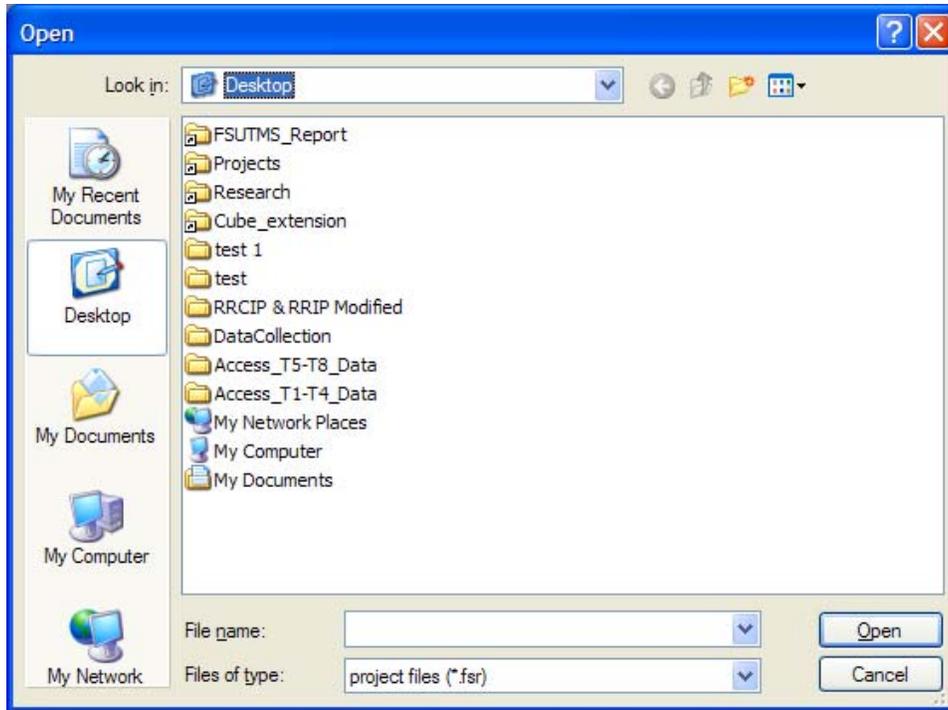


Figure 3-34: Open Project File Dialog

3.2.5. Closing a Project

To close a project(s), first select it in the Project List window and then select Project|Close. A closed project will be removed from the Project List window. If there are any unsaved projects, a reminder as shown in Figure 3-35 will pop-up. The user can choose to cancel the Close operation by clicking the Cancel button.

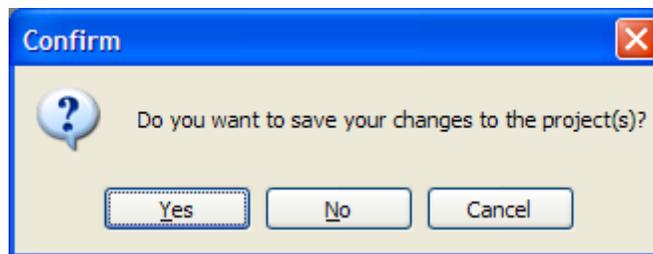


Figure 3-35: Reminder Message for Unsaved Projects

3.3. Report Operations

A report operation toolbar is available on the top of the Report Tab window. As shown in Figure 3-36, the following tool buttons are included in the toolbar:

- *Export Report:* export current report to an external file format (e.g., *.xls)
- *Print Report:* print current report
- *Refresh Report:* refresh current report
- *Go to First Page:* show first page of current report
- *Go to Previous Page:* show previous page of current report
- *Go to Next Page:* show next page of current report
- *Go to Last Page:* show last page of current report
- *Go to Page:* go to user-defined page of current report
- *Find Text:* search user-defined text in current report
- *Zoom:* zoom current report to a preset or user-selected level

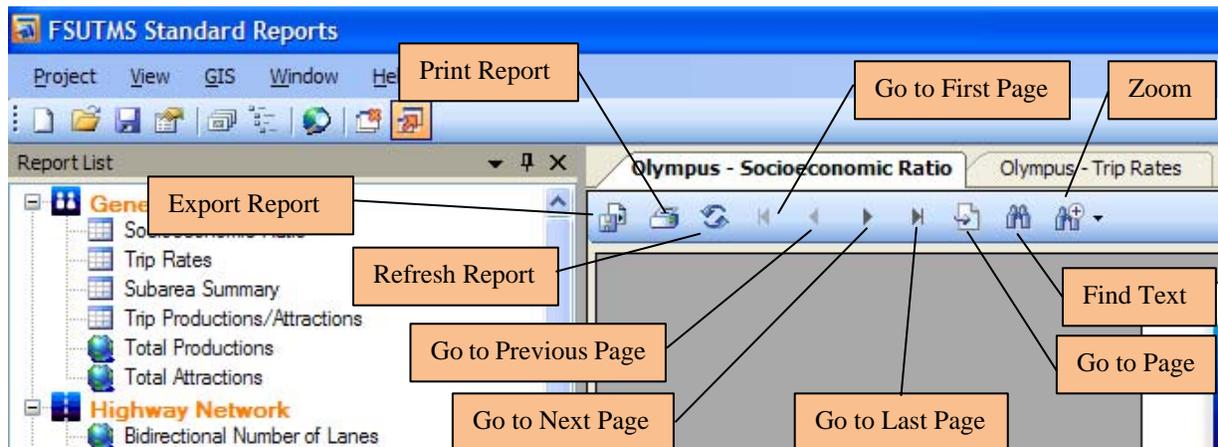


Figure 3-36: Report Operations Toolbar

3.4. GIS Map Settings

The system currently generates a total of 14 GIS map reports. Each map covers one full report page. This subsection describes the functions available to manipulate and customize the maps.

3.4.1. Selecting Additional Map Areas

By default, for each map report the system will include one map that covers the entire study area. The user may zoom into specific map areas and have them included as additional maps in the report. To define an additional map area:

- Select GIS|Select Additional Map Areas, or
- Click the Select Additional Map Areas icon button on the toolbar (see Figure 3-37).

This will open the window shown in Figure 3-38. When no additional areas have been defined, the screen will display some basic instructions on the left side of the screen.

