Updates and Enhancements to the LOSPLAN User Interface and Computational Procedures

Final Report
October 2003

Contract #: BC 354 RPWO 65
UF Project #: 4910-4504-893-12

Prepared for:
The Florida Department of Transportation

Prepared by:
Transportation Research Center
Department of Civil and Coastal Engineering
University of Florida

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Updates and Enhancements to the LOSPLAN User Interface and Computational Procedures		5. Report Date October 2003
		6. Performing Organization Code UF-TRC
7. Author(s) Scott S. Washburn		8. Performing Organization Report No. TRC-FDOT-893-2003
9. Performing Organization Name and Address University of Florida Dept. of Civil and Coastal Engineering 365 Weil Hall / P.O. Box 116580 Gainesville, FL 32611-6580		10. Work Unit No. (TRAIS)
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address Florida Department of Transportation 605 Suwannee St. MS 30 Tallahassee, Florida 32399 (850) 414 - 4615		13. Type of Report and Period Covered Final Report
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

This report summarizes the results of several tasks performed to improve and enhance the Florida Department of Transportation's (FDOT) LOSPLAN suite of software programs. Initial development of the LOSPLAN programs was conducted under a previous contract (BC-354-38). Specific accomplishments from this project include:

- Revisions, refinements, and debugging of the user interfaces and calculations,
- Revisions and updating of electronic help resources,
- Refinements of the program code structures to reduce computer resource consumption,
- Development of a batch processing utility,
- Integration of a remote automation capability,
- Demonstration of the potential for database interfacing with the LOSPLAN programs,
- Revision of the XML input routine,
- Development of dynamic link library (DLL) implementations of the calculations code, and
- Development of a preliminary version of CORPLAN.

17. Key Words	18. Distribution Statement
Level of service planning software, level of service planning methodologies, level of service	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, VA, 22161

19 Security Classif. (of this report)	20.Security Classif. (of this page)	21.No. of Pages	22 Price
Unclassified	Unclassified	93	

Form DOTF 1700.7 (8-72) Reproduction of completed form authorized

Table of Contents

Introduction 1
Overview of ARTPLAN, HIGHPLAN, and FREEPLAN Programs 3
Program Optimization 4
Batch Processing Utility and Remote Automation Feature 5
    Batch Processing Utility 5
    Remote Automation 5
Database Interfacing 6
MSXML Parser 8
Calculations Dynamic Link Library (DLL) 9
CORPLAN (Corridor Planning) 10
Appendices 21
    Appendix A: LOSPLAN Component Program User Interface Screens 22
    Appendix B: LOSPLAN Component Program Revision Histories 35
    Appendix C: Batch Processing Instructions 44
    Appendix D: Database Interface Example 52
    Appendix E: DLL Documentation 76

INTRODUCTION

This report summarizes the results of several tasks performed to improve and enhance the Florida Department of Transportation's (FDOT) LOSPLAN suite of software programs. Collectively, the FDOT planning level of service software is grouped under the name LOSPLAN, with the three primary component programs being ARTPLAN, FREEPLAN, and HIGHPLAN, each of which implements the HCM analysis procedures for their respective facilities (arterials, freeways and highways). The LOSPLAN programs complement the 2002 Q/LOS Handbook and are integral to the quality/level of service program for the FDOT.

Initial development of the LOSPLAN programs was performed under Task 1 (Updated Procedures for Florida Department of Transportation Planning Level Software Tools) of contract#: BC-354-38 (Facility Performance Model Enhancements for Multimodal Systems Planning).

With initial program design and development performed under this previous contract, this project was focused on improvements and enhancements to the LOSPLAN component programs, ARTPLAN, FREEPLAN, and HIGHPLAN, as well as the development of a new corridor planning analysis program. Specifically, the following tasks were addressed under this contract:

- Refine the user interface of the existing LOSPLAN programs,
- Refine and test the computational methodologies of the existing LOSPLAN programs,
- Extend the computational capabilities of the existing LOSPLAN programs, and
- Develop a new LOSPLAN component program that would integrate the analysis capabilities of the existing LOSPLAN programs into a corridor analysis program.

The specific accomplishments of this project include the following:

- Revisions, refinements, and debugging of the user interfaces
- Revisions, refinements, and debugging of the calculations
- Revisions and updating of electronic help resources
- Refinements of the program code structures to reduce computer resource consumption
- Development of a batch processing utility
- Integration of a remote automation capability
- Demonstration of the potential for database interfacing with the LOSPLAN programs
- Revision of the XML input routine
- Development of dynamic link library (DLL) implementations of the calculations code
- Development of a preliminary version of CORPLAN

The remaining sections of this report give specific details about each of the project accomplishments listed above.

The project was a combined effort between the Transportation Research Center (TRC) at the University of Florida and Polytechnic University of New York. Specific task responsibility by principal researcher was organized as follows:

- ARTPLAN user interface issues (Scott Washburn, UF-TRC)
- ARTPLAN calculations issues (Elena Prassas, Polytechnic)
- HIGHPLAN user interface and calculations issues (Scott Washburn, Kenneth Courage, UF-TRC)
- ARTPLAN and HIGHPLAN electronic documentation (Scott Washburn, UF-TRC)
- FREEPLAN user interface and calculations issues (Elena Prassas, Polytechnic)
- FREEPLAN electronic documentation (Elena Prassas, Polytechnic)
- Batch Processing Utility (Scott Washburn, Kenneth Courage, UF-TRC)
- Remote Automation Capabilities (Scott Washburn, Kenneth Courage, UF-TRC)
- Database Interface Sample (Scott Washburn, UF-TRC)
- DLL development (Scott Washburn, UF-TRC)
- CORPLAN development (Scott Washburn, UF-TRC)

OVERVIEW OF ARTPLAN, HIGHPLAN, AND FREEPLAN PROGRAMS

Details of the initial versions of the LOSPLAN programs were covered in the BC-354-38 final project report, which can be obtained from the following website:

www11.myflorida.com/research-center/Completed_Proj/Summary_PL/FDOT_BC354_38rpt.pdf

Additional details of the software, including an overview of the computational methodologies, can be found in the FDOT Quality/Level of Service Handbook.

www11.myflorida.com/planning/systems/sm/los/los_sw2.htm

User interface revisions/alterations suggested by beta testers were discussed between FDOT staff and TRC and/or Polytechnic researchers. Some of the suggestions were subsequently incorporated. Several revisions to the LOSPLAN user interfaces were also requested directly by FDOT staff. All but the ones requiring significant alteration of the code were incorporated. Additionally, any identified software bugs during the testing process were corrected.

The final user interface screens are shown in Appendix A.

The revision histories for the ARTPLAN and HIGHPLAN versions released, for testing and official purposes, are included in Appendix B. A detailed revision history was not maintained for FREEPLAN.

The latest versions (June 2003) of the ARTPLAN, HIGHPLAN, and FREEPLAN programs are included on the enclosed CD. Also included are the latest electronic help files (which were converted from their previous Windows format to the newer HTML format).

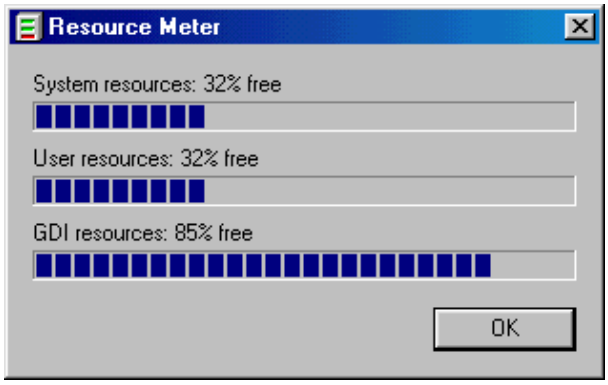
PROGRAM OPTIMIZATION

The initial versions of the LOSPLAN programs were developed in an incremental manner using a continuous input/feedback loop between TRC researchers and FDOT staff during the preliminary design stage. Consequently, the program software architecture was not constructed with optimization as a primary objective. Once the functional requirements and user interface (UI) design became fairly stable, the underlying software code and program architecture could be optimized. Optimization of resource utilization is a very important consideration for the final release of the software, as most users of the software will not have new computers with very fast processors and very large amounts of Random Access Memory (RAM).

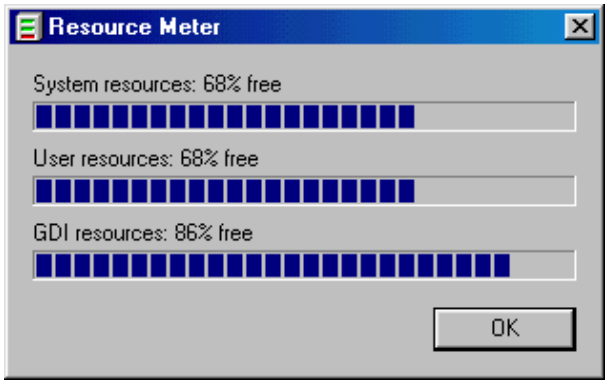
This task of optimization, for ARTPLAN, was performed over an approximately three-month period and resulted in a substantial reduction in resource utilization. The two figures below show the resource utilization for an example "average" computer with the latest non-optimized version, and the resource utilization with this same version after the optimization. As seen in these figures, the resource utilization was improved substantially through the optimization of the software code and program architecture.

The ARTPLAN computational engine was reviewed for potential memory usage efficiency improvements. Variable type definitions (e.g., integer vs. single) were reviewed and changed where possible to minimize memory allocation.

Note: Only ARTPLAN and the operating system running



Before optimization



After optimization

An optimization process analogous to that performed for ARTPLAN was also performed for HIGHPLAN and FREEPLAN. Resource utilization was not as critical an issue for HIGHPLAN due to its lesser complexity relative to ARTPLAN. Nonetheless, it was important to optimize HIGHPLAN to the extent possible so as to minimize its impact on system resources, particularly when other programs may be running at the same time.

BATCH PROCESSING UTILITY AND REMOTE AUTOMATION FEATURE

Batch Processing Utility

The batch processing utility was developed to allow multiple files to be processed automatically. This processing includes loading an input file, performing the level of service calculations for those inputs, and then saving those results back into the input file.

The instructions for using the batch processing utility are included in Appendix C.

Remote Automation

ARTPLAN, HIGHPLAN, and FREEPLAN have been structured to be executable with command line arguments that will bypass the user interface and execute the computations directly using a data file name supplied on the command line.

The remote automation is executed through command-line arguments. Command line instructions and arguments are typically specified through either the DOS command prompt or a shell-type command from within another program.