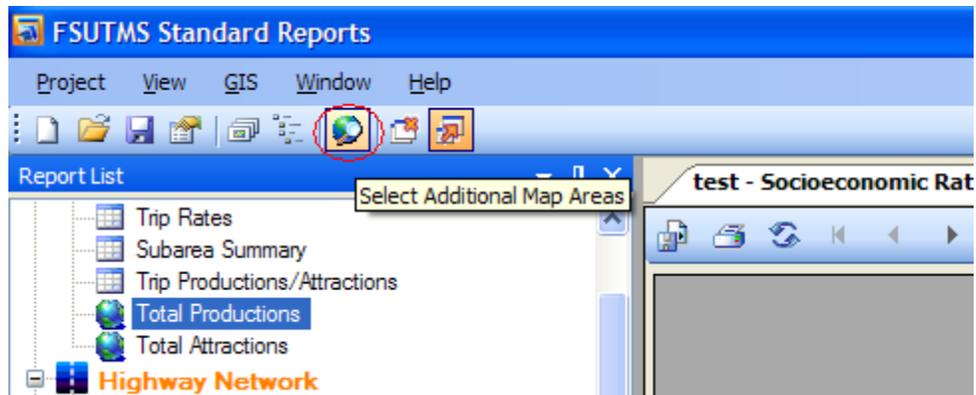


Figure 3-37: Icon Button for Selecting Additional Map Areas

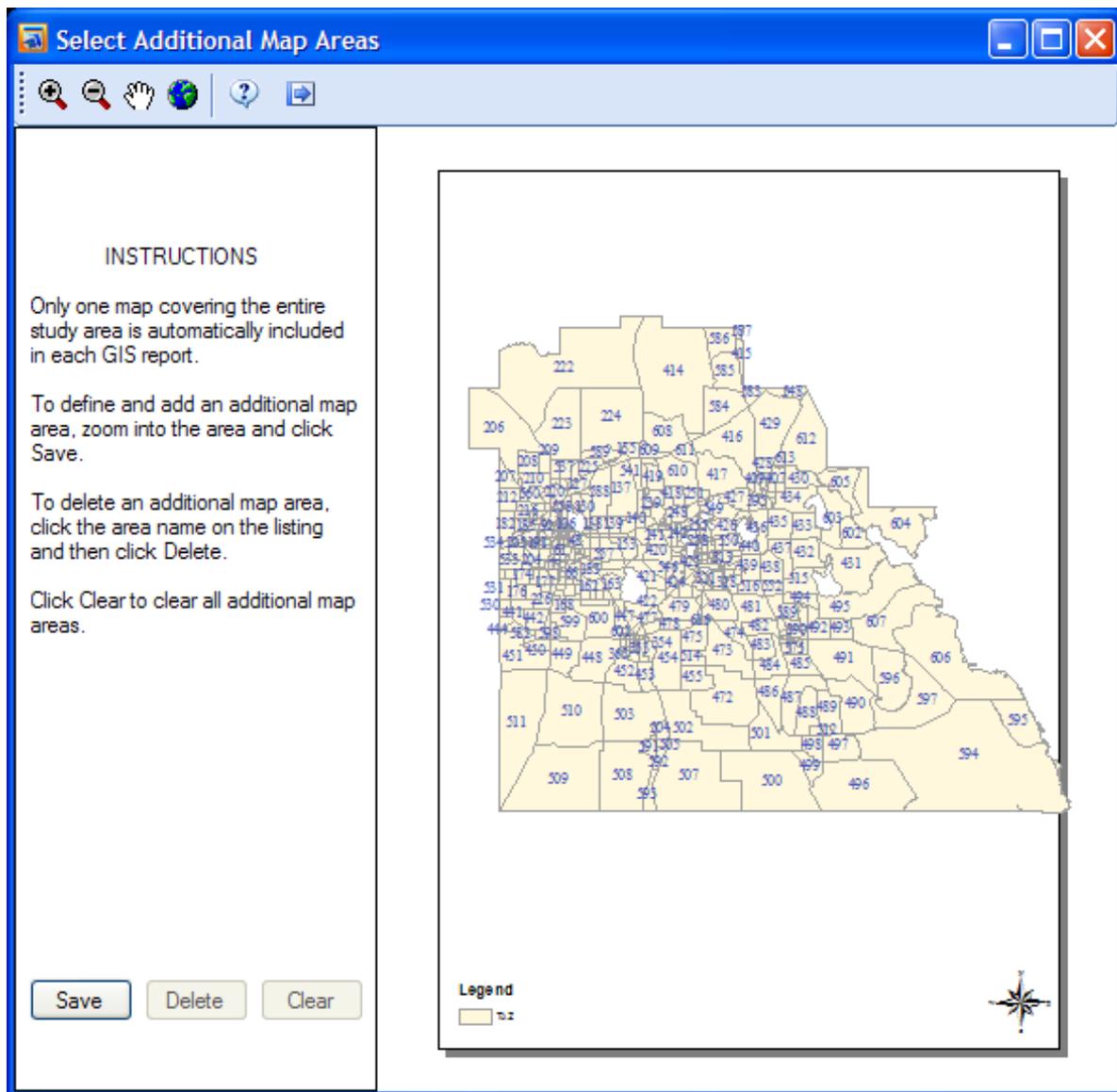


Figure 3-38: Interface for Additional Map Area Selection (with Instructions)

The system provides the following tool buttons to navigate the map for the purpose of defining map areas:

- *Zoom In*: Clicking this button and then moving the mouse cursor to the map window, the cursor will change to a magnifying glass with a plus sign. Left-click and hold the mouse button to drag a rectangle that defines the desired area. Release the mouse button to zoom into the defined area.
- *Zoom Out*: Clicking this button and moving the mouse cursor to the Map window, the cursor will change to a magnifying glass showing a minus sign. Left-click and hold the mouse button to drag a rectangle that defines a desired area to zoom out. Release the mouse button to zoom out from the defined area.
- *Pan*: Clicking this button and moving the mouse to the map window, the cursor will change to a hand shape. Click and drag a center location of the map area, move the hand-shaped mouse cursor to a new location on the map, and then release to place the selected center location in the new location.
- *Full Extent*: Click this button to display the entire map.

Once a desired map area is displayed in the map window, click the “Save” button at the bottom of the left window. The location of the new map area will be saved and the saved area will be listed under a general, sequential name, along with the time and date when the area was saved (see Figure 3-39). The default name will be “Additional Area #”, in which “#” is a number. It can be more meaningfully renamed by clicking a selected name node and then inputting a new name.

To delete a map area, click the map area name on the list to highlight it and then click the “Delete” button at the bottom to remove the map area. Click the “Clear” button to clear all additional map areas.

Two additional tool buttons listed on the toolbar are:

1. The “Help” tool button, which brings up a window that shows the same general instructions as described above (see Figure 3-40), and
2. The “Done” tool button to exit from the map area selection screen and return to the system main screen.

3.4.2. Selecting Map Quality

The image quality of GIS maps can be set by first selecting GIS|Select Map Quality and then selecting a preferred quality level. Three levels are available: high, medium, and low. By default, the medium level is selected. Higher quality images will require a longer time to generate and will consume more memory. The later increases the chance of exhausting the computer memory, causing the system to shut down, when too many reports are opened at the same time.