When a program is executed with no command line arguments, it opens in the current fully interactive mode with the default values displayed for all data items. When a file name is present as the first command line argument, the data from that file shall be loaded. If the command line argument "/R" is present with a data file name, the following action shall occur:

- Computations shall be performed on the data that were loaded from the specified file, ignoring any performance measures that were included in the file.
- The complete data file including performance measures shall be overwritten, using the originally specified file name.
- The program shall terminate.
- The screen display prior to the program execution shall not be disturbed.

If a file name is specified but the "/R" argument is not present on the command line, the data values shall be displayed in place of the default data and the normal interactive operation shall commence.

The batch processing utility uses this feature to interact with the LOSPLAN programs.

DATABASE INTERFACING

In previous LOS Task Team meetings, the issue of being able to interface existing FDOT district databases (of roadway, traffic, and control characteristics) with the computational processes of the LOSPLAN programs was raised.

For a district that maintains an inventory-type database that contains all of the necessary inputs to the LOSPLAN programs, it would be preferred to be able to export that data automatically into a file format that can be opened by the LOSPLAN programs. The alternative is to manually enter the data directly into the user interface of an LOSPLAN program. This is undesirable because it is inefficient, as the data have already been entered into an electronic format (i.e., the database), and entering the same data multiple times increases the opportunities for making a data entry mistake.

To address the issue, the TRC researchers developed a small demonstration database and conversion utility to take the contents of a database and put them in an LOSPLAN file compatible format. A sample database was designed to demonstrate how the LOSPLAN programs can interface with a database system of roadway, traffic, and control variables. An example Microsoft Access database was developed for this purpose. Note, however, that any database software that is XML compatible can be utilized. A brief description of XML is given in the next section.

The sample database contains typical inputs for an ARTPLAN analysis. These are in the form of tables. This is where a database user would typically enter their data. Queries were developed to extract the desired input data from the appropriate tables. These queries were then exported in XML format (while this process is done manually for the purposes of the demonstration, an Access macro can easily be written to automate this task). A conversion utility was written to organize the exported Access XML files into one XML file that is compatible for input into ARTPLAN. This file can then be loaded into ARTPLAN like any other file that was originally saved from ARTPLAN.

A detailed step-by-step tutorial to the demonstration is given in Appendix D. The sample database and conversion utility is included on the enclosed CD.

The greatest potential utility for this feature comes when combining it with the remote automation capabilities of the LOSPLAN programs (as described in the previous section). This would allow a database user to never have to directly interact with an LOSPLAN program. The user simply initiates the process that extracts their input data from the database and puts it into an LOSPLAN compatible file format. This process would then initiate the remote automation capability whereby the file created from the database input data is automatically loaded into the appropriate LOSPLAN program, the results calculated and then saved back into the data file. Although not developed as part of this demonstration, the database-LOSPLAN interface utility can also incorporate a component to handle the reverse direction of data flow. Once the LOSPLAN program has calculated and saved the results back into the data file, the utility could automatically extract the results from the file and place them into a specified database location.

Note that this is a fairly simple example and demonstration. Access contains a rich set of features for working with XML. In particular, XML schemas and stylesheets can be accommodated within Access, thus providing considerable control over the way XML data is organized, exported, and displayed.

MSXML PARSER

The traffic modeling markup language (TMML) was chosen as the format for writing saved LOSPLAN files. TMML is based on XML (Extensible Markup Language). XML describes the general requirements (formatting, syntax, etc.) that files must conform to be XML compliant. More information on XML can be obtained from the World Wide Web Consortium (W3C) website at:

http://www.w3.org/XML/

TMML is a traffic modeling vocabulary implementation of XML. More information on TMML can be obtained at:

http://www.ce.ufl.edu/trc/research/tmml.htm

The custom routine that was written to read TMML-formatted files in the original LOSPLAN programs was replaced with a commercial utility for performing this task (ARTPLAN and HIGHPLAN only). This utility is produced by Microsoft, and is called MSXML (version 4) and is designed specifically for the purpose of reading files written in an XML-compliant format. This utility is distributed freely. This change was made because Microsoft's utility is more robust to the full features and specifications that are possible with XML formatted files. This capability will allow the LOSPLAN programs to read files generated from third-party programs as long as their format is XML-compliant and the content contains the appropriate subset of the TMML vocabulary for the intended LOSPLAN program (i.e., arterial data for ARTPLAN, highway data for HIGHPLAN, freeway data for FREEPLAN).

More information on the MSXML utility can be found on Microsoft's web site at the following website:

http://www.microsoft.com/downloads/details.aspx?FamilyID=
3144b72b-b4f2-46da-b4b6-c5d7485f2b42&DisplayLang=en

Example TMML files for each of the LOSPLAN programs are included on the enclosed CD.

CALCULATIONS DYNAMIC LINK LIBRARY (DLL)

The calculations for each of the LOSPLAN programs were implemented in dynamic link library (DLL) files. The result is an implementation of the calculations that is completely separate from the core program code, in particular the user interface and file input/output code. The advantage to this approach is that the calculations can be called from a user interface (i.e., data input program) developed by another party. This capability was developed mostly in response to the desire of several consultants within the state to develop their own user interface for the ARTPLAN calculations methodology. The data validation routines have been duplicated within the DLLs and are applied when a DLL is first called by a user interface application. If there is any issue with the integrity of the input data, an error message will be displayed, the calculations process will terminate, and program control will be returned to the DLL calling application.

The supporting documentation on how to implement and interface with the DLLs for each of the LOSPLAN programs is included in Appendix E. The DLL files are included on the enclosed CD. These files and supporting documentation will be available on the FDOT's Level of Service website (www11.myflorida.com/planning/systems/sm/los/los_sw2.htm).

CORPLAN (CORRIDOR PLANNING)

The tasks in this group relate to the integration of the three existing component programs into a comprehensive corridor analysis tool, which was one of the principal project objectives.

A corridor is considered to contain two or more parallel routes that carry traffic in the same general direction and function as a system. Each route consists of either an arterial, a freeway, or a highway facility. CORPLAN allows the performance of two or more parallel facilities (routes) within a corridor to be compared. CORPLAN is currently constrained to the comparison of like facilities within the same corridor (i.e., arterial vs. arterial, freeway vs. freeway, highway vs. highway). This constraint was the result of LOS Task Team feedback in which it was expressed by a strong majority that they did not want to allow comparisons across differing facility types (e.g., freeway vs. arterial).

The remainder of this section gives an overview of the functionality of CORPLAN.

Figure 1 shows the corridor properties screen. This is where the basic information about the corridor being studied is entered. General file information is also entered on this screen.

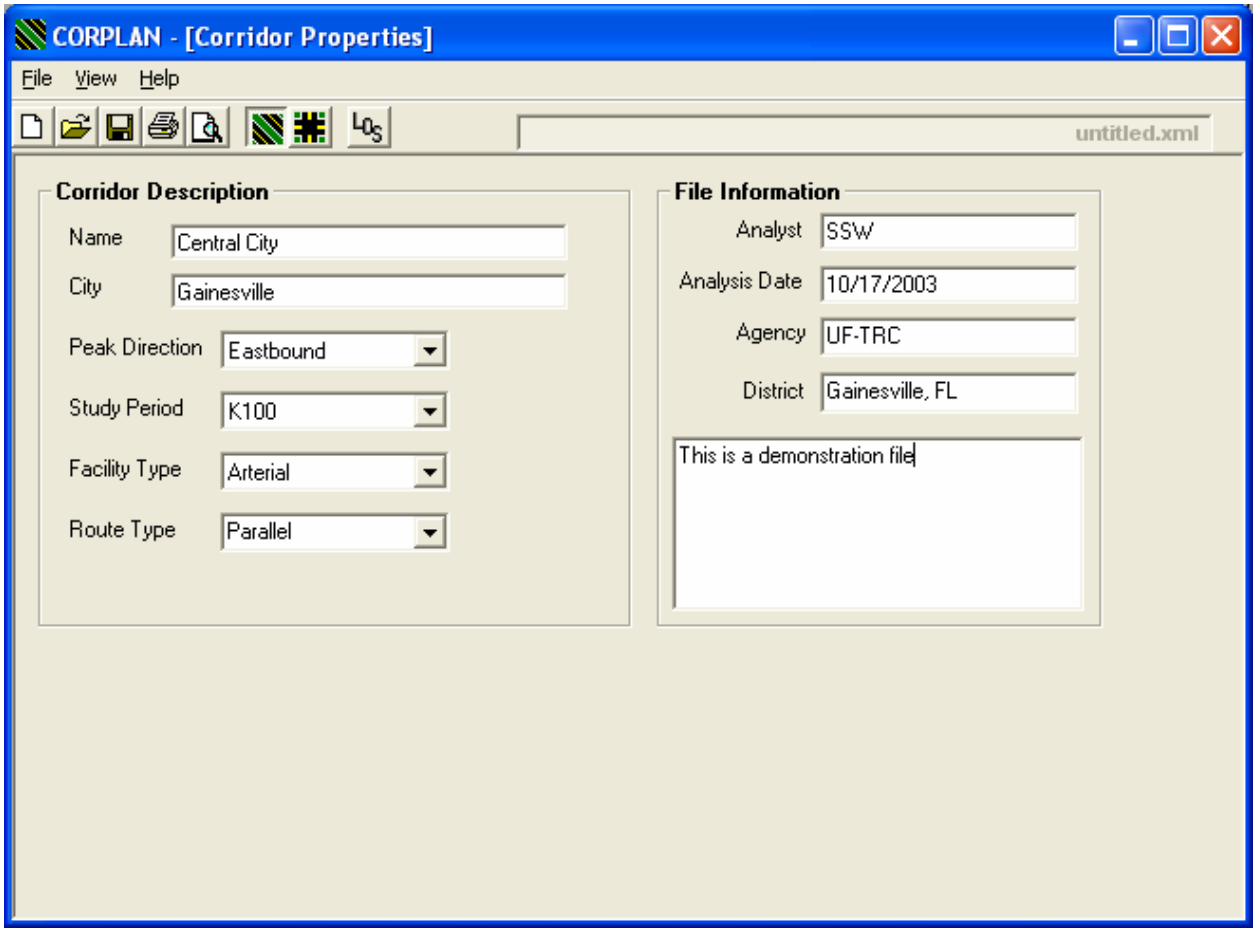


Figure 1. Corridor Properties Screen

Under the 'Facility Type' input field (see Figure 2), an arterial, highway, or freeway facility can be selected. CORPLAN currently constrains a corridor analysis to the same type of facility. That is, multiple arterials within a corridor can be compared, or multiple highways, but arterials cannot be compared to highways within the same corridor, etc. Additionally, another current constraint is that all routes must be parallel. The remaining data input fields are self-explanatory.