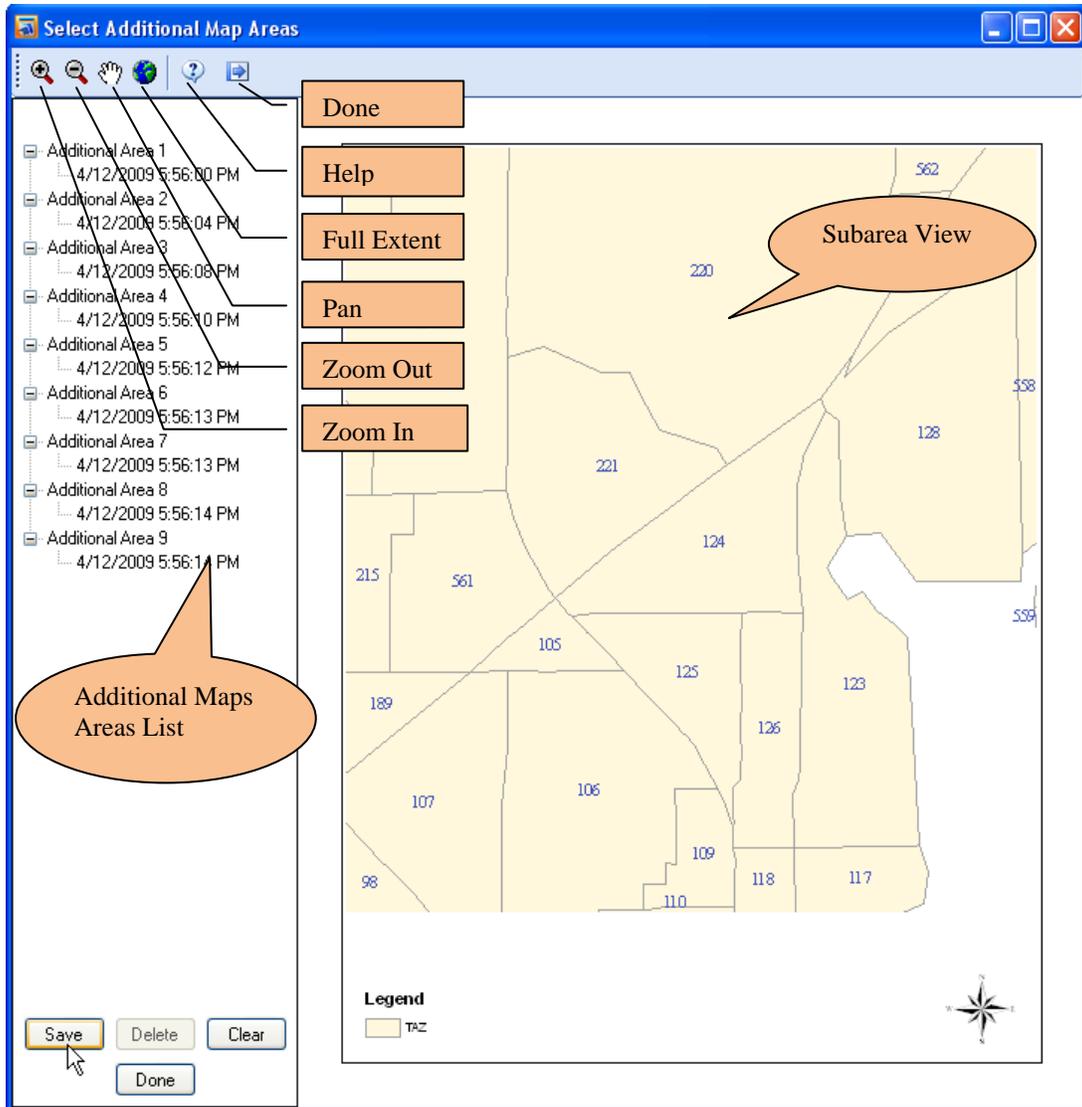


Figure 3-39: Interface for Defining Additional Map Areas (with Selected Map Areas)

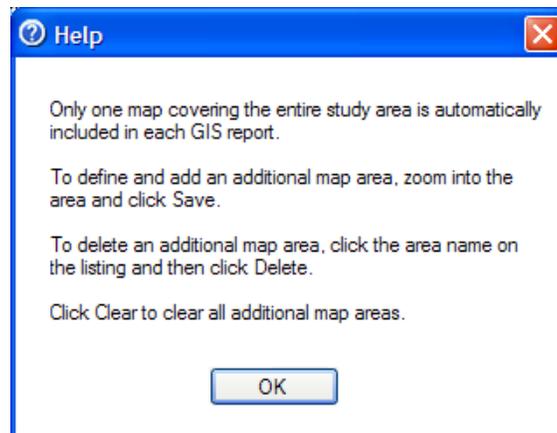


Figure 3-40: Help Dialog for Selecting Additional Map Areas

3.4.3. Generating MXD and Geodatabase Files

The system allows an MXD file and all the layers for all the GIS map reports to be exported. To generate these files, select GIS|Generate MXD and Geodatabase Files. A dialog as shown in Figure 3-41 will pop up to remind the user that the process to generate the files will take several minutes. The MXD file will store all the layers' specifications and settings, and the Geodatabase file will store all the map layers.

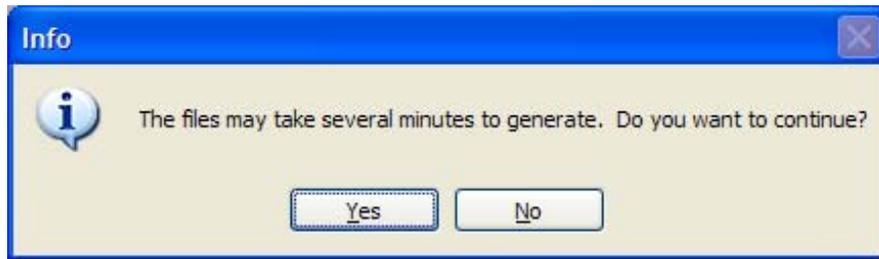


Figure 3-41: Message to Remind User of Long MXD and Geodatabase File Generation Time

Clicking the “Yes” button in Figure 3-41 will start the file exportation process. After the files are exported, the Save As dialog shown in Figure 3-42 will pop up to allow the MXD file to be named and saved to a select folder.

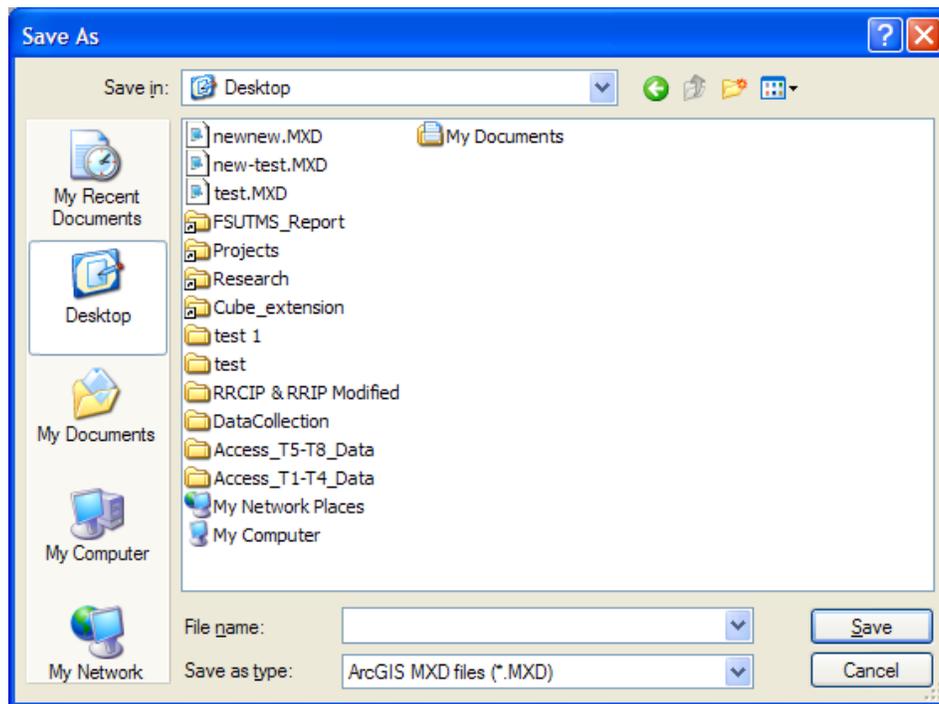


Figure 3-42: Save MXD File Dialog

After the MXD file is saved, the message box shown in Figure 3-43 will appear informing the user that the MXD file with the user-specified name has been saved to the user-specified path. The same message box also indicates that the Geodatabase files are saved to the same folder. In the

message box, by default, a checkbox to allow the user to open the saved folder is selected. Click “OK” to open the folder, or uncheck the box and then click “OK” to go back to the main window.

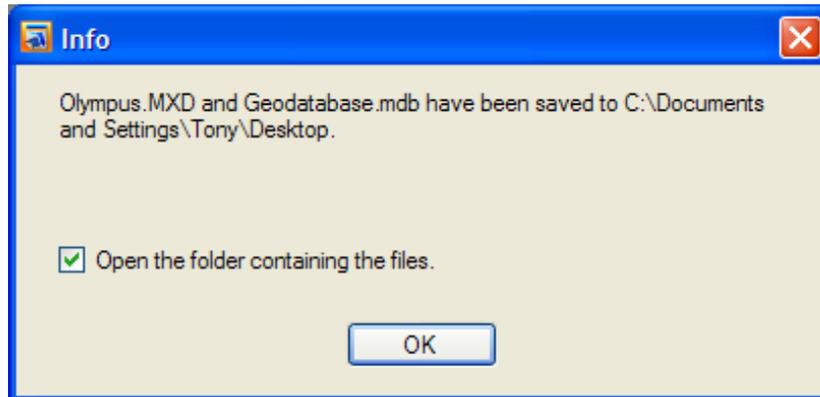


Figure 3-43: Message Box after MXD and Geodatabase Files are Successfully Exported

The user can manipulate the MXD and Geodatabase files exported with any GIS software supporting MXD files. Figure 3-44 shows the MXD file opened in ArcMap. In the ArcMap legend box, the first level of the legend lists the individual reports. Clicking on the “+” box of a specific report will display all the layers and any associated pre-defined themes for each layer.