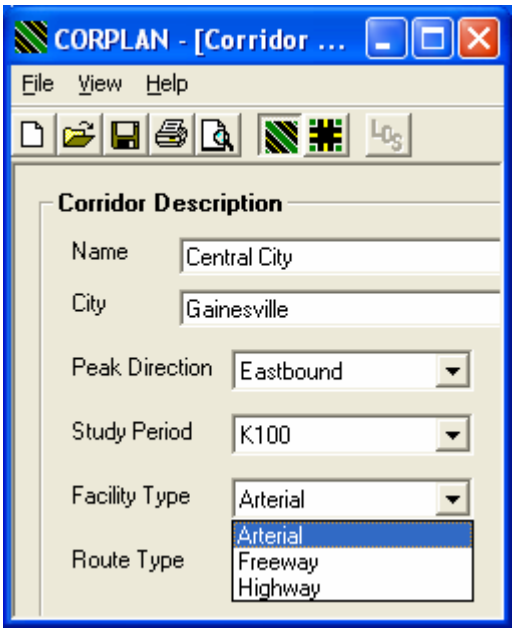


Figure 2. Facility Type Input Field Options

The Facility Data screen (accessed by the intersection toolbar button or from the 'View' menu), Figure 3, consists of two screens, Facility Properties and Facilities Report. From the Facility Data screen, the user can select the facility data files to be included in the corridor analysis. Up to six parallel facilities (again, of the same type) can be specified for a single corridor. These files are specified by pressing the 'Open File' button. The open data file dialog box (Figure 4) will prompt you for filenames specific to the facility type indicated on the Corridor Properties screen, as denoted by the two letter prefix in front of the filename (i.e., 'AP', 'HP', or 'FP'). The file will be entered into the currently active row, indicated by the adjacent radio button selection.

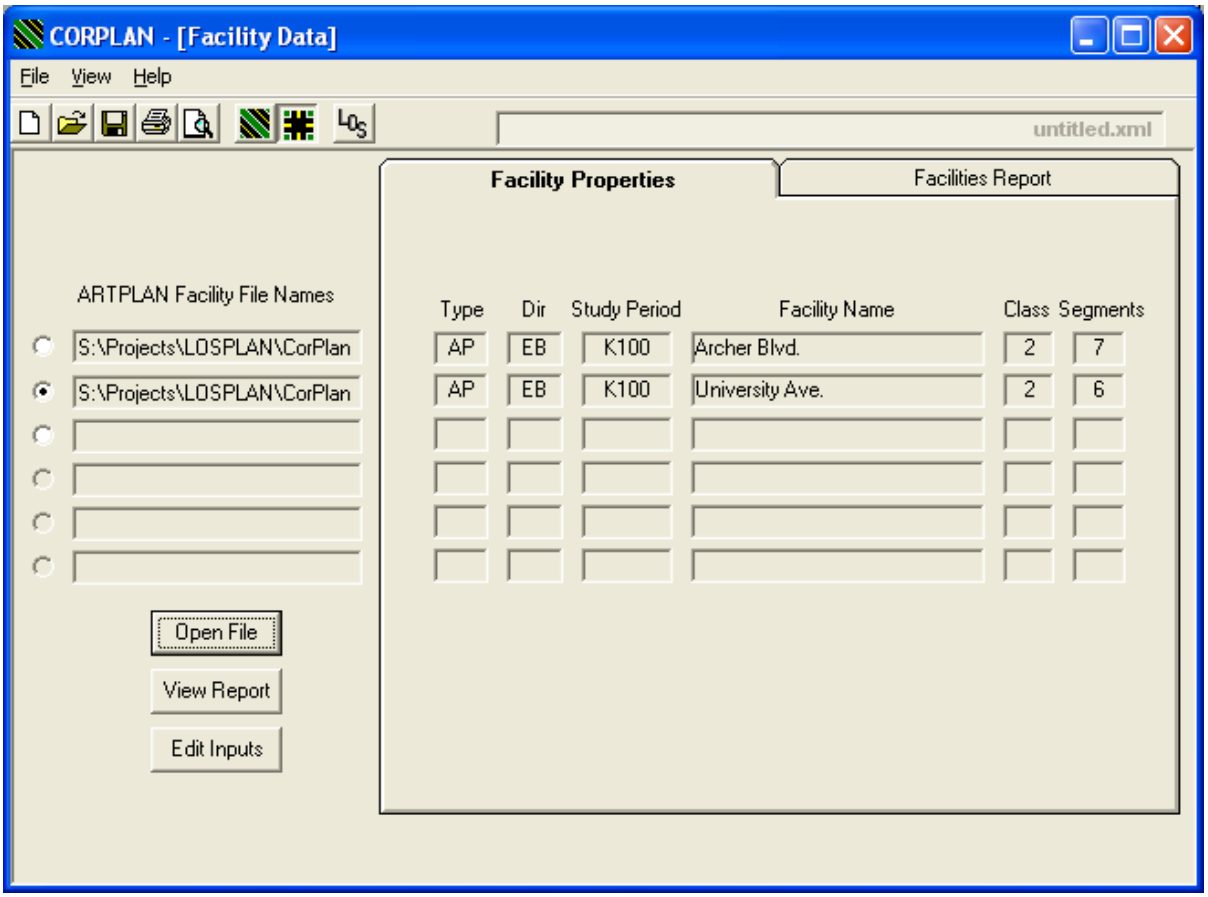


Figure 3. Facility Data (Facility Properties)

Once a facility file is opened, basic summary information for that facility analysis will be displayed. Note that if the type, direction, or study period from the facility file do not match with the values specified on the Corridor Properties screen, an error message will be generated and the file loading process will be aborted.

Pressing the 'Edit Inputs' button launches the applicable LOSPLAN program (displaying the facility data screen) and loads the file specified in the filename box that has the radio button next to it selected. This allows a user to make input edits, rerun the level of service results, and resave the file. Once the LOSPLAN program is closed, the file will be automatically reloaded into CORPLAN, displaying any new results.

Pressing the 'View Report' button performs a similar function as above, but the print preview screen is displayed (showing the formatted report), with inputs and level of service results. While the user can also edit the file in this mode, upon closing the LOSPLAN program, the file will not be automatically reloaded into CORPLAN.

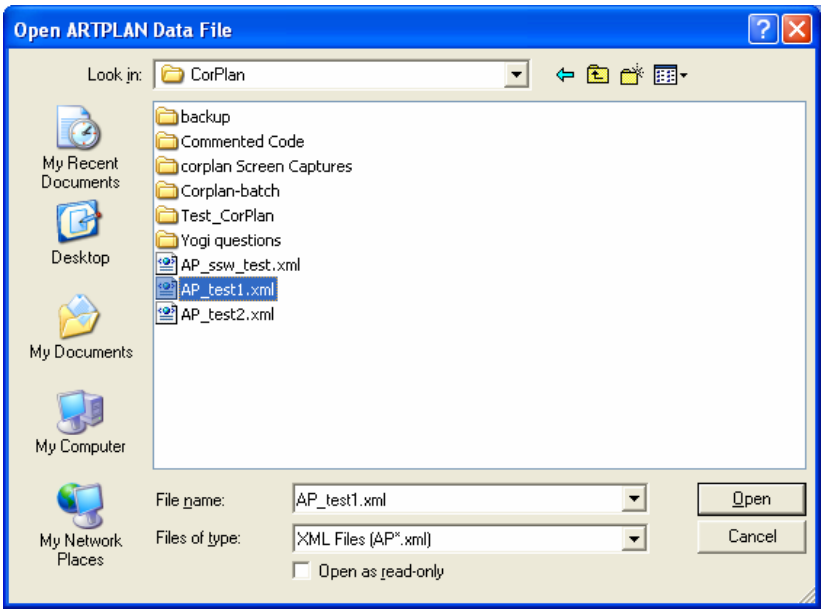


Figure 4. Open Data File Dialog Box

The Facilities Report screen (Figure 5) displays summary level of service results.

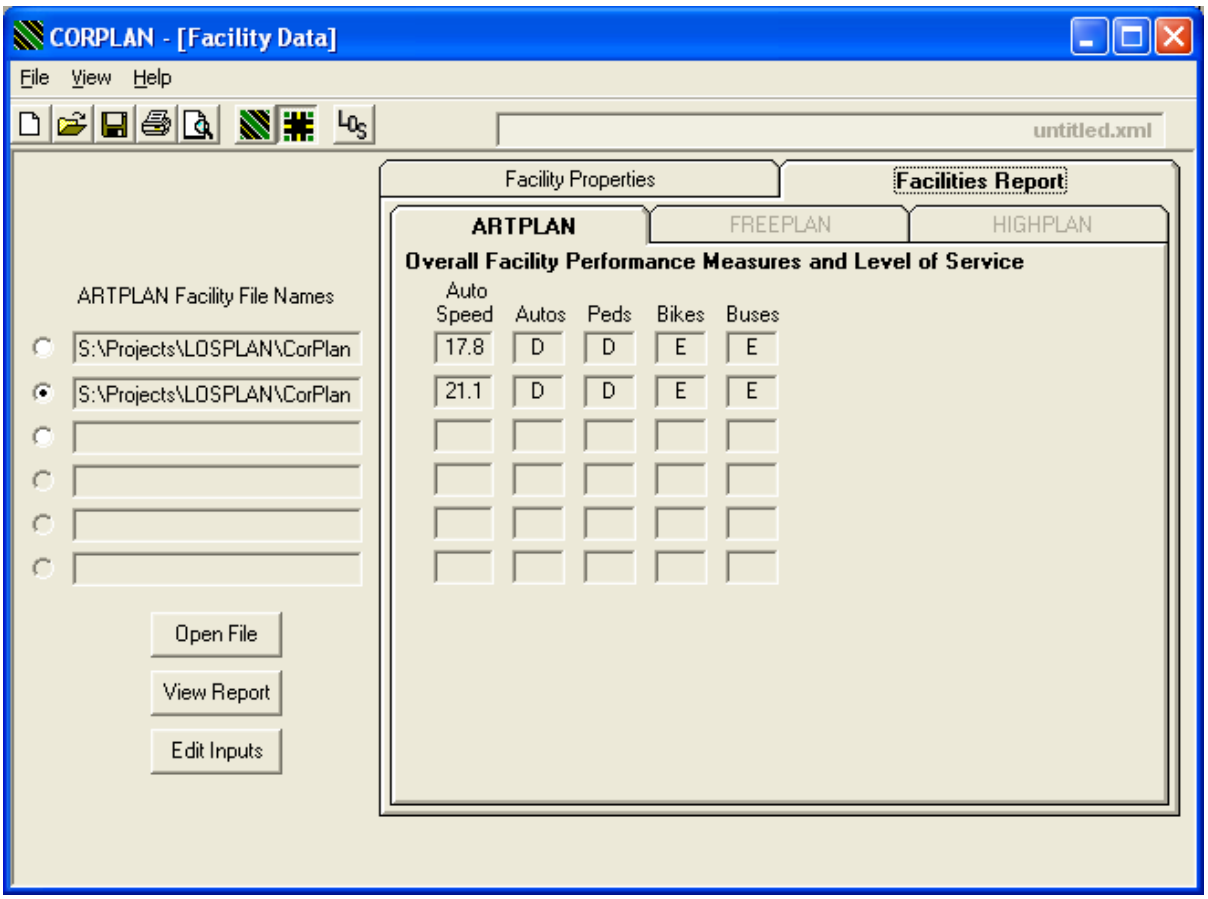


Figure 5. Facility Data (Facilities Report)

Pressing the 'LOS' button the toolbar, or selecting 'LOS Hotspots' from the 'View' menu, displays a summary of level of service results for individual segments meeting the specified LOS selection criteria (Figure 6). Note that this button or menu item will not become active until at least valid facility file has been loaded into the Facility Data screen.

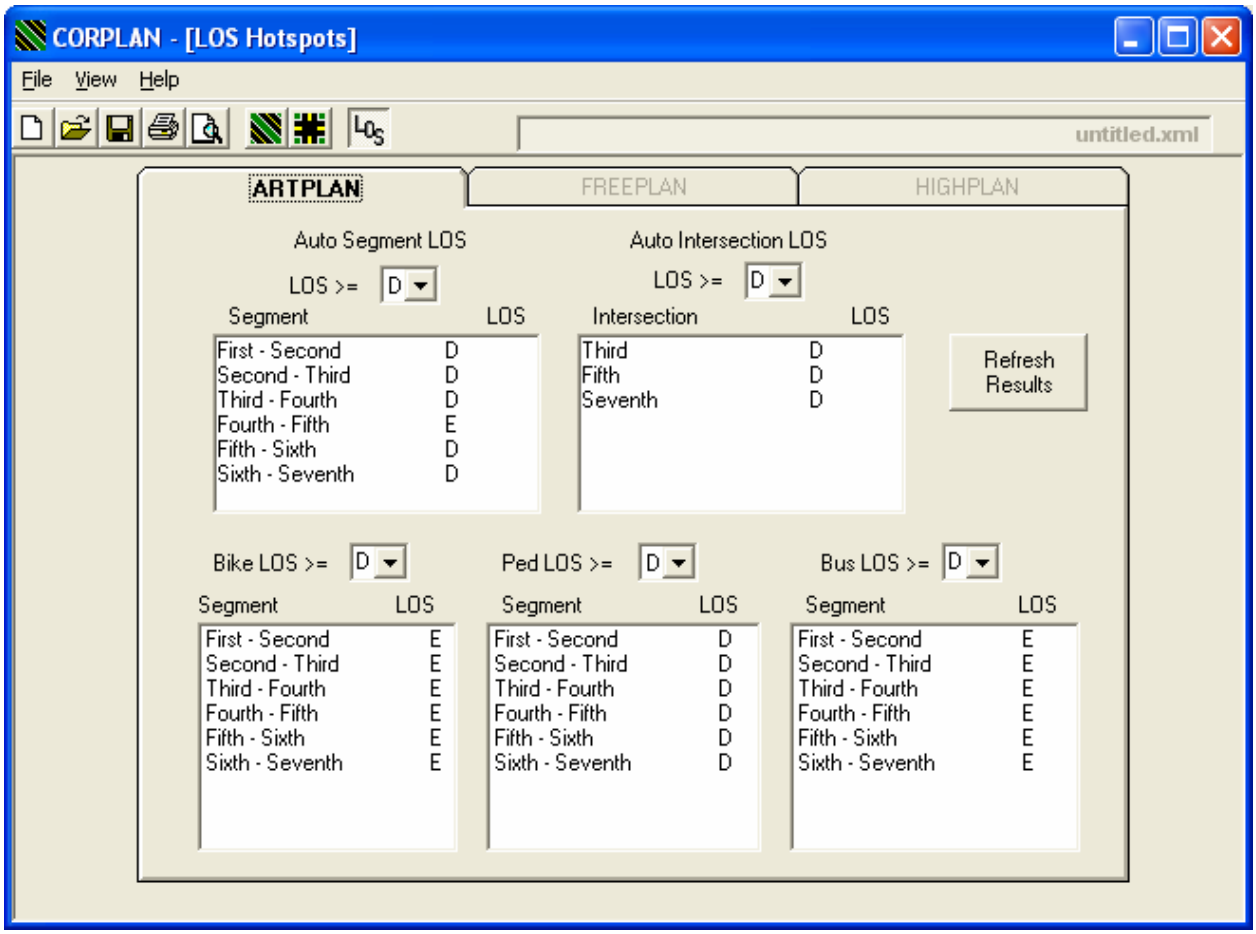


Figure 6. LOS Hotspots Screen

The drop-down list boxes above each results display list box can be used to change the LOS criterion for displaying results (see Figure 7 and Figure 8). Tabbing out from this list box or pressing the 'Refresh Results' screen will automatically update the display according to the selected criteria for all results display boxes.