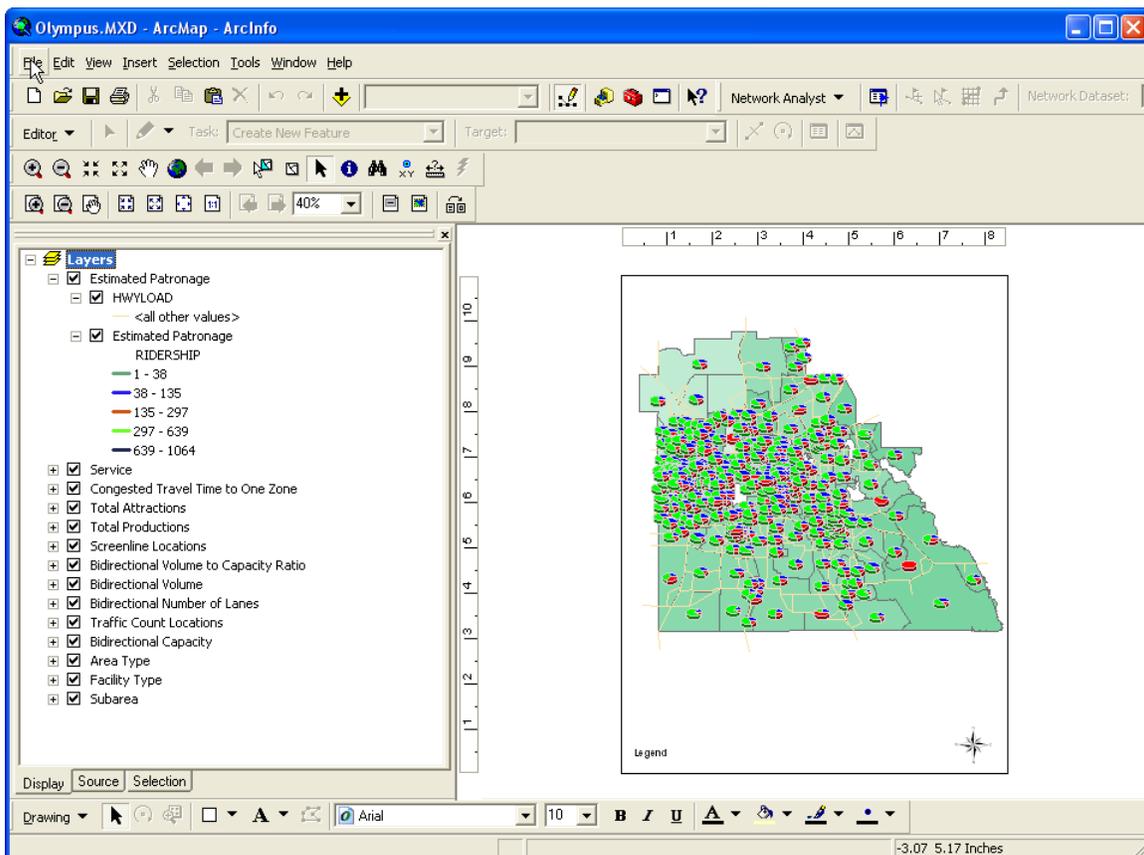


Figure 3-44: Generated MXD File Opened in ArcMap

4. REPORT DETAILS

In this section, taking the Olympus Training model as an example, the data files and the steps applied to obtain the input data for each of the 42 standard reports included in this version of the system are documented in detail. The associated modeling step for each report is also indicated. Any differences between the current reports and the original report templates developed by Garnett Fleming are noted.

4.1. Socioeconomic Ratio

Model Step: Generation

Data Files Required:

- Input\ZONEDATA_YYA.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Extract and calculate the following report data with input from the ZONEDATA_YYA.DBF file using the following formulas:

Column Name	Calculation Formula
<i>TAZ</i>	ZONE
<i>DU</i>	SFDU + MFDU + HMDU
<i>OCC_DU</i>	$SFDU * (1 - SF_PCTVAC)\% + MFDU * (1 - MF_PCTVAC)\% + HMDU * HMOCC\%$
<i>POP</i>	SFPOP + MFPOP + HMPOP
<i>POP/OCC</i>	POP / OCC_DU
<i>EMP</i>	TOT_EMP
<i>VAR %</i>	$(DU - OCC_DU) / DU * 100$
<i>0 CAR %</i>	$(SFDU * SF_0AUTO\% + MFDU * MF_0AUTO\%) / DU * 100$
<i>1 CAR %</i>	$(SFDU * SF_1AUTO\% + MFDU * MF_1AUTO\%) / DU * 100$
<i>2 CAR %</i>	$(SFDU * SF_2AUTO\% + MFDU * MF_2AUTO\%) / DU * 100$
<i>SCHOOL</i>	SCHOOL

4. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

Notes:

- Duplicate column “VAC%” was removed.
- Column “3 CAR%” was replaced with column “1 CAR%” and column “2 CAR%”.
- Column “TRK/EMP” was replaced with column “SCHOOL”.

4.2. Trip Rates

Model Step: Generation

Data Files Required:

- Input\ZONEDATA_yya.DBF
- Output\PANDA_ayy.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Calculate total population, dwelling units, and employees in all zones, respectively, with fields in the file Input\ZONEDATA_yya.DBF:

Variable Name	Calculation Formula
TOTAL_POP	Sum (SFPOP + MFPOP + HMPOP)
TOTAL_DU	Sum (SFDU + MFDU + HMDU)
TOTAL_EMP	Sum (TOT_EMP)

4. Calculate total trips for each trip purpose by summing the responding fields in file Output\PANDA_ayy.DBF:

Variable Name	Calculation Formula
HBW_TRIPS	Sum (HBWP)
HBNW_TRIPS	Sum (HBSHP + HBSRP + HBOP)
NHB_TRIPS	Sum (NHBP)
TRK_TRIPS	Sum (TRKTAXIP)
IE_TRIPS	Sum (IEP)

5. Calculate each column of the report with the fields of input file ZONEDATA_yya.DBF by the following formulas:

Column Name	Calculation Formula
<i>DESCRIPTION</i>	Hard coding text
<i>HBW</i>	$HBW_TRIPS / TOTAL_POP$, $HBW_TRIPS / TOTAL_DU$, $HBW_TRIPS / TOTAL_EMP$
<i>HBNW</i>	$HBNW_TRIPS / TOTAL_POP$, $HBNW_TRIPS / TOTAL_DU$, $HBNW_TRIPS / TOTAL_EMP$
<i>NHB</i>	$NHB_TRIPS / TOTAL_POP$, $NHB_TRIPS / TOTAL_DU$, $NHB_TRIPS / TOTAL_EMP$
<i>TRKTAXI</i>	$TRK_TRIPS / TOTAL_POP$, $TRK_TRIPS / TOTAL_DU$, $TRK_TRIPS / TOTAL_EMP$
<i>IE</i>	$IE_TRIPS / TOTAL_POP$, $IE_TRIPS / TOTAL_DU$, $IE_TRIPS / TOTAL_EMP$

6. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

Notes:

- The name of the column “TRK” has been changed to “TRKTAXI”.

4.3. Subarea Summary

Model Step: Generation

Data Files Required:

- Input\ ZONEDATA_yya.DBF
- Output\PANDA_ayy.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Copy input file, Input\ZONEDATA_yya.DBF, to output folder for temporary use.
4. Link ZONEDATA_yya.DBF with output file, Output\PANDA_ayy.DBF, to get production and attraction values for each trip purpose in each subarea.

Column Name	Calculation Formula
<i>SUABAREA</i>	SECTOR
<i>HBW-P</i>	SUM (HBWP) GROUP BY SECTOR
<i>HBNW-P</i>	SUM (HBSHP + HBSRP + HBOP) GROUP BY SECTOR
<i>NHB-P</i>	SUM (NHBP) GROUP BY SECTOR
<i>HBW-A</i>	SUM (HBWA) GROUP BY SECTOR
<i>HBNW-A</i>	SUM (HBSHA + HBSRA + HBOA) GROUP BY SECTOR
<i>NHB-A</i>	SUM (NHBA) GROUP BY SECTOR

5. Delete the temporary file: Output\ ZONEDATA_yya.DBF.
6. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.4. Trip Productions/Attractions

Model Step: Generation

Data Files Required:

- Output\PANDA_ayy.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Retrieve production and attraction for each trip purpose in each zone from output file: Output\PANDA_ayy.DBF

Column Name	Calculation Formula
<i>ZONE</i>	Z
<i>PRODUCTION – HBW</i>	HBWP
<i>PRODUCTION – HBNW</i>	HBSHP + HBSRP + HBOP
<i>PRODUCTION – NHB</i>	NHBP
<i>HBW ATTRACTION – UNBAL</i>	
<i>HBW ATTRACTION – BAL</i>	HBWA
<i>HBW ATTRACTION – DIF%</i>	
<i>HBNW ATTRACTION – UNBAL</i>	
<i>HBNW ATTRACTION – BAL</i>	HBSHA + HBSRA + HBOA
<i>HBNW ATTRACTION – DIF%</i>	
<i>NHB ATTRACTION – UNBAL</i>	
<i>NHB ATTRACTION – BAL</i>	NHBA
<i>NHB ATTRACTION – DIF%</i>	

4. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

Notes:

- The report name has been changed from “GENERATION – SUMMARY” to “GENERATION – TRIP ATTRATIONS/PRODUCTIONS”.
- The data for the columns “UNBAL” and “DIF%” for each trip purpose requires assistance from Citilabs, the developer, and cannot be retrieved at this point.

4.5. Total Production by Purpose

Model Step: Generation

Data Files Required:

- Output\PANDA_ayy.DBF
- Input\ZONEDATA_yya.DBF
- Input\ZONEDATA_yya.shp
- Input\ZONEDATA_yya.shx

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user the missing files.
3. Copy Input\ZONEDATA_yya.shp and Input\ZONEDATA_yya.shx to project data folder.
4. Generate new ZONEDATA_yya.DBF in project data folder with productions information from Input\ZONEDATA_yya.DBF and Output\PANDA_ayy.DBF if it has been outdated. The calculations of the fields are listed as follows:

Field in New ZONEDATA_yya.DBF	Calculation Formula
ZONE_ID	Z
HBW-P	HBWP
HBNW-P	HBSHP + HBSRP + HBOP
NHB-P	NHBP
PRODUCTION	HBW-P + HBNW-P + NHBP

5. Generate the image file (if it is outdated) for the GIS map with the files ZONEDATA_yya.shp, ZONEDATA_yya.shx, and ZONEDATA_yya.DBF. For each zone, the percentage of the production for each trip purpose is calculated by dividing the production for that trip purpose by the total production in that zone.
6. Load the image file(s) into the report template.

4.6. Total Attractions by Purpose

Model Step: Generation

Data Files Required:

- Output\PANDA_ayy.DBF
- Input\ZONEDATA_yya.DBF
- Input\ZONEDATA_yya.shp
- Input\ZONEDATA_yya.shx

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Copy Input\ZONEDATA_yya.shp and Input\ZONEDATA_yya.shx to project data folder.
4. Generate new ZONEDATA_yya.DBF in project data folder with attractions information from Input\ZONEDATA_yya.DBF and Output\PANDA_ayy.DBF if it is outdated. The calculations of the fields are listed below:

Field in New ZONEDATA_yya.DBF	Calculation Formula
ZONE_ID	Z
HBW-A	HBWA
HBNW-A	HBSHA + HBSRA + HBOA
NHB-A	NHBA
ATTRACTION	HBW-A + HBNW-A + NHBA

5. Generate the image file(s) (if it has been outdated) for the GIS map(s) with data from the files, ZONEDATA_yya.shp, ZONEDATA_yya.shx, and ZONEDATA_yya.DBF. For each zone, the percentage of the attraction for each trip purpose is calculated by dividing the attraction for that trip purpose by the total attraction in that zone.
6. Load the image file(s) to the report template.

4.7. Bidirectional Number of Lanes

Model Step: Highway Network

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to the next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and then convert highway link and node DBF files from highway network file, first. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager Engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. Modify the link DBF file (Hwyload.dbf) to convert the number of lanes from directional to bidirectional. For example, if one link from A to B has two lanes and the reversed link from B to A has three lanes, the bidirectional number of lanes for the highway between A and B is five.

8. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg) for the GIS map if it is outdated. The field in Hwyload.dbf used for this report is “LANES”.
9. Load GIS image file(s) into the report template.

Notes:

- The number of lanes was changed from directional to bidirectional for proper map display.

4.8. Area Type

Model Step: Highway Network

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and then convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it is outdated. The field in Hwyload.dbf used for this report is "ATYPE".
8. Load GIS image file(s) into the report template.

4.9. Facility Type

Model Step: Highway Network

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and then convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it has been outdated. The field in Hwyload.dbf used for this report is "FTYPE".
8. Load GIS image file(s) into the report template.

4.10 Subarea

Model Step: Highway Network

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it has been outdated. The field in Hwyload.dbf used for this report is "DISTRICT".
8. Load GIS image file(s) into the report template.

4.11. Traffic Count Locations

Model Step: Highway Network

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to the next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder first and convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it has been outdated. The field in Hwyload.dbf used for this report is "COUNT".
8. Load GIS image file(s) into the report template.

4.12. Screenline Locations

Model Step: Highway Network

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder first and convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it has been outdated. The field in Hwyload.dbf used for this report is "SCREENLINE".
8. Load GIS image file(s) into the report template.

4.13. Transit Service

Model Step: Transit Network

Data Files Required:

- Output\HWYLOAD_yya.net
- Input\TROUTE_yya.LIN
- Output\tasum_ayy.PRN

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link and node DBF files from highway network file. An example to generate highway node DBF file (Hwyload_node.dbf) is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. Generate transit route files (Transit.shp, Transit.shx, and Transit.dbf) with Hwyload_node.dbf, Input\TROUTE_yya.LIN, and Output\tasum_ayy.PRN, and save the route files to the project data folder. The name and past nodes of each route can be

retrieved from Input\TROUTE_yya.LIN. The spatial information of the nodes can be extracted from the file Hwyload_node.dbf. Output\tasum_ayy.PRN contains the ridership of each route.

8. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) and route shape files (Transit.shp, Transit.shx, and Transit.dbf) generated above, generate the image file for the GIS map if it is outdated.
9. Load GIS image file(s) into the report template.