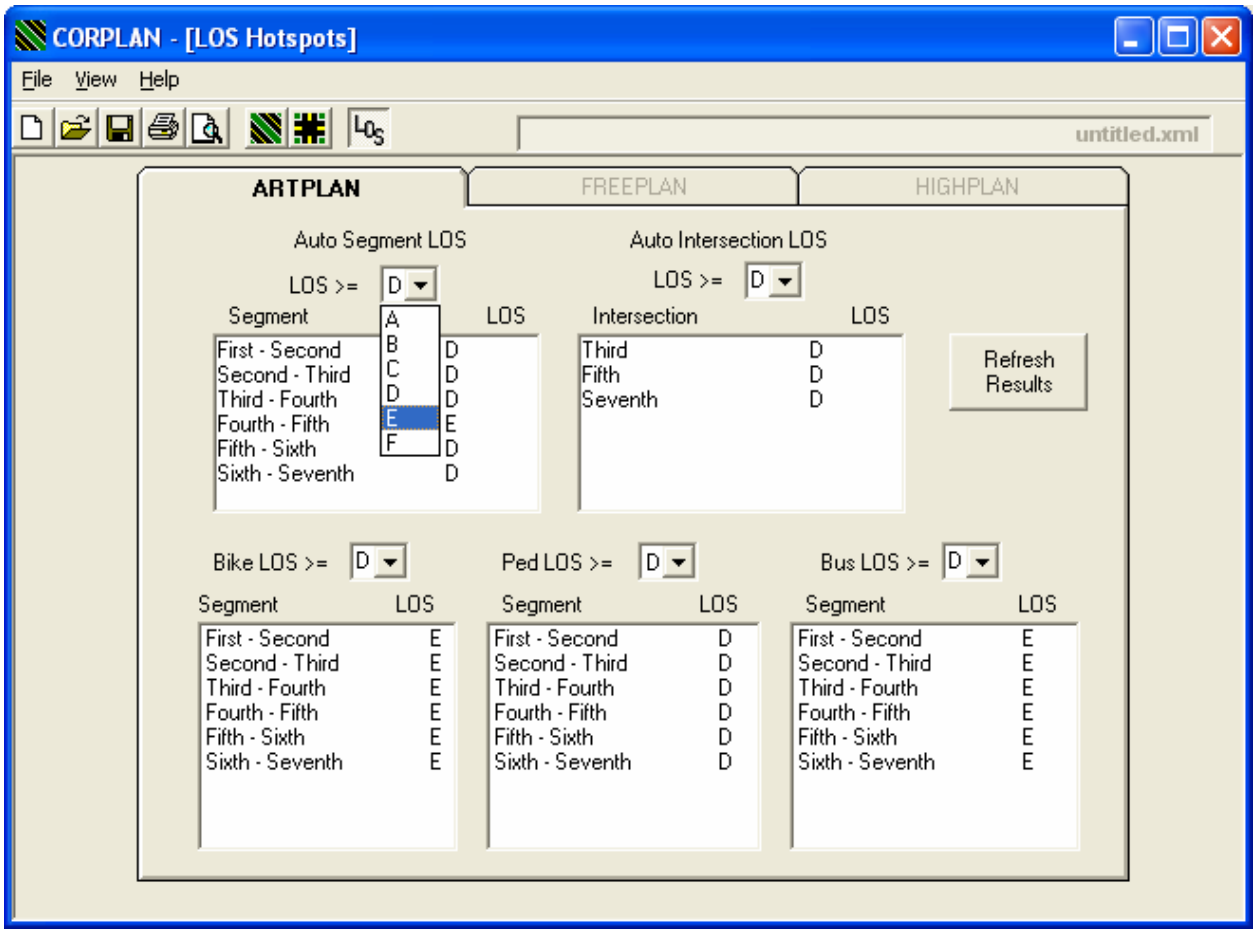


Figure 7. Changing the LOS Criterion for Displaying Auto Segment Results

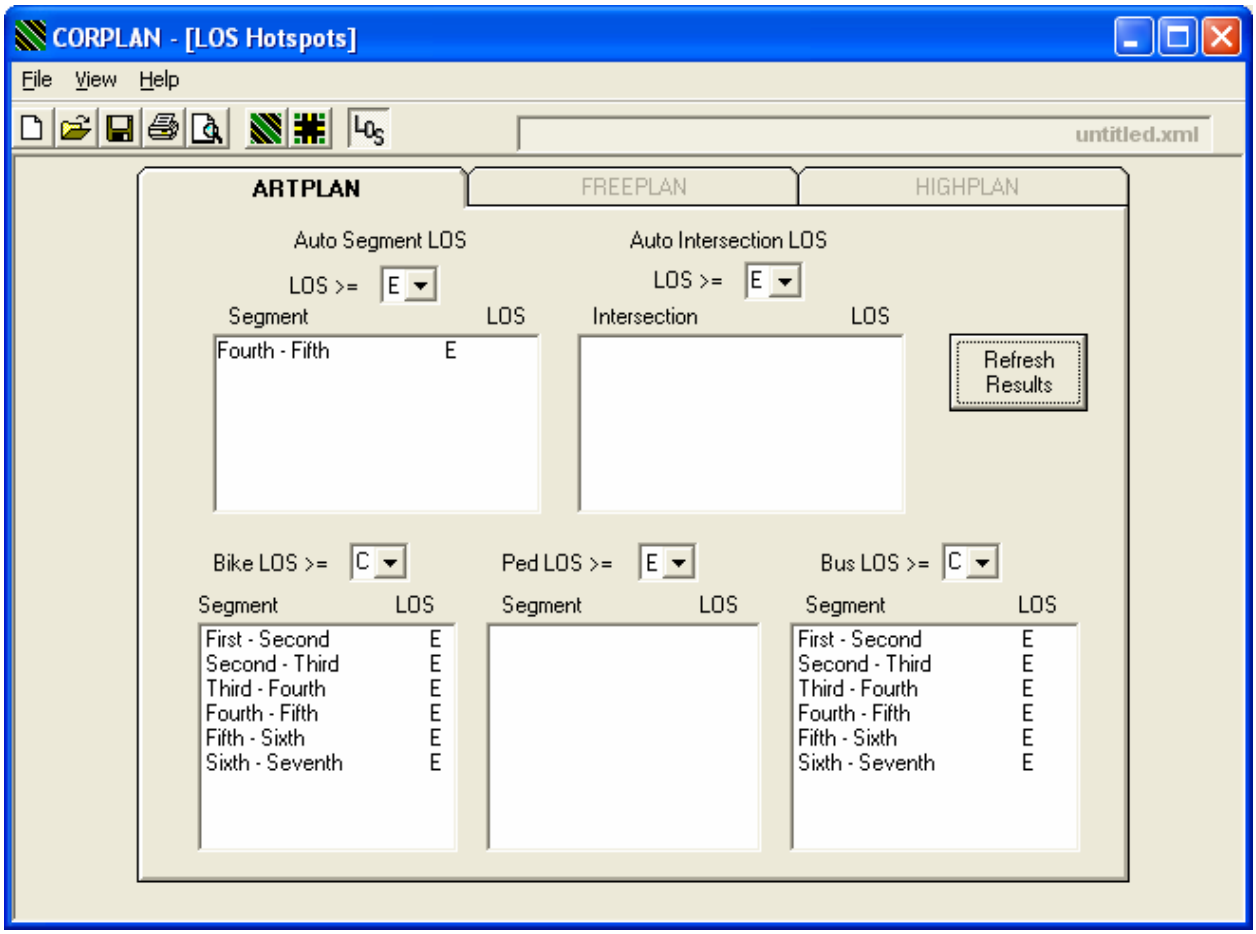


Figure 8. Updated Results Display After Changing All LOS Criteria

Figures 9 through 12 show some additional screen shots for HIGHPLAN and FREEPLAN loaded facility files.