4.14. Speed Curve Lookup

Model Step: Transit Network

Data Files Required:

- Parameters\TRANSPD.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read the speed curve information from the parameter file: Parameters\TRANSPD.DBF. The fields include:

Field in TRANSPD.dbf	Description
CURVE_NO	the speed curve ID
LOW_MODE	the lowest value of the mode range
HIGH_MODE	the highest value of the mode range
LOW_FT	the lowest value of the facility
type range	Error! No table of figures entries found.
HIGH_FT	the highest value of the facility type range
LOW_AT	the lowest value of the area type range
HIGH_AT	the highest value of the area type range

4. Convert the speed curve information read in the above step to the format in the report, and save the converted data to the database.
5. Save all extracted data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.15. Speed Curve Graph

Model Step: Transit Network

Data Files Required:

- Parameters\TRANSPD.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read the speed ratio information from the parameter file: Parameters\TRANSPD.DBF. The fields include:

Field in TRANSPD.dbf	Description
CURVE_NO	the speed curve ID
SPEED_RATIO	the speed ratio between transit speed and highway speed

4. Convert the speed ratio information read in the previous step to the format in the report, and save the converted data to the database.
5. Save all extracted data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.16. Input vs. Output Attractions

Model Step: Distribution

Data Files Required:

- Output\PSNTRIPS_ayy.MAT
- Output\PANDA_ayy.DBF

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the person trips DBF file for each trip purpose (HBW.dbf, HBSH.dbf, HBSR.dbf, HBO.dbf, NHB.dbf, TRKTAXI.dbf, and IE.dbf) exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert DBF file for each trip purpose from matrix file Output\PSNTRIPS_ayy.MAT. An example to generate the file IE.dbf is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\IE.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\PSNTRIPS_B00.MAT"
FILLMW MW[1]=mi.1.7
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate output attractions for each trip purpose (ATTR_HBW, ATTR_HBSH, ATTR_HBSR, ATTR_HBO, ATTR_NHB, ATTR_TRKTAXI, and ATTR_IE) in each zone from the DBF files converted from the output file: Output\PSNTRIPS_ayy.MAT. This is done by summing up the trips from all other zones for each zone.
7. Retrieve total input attractions for each zone from output file: Output\PANDA_ayy.DBF. The fields include:

Field in PANDA_ayy.dbf	Description
Z	the zone ID
TOTA	the total attraction of a zone

8. Get the data for each column in the report with the following correspondence and save them to the database:

Column Name	Calculation Formula
Zone	Z

<i>Total Input Attractions</i>	TOTA
<i>Total Output Attractions</i>	ATTR_HBW + ATTR_HBSH + ATTR_HBSR + ATTR_HBO + ATTR_NHB + ATTR_TRKTAXI + ATTR_IE
<i>Ratio</i>	<i>Total Output Attractions / Total Input Attractions</i>

9. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.17. Intrazonal Vs. Total Trips

Model Step: Distribution

Data Files Required:

- Output\PSNTRIPS_ayy.MAT

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the person trips DBF file for each trip purpose (HBW.dbf, HBSH.dbf, HBSR.dbf, HBO.dbf, NHB.dbf, TRKTAXI.dbf, and IE.dbf) exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert DBF file for each trip purpose from matrix file Output\PSNTRIPS_ayy.MAT. An example to generate the file IE.dbf is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\IE.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\PSNTRIPS_B00.MAT"
FILLMW MW[1]=mi.1.7
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the intrazonal and interzonal trips for each trip purpose from the .dbf files converted from this output file: Output\PSNTRIPS_ayy.MAT, and save the data to the database. For each trip purpose, sum up the trips between different zones to calculate the interzonal trips, and sum up the trips between the same zones to get the intrazonal trips. The results calculated from HBSH, HBSR, HBO.dbf are summed up to get the intrazonal and interzonal trips for trip purpose HBO in the report.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.18. Average Trip Length

Model Step: Distribution

Data Files Required:

- Output\PSNTRIPS_ayy.MAT
- Output\FF.csv
- Output\CONGSKIM

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the person trips DBF file for each trip purpose (HBW.dbf, HBSH.dbf, HBSR.dbf, HBO.dbf, NHB.dbf, TRKTAXI.dbf, and IE.dbf) exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert DBF file for each trip purpose from matrix file Output\PSNTRIPS_ayy.MAT. An example to generate the file IE.dbf is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\IE.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\PSNTRIPS_B00.MAT"
FILLMW MW[1]=mi.1.7
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Create file FF.dbf in project data folder from the output file: Output\FF.csv.
7. Convert the file CONGSKIM.DBF from this matrix file: Output\CONGSKIM_ayy.MAT if is already outdated. This is done by calling the Cube Voyager engine to run the script generated dynamically, as follows:

```
RUN PGM = MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\CONGSKIM.DBF",
MAXFIELDS=3,MO = 1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\CONGSKIM_B00.MAT"
FILLMW MW[1] = mi.1.1
ENDRUN
```

8. Calculate the field values based on the .dbf files converted and output file: Output\PANDA_ayy.DBF, and save the results to the database. The calculations for trip purpose HBW are described as follows.

Column Name	Calculation Formula
<i>TRIPS</i>	Sum (V1 in HBW.dbf)
<i>TRIP HOURS</i>	Sum ((V1 in HBW.dbf) * (V1 in CONGSKIM.dbf))/60
<i>LENGTH EST</i>	<i>TRIP HOURS</i> / <i>TRIPS</i> * 60
<i>LENGTH OBS</i>	Sum (MINUTE_MRK * HBWFF) / Sum (HBWFF)
	Note: MINUTE_MRK and HBWFF are fields in FF.dbf
<i>VARIANCE</i>	Sum (((V1 in CONGSKIM.dbf) - <i>LENGTH EST</i>) ²) / <i>TRIPS</i>
<i>STANDARD DEV</i>	Sqrt (<i>VARIANCE</i>)

The calculations for other trip purposes are similar.

9. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.19. Trip Length Frequency Curve

Model Step: Distribution

Data Files Required:

- Output\PSNTRIPS_ayy.MAT
- Output\FF.csv
- Output\CONGSKIM_ayy.MAT

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the person trips DBF file for each trip purpose (HBW.dbf, HBSH.dbf, HBSR.dbf, HBO.dbf, NHB.dbf, TRKTAXI.dbf, and IE.dbf) exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert DBF file for each trip purpose from matrix file Output\PSNTRIPS_ayy.MAT. An example to generate the file IE.dbf is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\IE.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\PSNTRIPS_B00.MAT"
FILLMW MW[1]=mi.1.7
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Create file FF.dbf in project data folder from the output file: Output\FF.csv.
7. Convert file CONGSKIM.DBF from matrix file: Output\CONGSKIM_ayy.MAT if it has been outdated. This is done by calling the Cube Voyager engine to run the script generated dynamically below:

```
RUN PGM = MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\CONGSKIM.DBF",
MAXFIELDS=3,MO = 1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\CONGSKIM_B00.MAT"
FILLMW MW[1] = mi.1.1
ENDRUN
```

8. Calculate the trip length frequency for each trip purpose based on the .dbf files converted, and save the results to the database. The calculations are done by summing up the trips in each trip length interval.

9. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.20. Trip Length Frequency

Model Step: Distribution

Data Files Required:

- Output\PSNTRIPS_ayy.MAT
- Output\FF.csv
- Output\CONGSKIM_ayy.MAT

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the person trips DBF file for each trip purpose (HBW.dbf, HBSH.dbf, HBSR.dbf, HBO.dbf, NHB.dbf, TRKTAXI.dbf, and IE.dbf) exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert DBF file for each trip purpose from matrix file Output\PSNTRIPS_ayy.MAT. An example to generate the file IE.dbf is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\IE.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\PSNTRIPS_B00.MAT"
FILLMW MW[1]=mi.1.7
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Create file FF.dbf in project data folder from the output file: Output\FF.csv.
7. Convert file CONGSKIM.DBF from matrix file: Output\CONGSKIM_ayy.MAT if it has been outdated. This is done by calling the Cube Voyager engine to run the script generated dynamically below:

```
RUN PGM = MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\CONGSKIM.DBF",
MAXFIELDS=3,MO = 1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\CONGSKIM_B00.MAT"
FILLMW MW[1] = mi.1.1
ENDRUN
```

8. Calculate the trip length frequency for each trip purpose based on the .dbf files converted, and save the results to the database. The calculations are done by summing up the trips in each trip length interval.
9. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.21. Congested Travel Time to One Zone

Model Step: Distribution

Data Files Required:

- Input\ZONEDATA_yya.DBF
- Input\ZONEDATA_yya.shp
- Input\ZONEDATA_yya.shx
- Output\CONGSKIM_ayy.dbf
- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Copy Input\ZONEDATA_yya.shp and Input\ZONEDATA_yya.shx to project data folder.
4. Generate new ZONEDATA_yya.DBF in project data folder with total attractions information from Input\ZONEDATA_yya.DBF and Output\PANDA_ayy.DBF if it has been outdated.
5. Update the congested travel time with the file Output\CONGSKIM_ayy.dbf according to the destination zone selected by the user.
6. Generate the image file for the GIS map if it has been outdated.
7. Load GIS image file(s) into the report template.

4.22. Flow Chart

Model Step: Mode

Data Files Required:

- Applications\05MAT00H.S

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read mode information from script file: Applications\05MAT00H.S. The scripts to define the mode structure of Olympus model are listed below:

```
; Top Level Nest
  SPLIT = TOTAL NCDASR Auto NCTACC Transit,
; Auto Nest
  SPLIT = Auto 1.0 da NCSR HOV, SPLITCOMP=153,
; SR Nest
  SPLIT = HOV 1.0 hov2 1.0 hov3,
; Transit Nest
  SPLIT = Transit 1.0 walk 1.0 pnr 1.0 knr,
; Working matrices
  STARTMW=200
```

4. Generate the mode tree image file (Flowchart.bmp).
5. Load mode tree image file (flowchart.bmp) into the report template.

Notes:

- For Olympus model only, the mode structure information is obtained from the 05MAT00H.S script file. This could be an unreliable process.

4.23. Coefficients

Model Step: Mode

Data Files Required:

- Output\ModeSum_ayy.DAT

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read coefficients for each trip purpose from output file: Output\ModeSum_ayy.DAT, and save the data to the database. The coefficients defined in ModeSum_ayy.DAT of Olympus model are listed below:

```
+++++ MODE XCHOICE SUMMARY +++++  
Olympus 2.0  
Base
```

COEFFICIENTS	HBW
IN VEHICLE TIME	=-0.02000
COMMUTER RAIL TIME	=-0.01600
OUT OF VEHICLE TIME	=-0.04000
NUMBER OF TRANSFERS	=-0.10000
OPERATING COST	=-0.00300

COEFFICIENTS	HBO
IN VEHICLE TIME	=-0.01000
COMMUTER RAIL TIME	=-0.00800
OUT OF VEHICLE TIME	=-0.02000
NUMBER OF TRANSFERS	=-0.05000
OPERATING COST	=-0.00150

COEFFICIENTS	NHB
IN VEHICLE TIME	=-0.02000
COMMUTER RAIL TIME	=-0.01600
OUT OF VEHICLE TIME	=-0.04000
NUMBER OF TRANSFERS	=-0.10000
OPERATING COST	=-0.00150

4. Save all extracted data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.24. Bias Constants

Model Step: Mode

Data Files Required:

- Output\MCCONST.CSV

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read bias constants for each trip purpose from output file: Output\MCCONST.CSV, and save the data to the database.
4. Save all extracted data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.25. Split by Purpose

Model Step: Mode

Data Files Required:

- Output\ModeSum_ayy.DAT

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read and calculate the field values from output file: Output\ModeSum_ayy.DAT, and save the data to the database. The mode choice results of Olympus model in ModeSum_ayy.DAT are shown below:

```
+++++ MODE XCHOICE SUMMARY +++++
Olympus 2.0
Base

HBW - PK MODE XCHOICE RESULTS
HBW TOTAL                = 306,939 1.0000
DRIVE ALONE              = 244,534 0.7967
HOV2                    = 51,506 0.1678
HOV3+                   = 9,641 0.0314
WALK TO TRANSIT         = 1,217 0.0040
PNR TO TRANSIT          = 20 0.0001
KNR TO TRANSIT          = 20 0.0001
...
HBO - OP MODE XCHOICE RESULTS
HBO TOTAL                = 853,660 1.0000
DRIVE ALONE              = 319,787 0.3746
HOV2                    = 358,664 0.4201
HOV3+                   = 173,334 0.2030
WALK TO TRANSIT         = 1,762 0.0021
PNR TO TRANSIT          = 70 0.0001
KNR TO TRANSIT          = 42 0.0000
...
NHB - OP MODE XCHOICE RESULTS
NHB TOTAL                = 474,595 1.0000
DRIVE ALONE              = 138,577 0.2920
HOV2                    = 235,269 0.4957
HOV3+                   = 99,661 0.2100
WALK TO TRANSIT         = 977 0.0021
PNR TO TRANSIT          = 38 0.0001
KNR TO TRANSIT          = 73 0.0002
```

4. Save all extracted data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.26. Split by Zone

Model Step: Mode

Data Files Required:

- Output\ HWYTRIPS_ayy.MAT
- Output\TRNTRIPS_ayy.MAT

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if DRIVEALONE.dbf and CARPOOL.dbf converted from Output\ HWYTRIPS_ayy.MAT exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder first and convert DRIVEALONE.dbf and CARPOOL.dbf from HWYTRIPS_ayy.MAT. An example is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\CARPOOL.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYTRIPS_B00.MAT"
FILLMW MW[1]=mi.1.2
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Check if the following DBF files converted from Output\ TRNTRIPS_ayy.MAT exist. If they do not exist or are outdated, go to next step; otherwise, go to step 9.
 - PKWalkTransit.dbf
 - PKPNRTransit.dbf
 - PKKNRTransit.dbf
 - OPWalkTransit.dbf
 - OPPNRTransit.dbf
 - OPKNRTransit.dbf
7. Create the dynamic Cube Voyager script files in the project data folder and convert the DBF files from TRNTRIPS_ayy.MAT. An example is listed below:

```
RUN PGM=MATRIX
FILEO MATO[1] = "C:\Documents and Settings\Tony\Desktop\test\test data\OPKNRTransit.dbf",
MAXFIELDS=3,MO=1,FORMAT=DBF,PATTERN=IJM:V,FIELDS=4,4,0,6
FILEI MATI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\TRNTRIPS_B00.MAT"
```

FILLMW MW[1]=mi.1.6
 ENDRUN

8. Call the Cube Voyager engine to run the script files generated in the previous step.
9. Calculate the field values with the .dbf files converted above, and save the data to the database:

Column Name	Calculation Formula
<i>ZONE</i>	Field I in each DBF file
<i>AUTO 1</i>	Sum (V1 in DRIVEALONE.dbf) group by I in DRIVEALONE.dbf
<i>AUTO 1 %</i>	$AUTO\ 1 / PERSON\ TRIPS * 100$
<i>AUTO 2+</i>	Sum (V1 in CARPOOL.dbf) group by I in CARPOOL.dbf
<i>AUTO 2+ %</i>	$AUTO\ 2+ / PERSON\ TRIPS * 100$
<i>TRANSIT</i>	Sum (V1 in PKWalkTransit.dbf) + Sum (V1 in PKPNRTransit.dbf) + Sum (V1 in PKKNRTransit.dbf) + Sum (V1 in OPWalkTransit.dbf) + Sum (V1 in OPPNRTransit.dbf) + Sum (V1 in OPKNRTransit.dbf) group by I in each DBF file
<i>TRANSIT %</i>	$TRANSIT / PERSON\ TRIPS * 100$
<i>PERSON TRIPS</i>	$AUTO\ 1 + AUTO\ 2+ + TRANSIT$

10. Save all extracted data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.27. Free Flow Speed By Area Type and Facility Type

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK  
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",  
FORMAT = DBF  
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"  
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include ATYPE, FTYPE, LANES, and SPEED. To get the free flow speed for the highway with certain area type (or facility type) and certain number of lanes, the average speed of all links with the same area type (or facility type) and number of lanes is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.28. Congested Speed by Area Type and Facility Type

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",
FORMAT = DBF
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include ATYPE, FTYPE, LANES, and CGSTD_SPEED. To get the congested speed for the highway with certain area type (or facility type) and certain number of lanes, the average congested speed of all links with the same area type (or facility type) and number of lanes is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.29. Volume-to-Count Ratio by Area Type and Facility Type

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK  
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",  
FORMAT = DBF  
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"  
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include ATYPE, FTYPE, TOTALVOL, and COUNT. To get the volume-to-count ratio for a highway with certain area type and certain facility type, the average volume divided by the average count of all links with the same area type and facility type is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.30. Volume to Capacity Ratio by Screenline

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK  
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",  
FORMAT = DBF  
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"  
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include SCREENLINE, TOTALVOL, and CAPACITY. To get the volume-to-capacity ratio for the highway with certain screenline, the average volume divided by the average capacity of all links with the same screenline is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.31. Bidirectional Capacity

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it is outdated. The field in Hwyload.dbf used for this report is "CAPACITY".
8. Load GIS image file(s) into the report template.