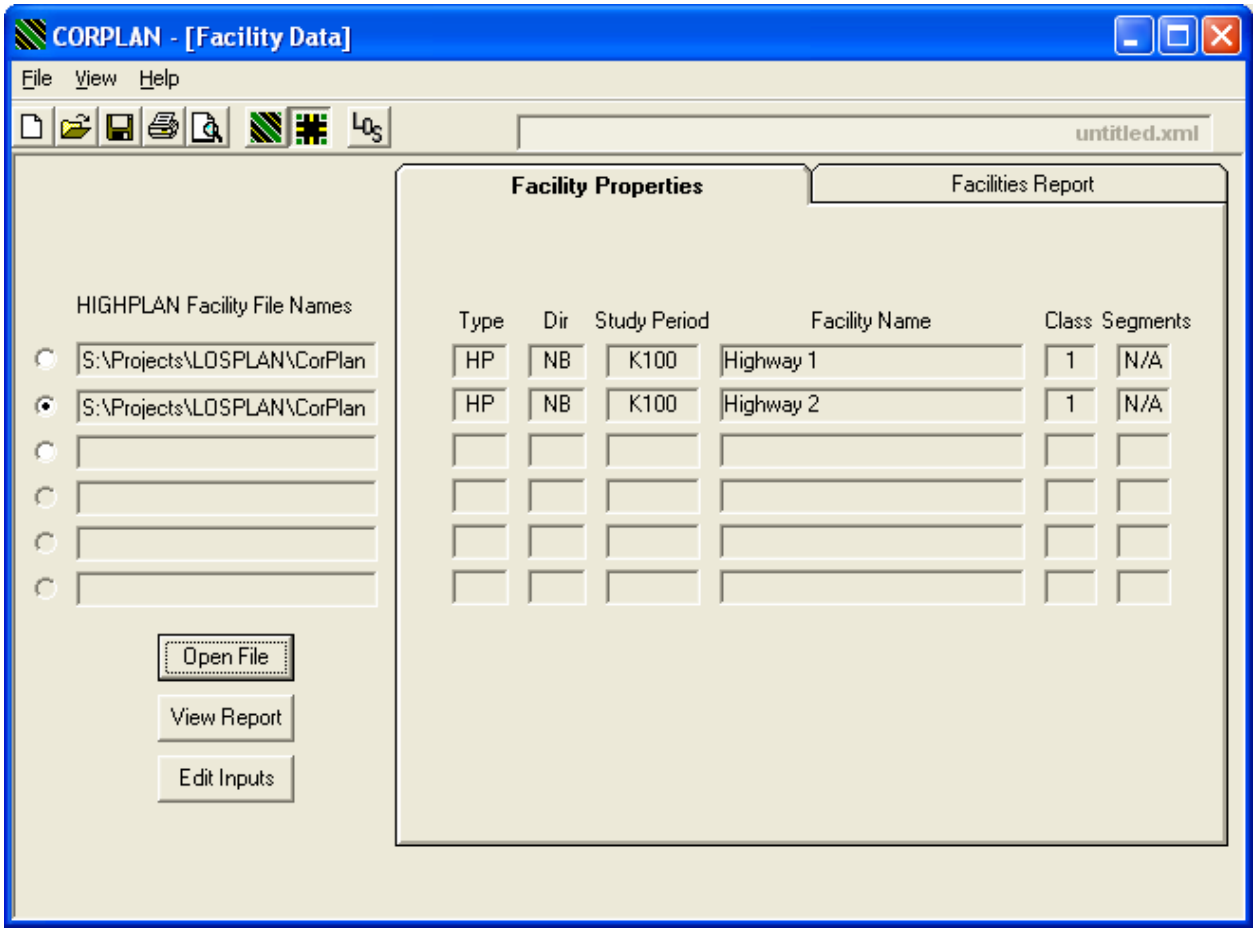


Figure 9. HIGHPLAN Files Loaded Into Facility Properties Screen

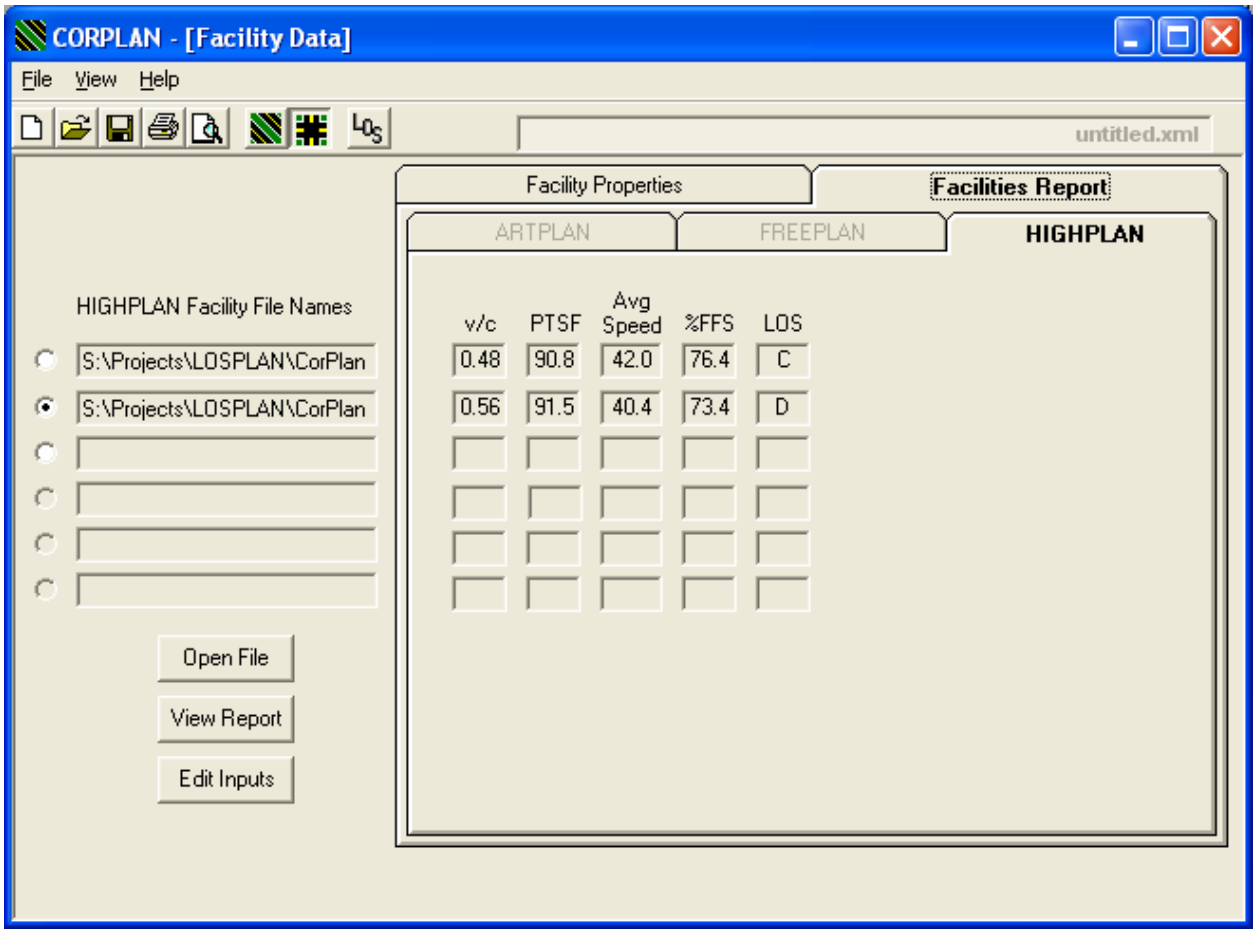


Figure 10. Facilities Report Screen for Loaded HIGHPLAN Files

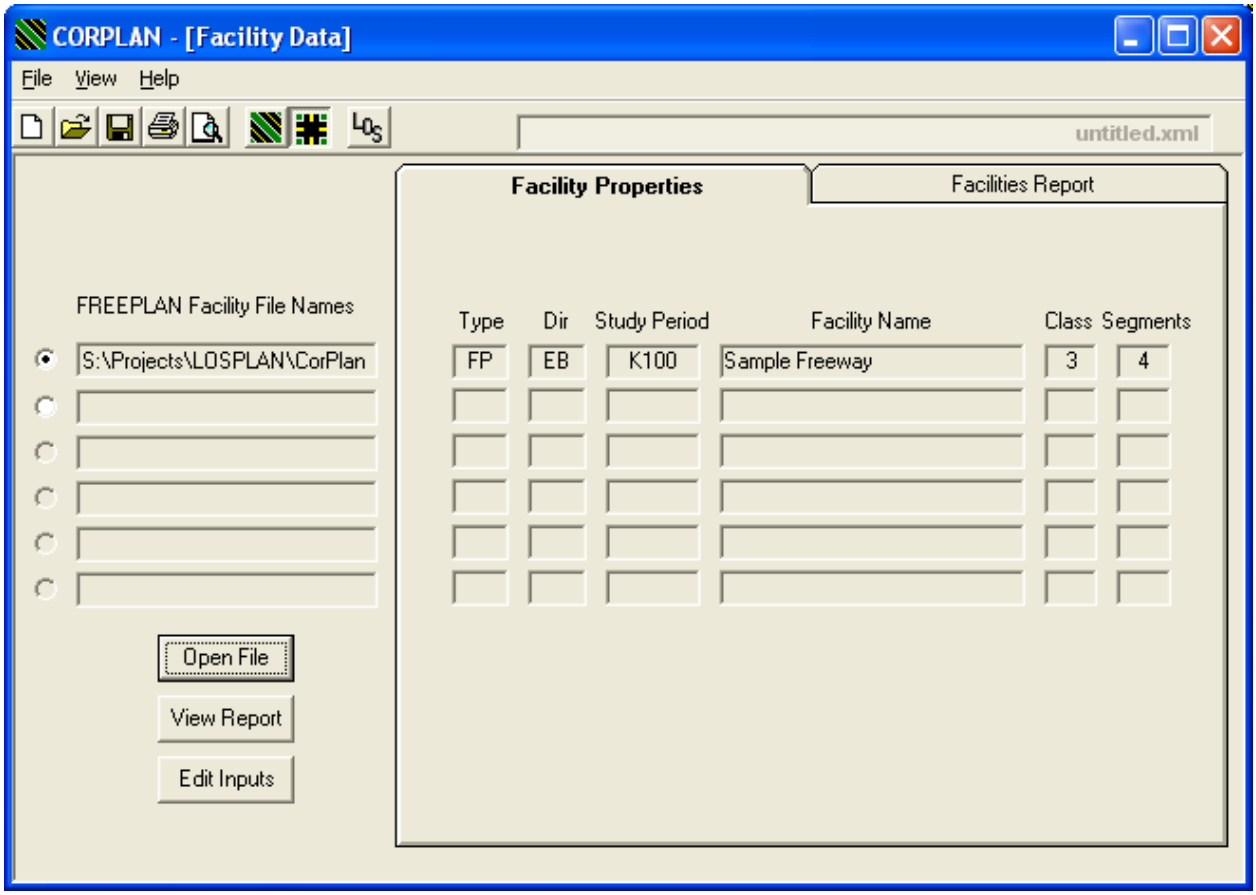


Figure 11. FREEPLAN File Loaded Into Facility Properties Screen

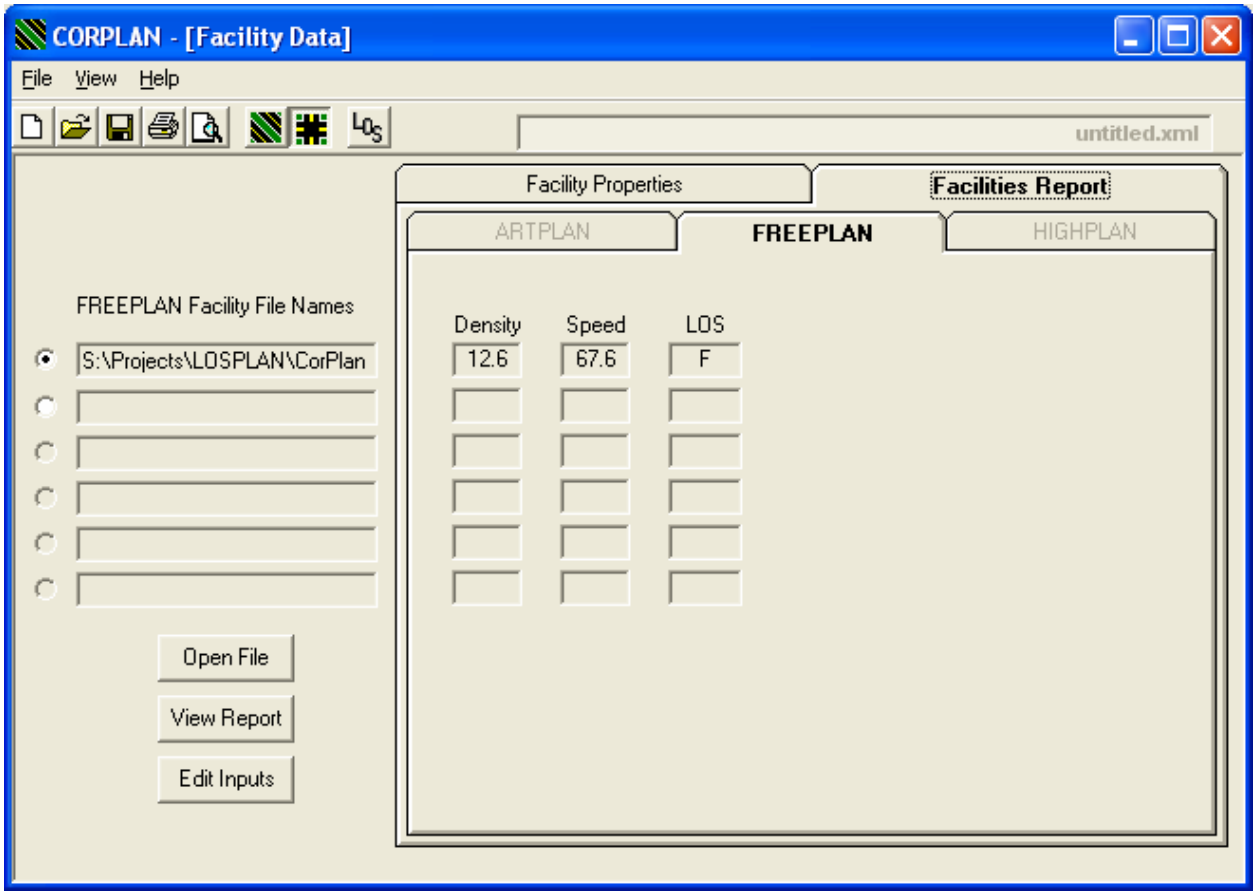


Figure 12. Facilities Report Screen for Loaded FREEPLAN File

The current version of CORPLAN is included on the enclosed CD.

APPENDICES

APPENDIX A:

LOSPLAN COMPONENT PROGRAM USER INTERFACE SCREENS

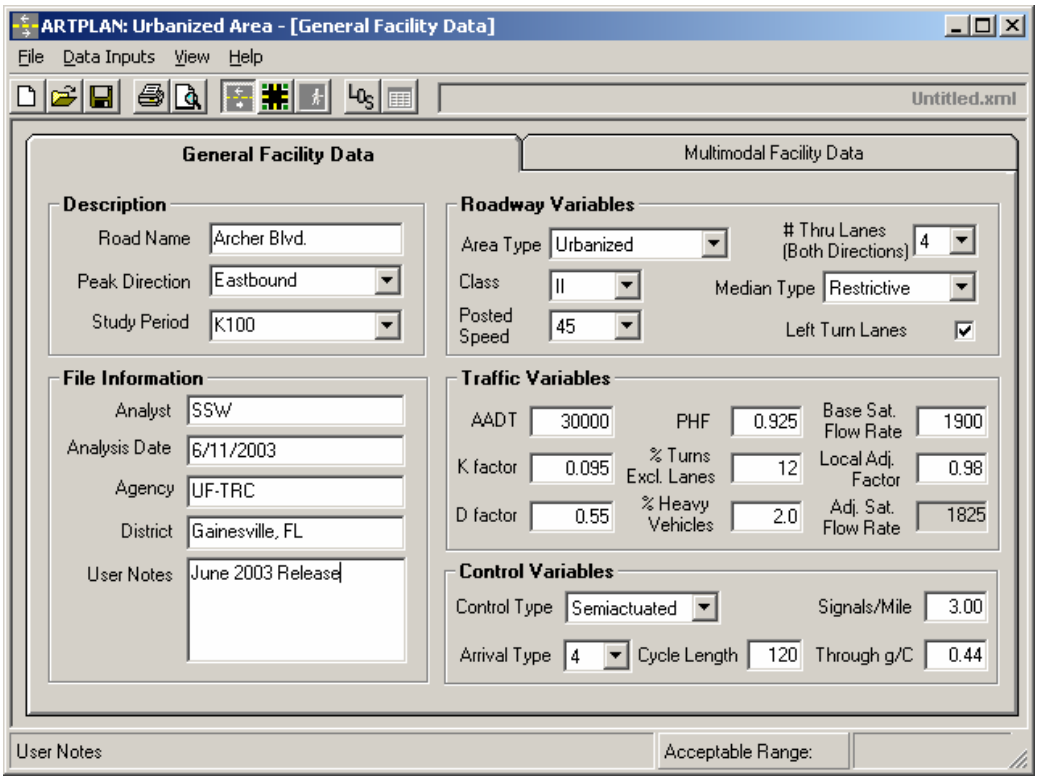


Figure 13. ARTPLAN General Facility Data Input Screen

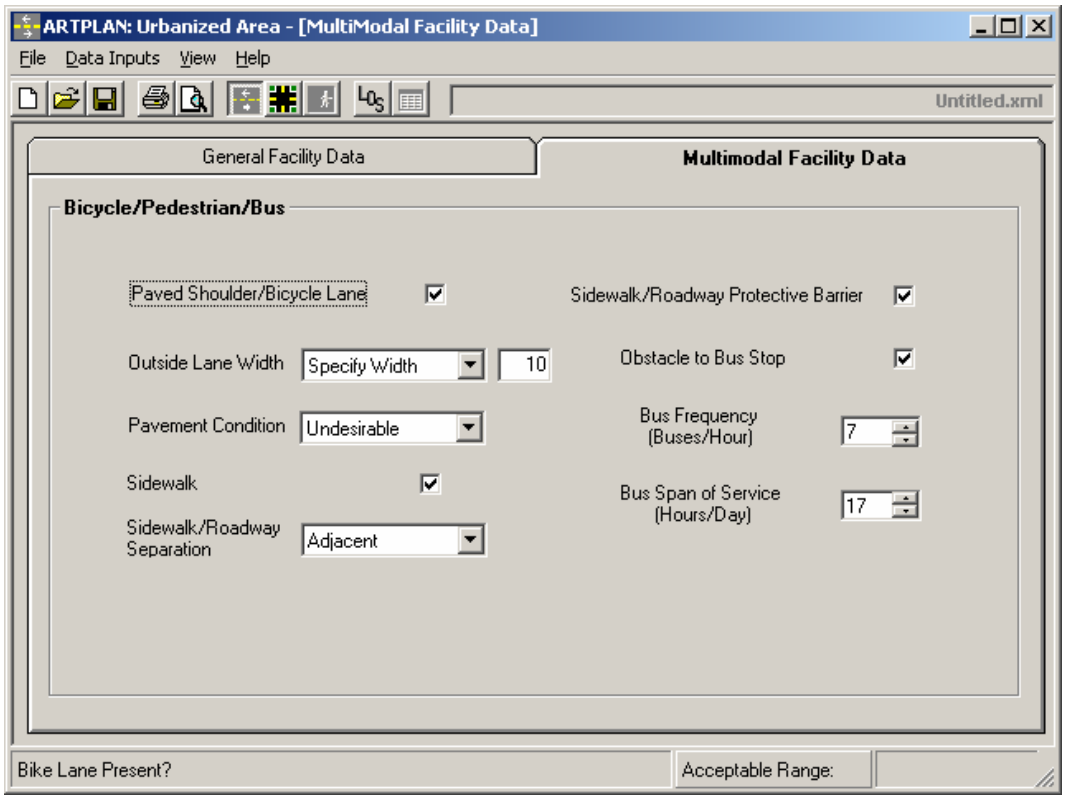


Figure 14. ARTPLAN Multimodal Facility Data Input Screen

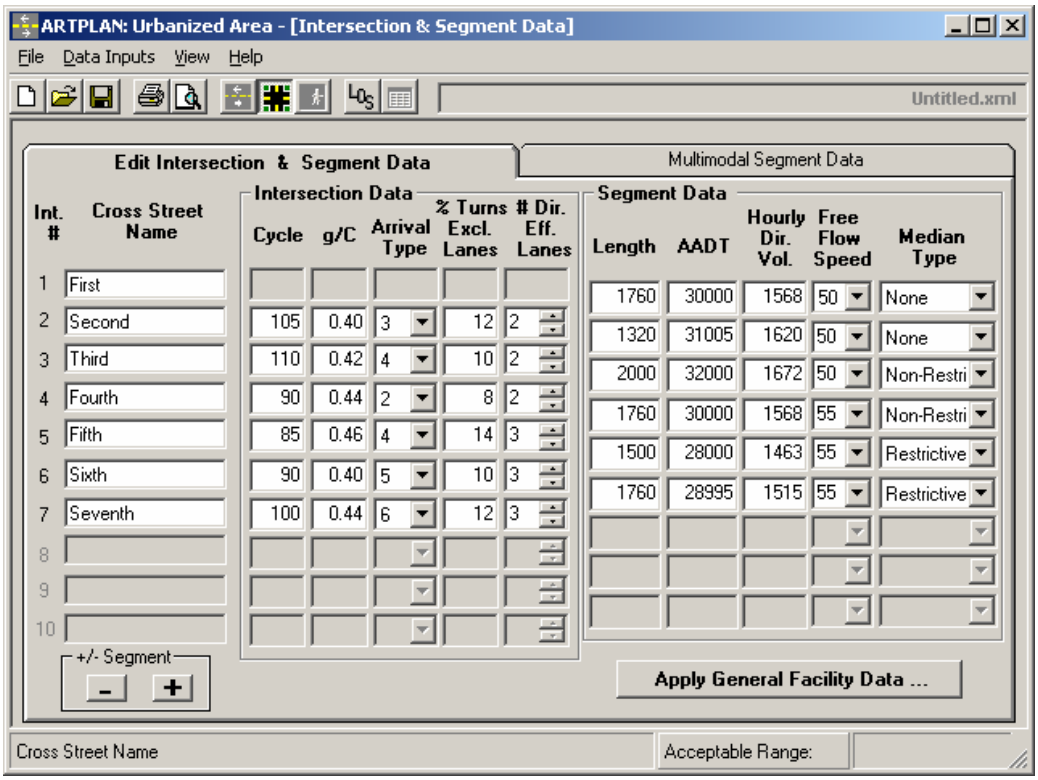


Figure 15. ARTPLAN Intersection and Segment Data Input Screen

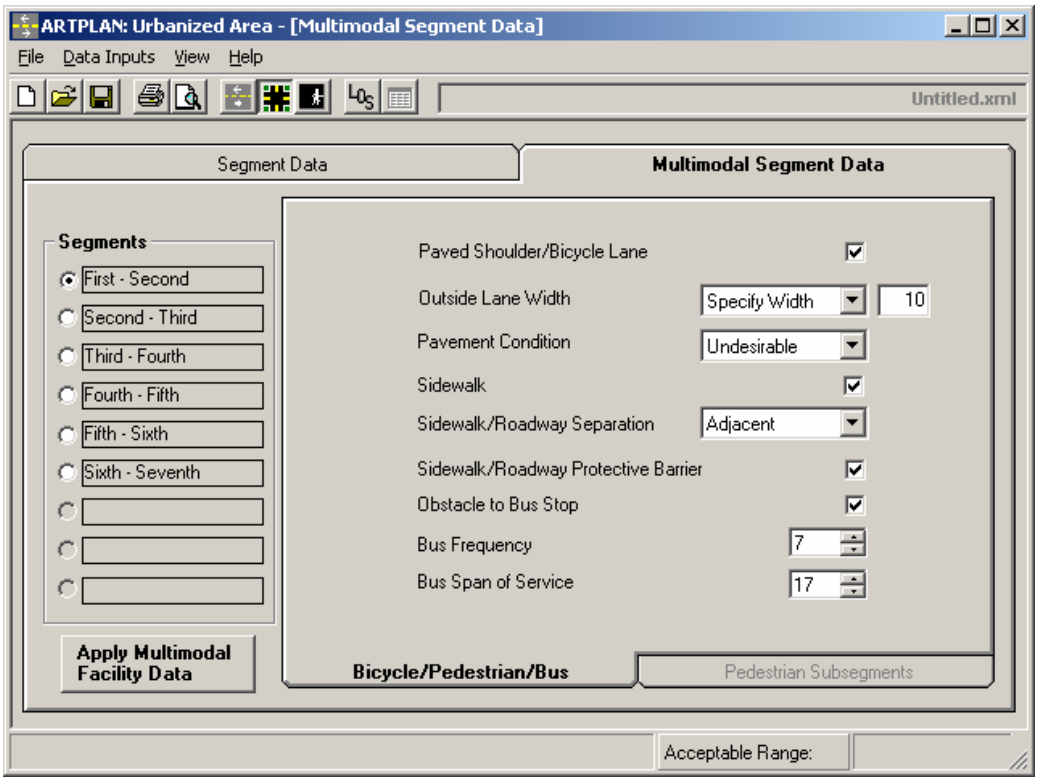


Figure 16. ARTPLAN Multimodal Segment Data Input Screen (Bicycle/Pedestrian/Bus)