4.32. Bidirectional Volume

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it is outdated. The field in Hwyload.dbf used for this report is "TOT_TWOWAY_N".
8. Load GIS image file(s) into the report template.

4.33. Bidirectional Volume-to-Capacity Ratio

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link and node DBF files from highway network file. An example to generate highway node .dbf file is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) generated above, generate the image file (.jpg file) for the GIS map if it is outdated. The field in Hwyload.dbf used for this report is "VOLCAP".
8. Load GIS image file(s) into the report template.

4.34. Volume-to-Capacity by Area Type and Facility Type

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK  
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",  
FORMAT = DBF  
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"  
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include ATYPE, FTYPE, TOTALVOL, and CAPACITY. To get the volume to capacity ratio for the highway with certain area type and certain facility type, the average volume divided by the average capacity of all links with the same area type and facility type is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.35. VMT by Area Type and Facility Type

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK  
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",  
FORMAT = DBF  
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"  
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include ATYPE, FTYPE, and VMT. To get the VMT for the highway with certain area type and certain facility type, the average VMT of all links with the same area type and facility type is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.36. VHT by Area Type and Facility Type

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK  
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",  
FORMAT = DBF  
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"  
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the results to the database. The fields used include ATYPE, FTYPE, and VHT. To get the VHT for the highway with certain area type and certain facility type, the average VHT of all links with the same area type and facility type is calculated.
7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.37. RMSE

Model Step: Highway Assignment

Data Files Required:

- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) converted from Output\HWYLOAD_yya.net exists. If it does not exist or is outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link DBF files from highway network file. An example to generate highway link DBF file is listed below:

```
RUN PGM = NETWORK
FILEO LINKO = "C:\Olympus\Olympus data\hwyload.DBF",
FORMAT = DBF
FILEI LINKI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Calculate the report values with Hwyload.dbf, and save the data to the database:

Column Name	Calculation Formula
<i>ANODE</i>	A
<i>BNODE</i>	B
<i>COUNT</i>	COUNT
<i>LOAD</i>	TOTALVOL
<i>DIFF</i>	<i>LOAD - COUNT</i>
<i>RATIO</i>	<i>LOAD / COUNT</i>
<i>FT</i>	FTYPE
<i>AT</i>	ATYPE
<i>NUMLANE</i>	LANES
<i>SCR</i>	SCREENLINE

7. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.38. Estimated Patronage

Model Step: Transit Assignment

Data Files Required:

- Input\TROUTE_yya.LIN
- Output\tasum_ayy.PRN
- Output\HWYLOAD_yya.net

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Check if the highway link DBF file (Hwyload.dbf) and node DBF file (Hwyload_node.dbf) converted from Output\HWYLOAD_yya.net exist. If they do not exist or are outdated, go to next step; otherwise, go to step 6.
4. Create the dynamic Cube Voyager script files in the project data folder and convert highway link and node DBF files from highway network file. An example to generate highway node DBF file (Hwyload_node.dbf) is listed below:

```
RUN PGM = NETWORK
FILEO NODEO = "C:\Olympus\Olympus data\hwyload_node.DBF",
FORMAT=DBF
FILEI NODEI[1] = "C:\FSUTMS\general\OlympusTest\base\Output\HWYLOAD_B00.NET"
ENDRUN
```

5. Call the Cube Voyager engine to run the script files generated in the previous step.
6. Generate highway network shape file (Hwyload.shp) and index file (Hwyload.shx) in the project data directory with the link and node.dbf files converted. The fields used are listed below:

Field in Hwyload.dbf	Description
A	the origin node
B	the destination node

Field in Hwyload_node.dbf	Description
N	the node ID
X	the X coordinate value of the node
Y	the Y coordinate value of the node

7. Generate transit route files (Transit.shp, Transit.shx, and Transit.dbf) with Hwyload_node.dbf, Input\TROUTE_yya.LIN, and Output\tasum_ayy.PRN, and save the route files to the project data directory. The name and past nodes of each route can be

retrieved from Input\TROUTE_yya.LIN. The spacial information of the nodes can be extracted from the file Hwyload_node.dbf. Output\tasum_ayy.PRN contains the ridership of each route.

8. With the highway network files (Hwyload.shp, Hwyload.shx, and Hwyload.dbf) and route shape files (Transit.shp, Transit.shx, and Transit.dbf) generated above, generate the image file for the GIS map if it has been outdated.
9. Load GIS image file(s) into the report template.

4.39. Transfer by Mode

Model Step: Transit Assignment

Data Files Required:

- Output\06PTR00J.PRN

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read transfer by mode information from output file: Output\06PTR00J.PRN, and save the data to the database. An example of such information in the Olympus model is shown below:

```
REPORT XFERSUM=MODE UserClass=1
MODE 21 31 1 12
-----
21 67.36 35.89 231.61 1.99
31 23.98 422.96 985.64 0.55
1 244.96 972.29 -- --
12 0.55 1.99 -- --
```

4. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.40. Station Load

Model Step: Transit Assignment

Data Files Required:

- Output\tasroute_B00.PRN

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read station load information from output file: Output\tasroute_B00.PRN.
4. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

4.41. Riders by Mode and Route

Model Step: Transit Assignment

Data Files Required:

- Output\tasum_ayy.PRN

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read riders information from output file: Output\tasum_ayy.PRN.
4. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

Notes:

- There are no data for the column "OBS. RIDERS".
- The data for other columns are in the file Output\tasum_ayy.PRN, but cannot be retrieved because of its random data format.

4.42. Peak Period Vehicle Requirement

Model Step: Transit Assignment

Data Files Required:

- Output\tasum_ayy.PRN

Data Generation Procedure:

1. Check if the data required for this report have been retrieved and are up-to-date. If not, continue to the next step; otherwise, go to the last step.
2. Check if the required data files exist. If yes, continue; otherwise, notify the user of the missing files.
3. Read peak period vehicle requirement information from output file: Output\tasum_ayy.PRN.
4. Save all extracted and calculated data to a temporary placeholder database, retrieve data from the placeholder database, and import retrieved data into the report template.

Notes:

- Missing columns "STANDING TIME" and "REQ. VEH" due to the lack of data.

5. SYSTEM ENHANCEMENTS

The system described in this report has provided good capabilities for managing, generating, and displaying the set of standard reports developed by Gannett Fleming, etc. Currently FSUTMS, NERPM, SERPM, and Statewide models have been implemented, and all other FSUTMS urban and regional models will be added and supported soon.

While the existing system already generates a total of 42 standard reports, many more useful reports can and should be defined and implemented. A survey of expert FSUTMS users with good knowledge of practical needs for FSUTMS reports can be conducted to identify these reports. Based on the survey results, additional reports can be identified, designed, and implemented.