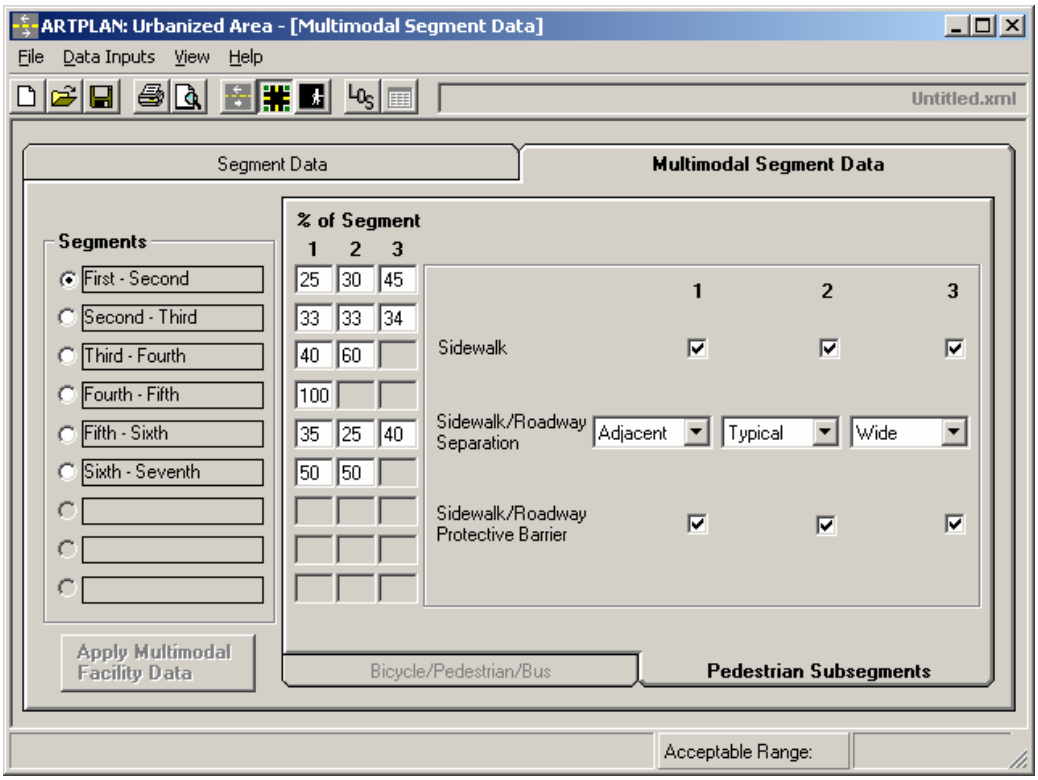


Figure 17. ARTPLAN Pedestrian Subsegments Data Input Screen

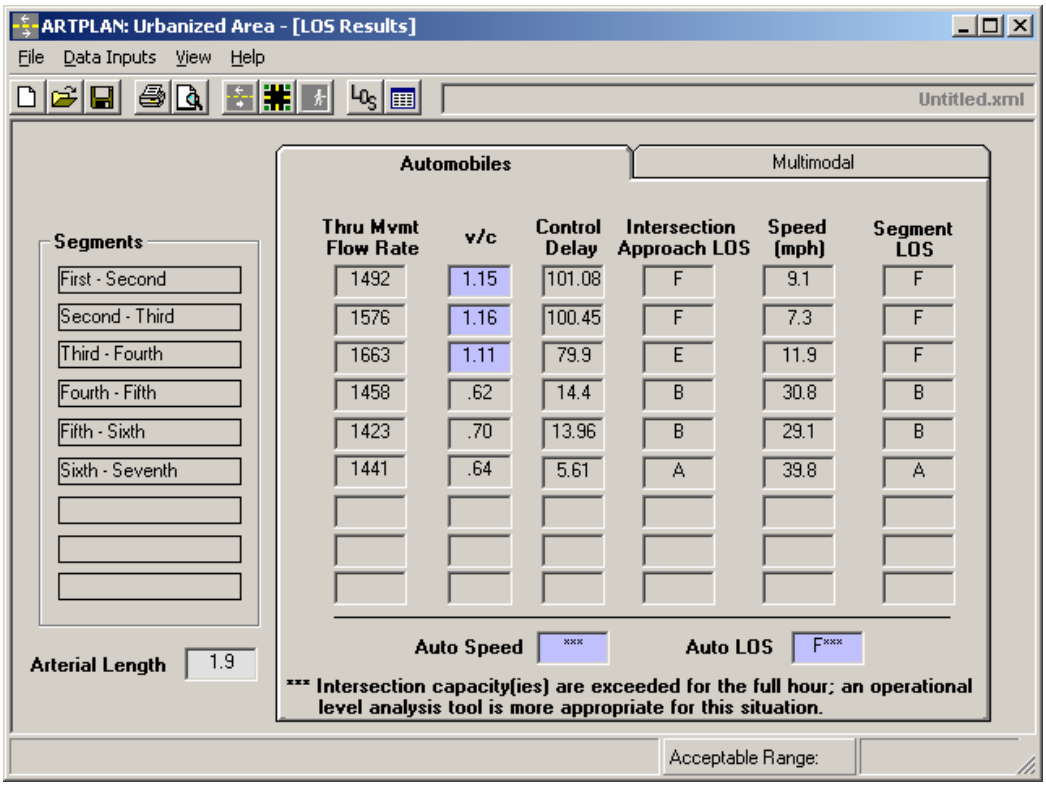


Figure 18. ARTPLAN Results Screen (Automobiles)

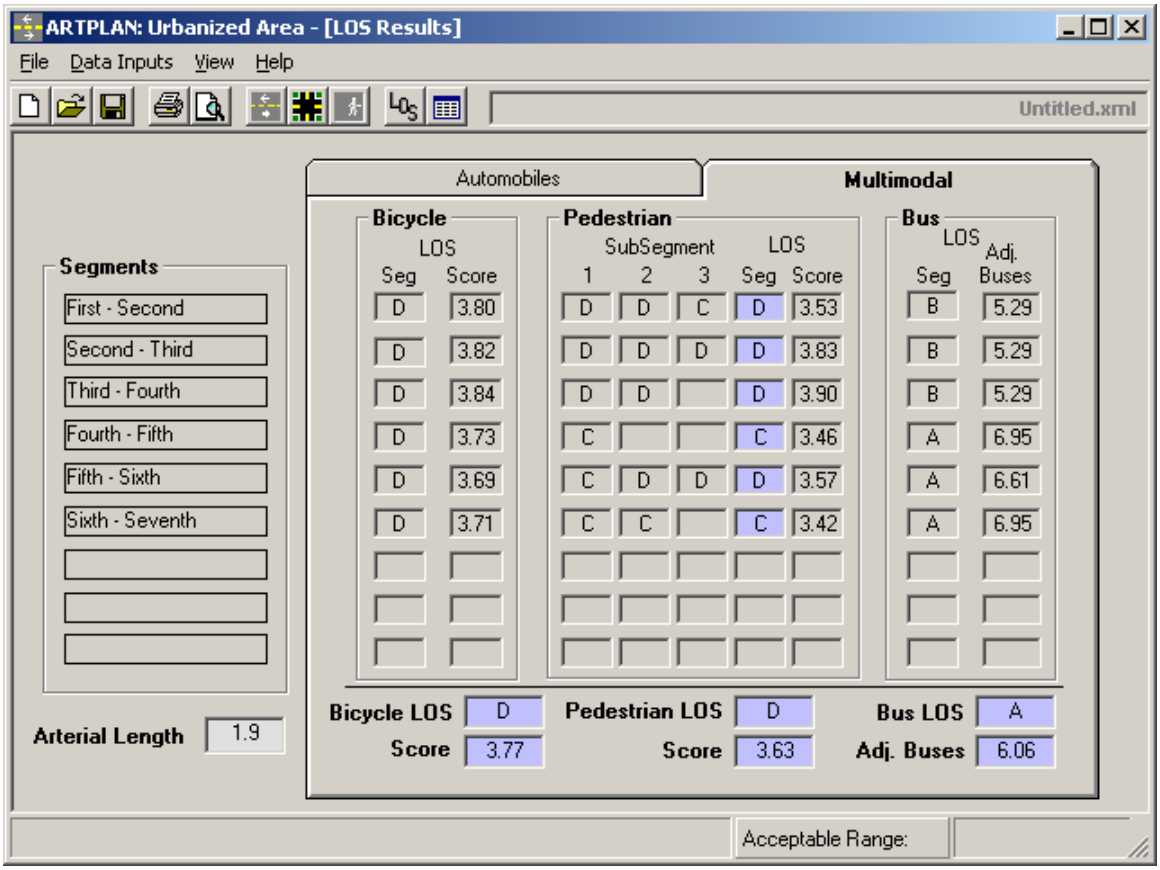


Figure 19. ARTPLAN Results Screen (Multimodal)

Maximum Service Volumes - Urbanized Area

Automobile | Bicycle | Pedestrian | Bus

Lanes	A	B	C	D	E
Hourly Volume in Peak Direction					
1	**	330	680	760	***
2	**	750	1420	1510	***
3	**	1180	2160	2270	***
4	**	1620	2890	3030	***
*	**	850	1490	1550	***
Hourly Volume in Both Directions					
2	**	590	1230	1380	***
4	**	1350	2590	2750	***
6	**	2140	3920	4130	***
8	**	2940	5250	5510	***
*	**	1550	2710	2820	***
Annual Average Daily Traffic					
2	**	6200	12900	14500	***
4	**	14300	27200	29000	***
6	**	22600	41300	43500	***
8	**	31000	55300	58000	***
*	**	16300	28500	29600	***

* Service Volumes for the specific facility being analyzed, based on the number of lanes appearing in the segment data screen.

** Cannot be acheived using table input value defaults.

*** Not applicable for that level of service letter grade. See generalized tables notes for more details.

Close

Figure 20. ARTPLAN Maximum Service Volumes Screen (Automobile)

Maximum Service Volumes - Urbanized Area

Automobile | Bicycle | Pedestrian | Bus

Lanes	A	B	C	D	E
Motor Vehicle Hourly Volume in Peak Direction					
1	**	150	340	>340	***
2	180	300	670	>670	***
3	260	450	1010	>1010	***
4	350	600	1340	>1340	***
*	210	360	820	>820	***
Motor Vehicle Hourly Volume in Both Directions					
2	**	280	610	>610	***
4	320	550	1220	>1220	***
6	480	820	1830	>1830	***
8	640	1100	2440	>2440	***
*	380	650	1500	>1500	***
Annual Average Daily Traffic					
2	**	2900	6400	>6400	***
4	3400	5800	12900	>12900	***
6	5100	8700	19300	>19300	***
8	6700	11600	25700	>25700	***
*	4000	6800	15800	>15800	***

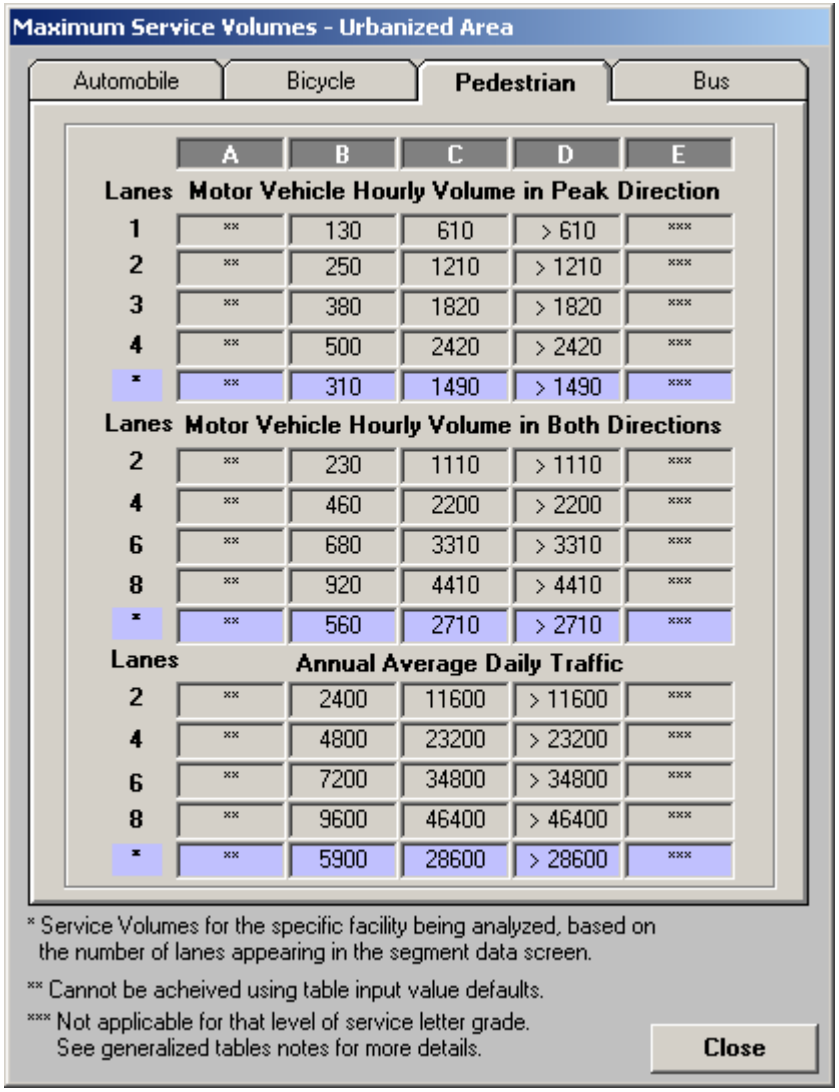
* Service Volumes for the specific facility being analyzed, based on the number of lanes appearing in the segment data screen.

** Cannot be acheived using table input value defaults.

*** Not applicable for that level of service letter grade. See generalized tables notes for more details.

Close

Figure 21. ARTPLAN Maximum Service Volumes Screen (Bicycle)



Maximum Service Volumes - Urbanized Area

Automobile | Bicycle | **Pedestrian** | Bus

Lanes	A	B	C	D	E
	Motor Vehicle Hourly Volume in Peak Direction				
1	**	130	610	> 610	***
2	**	250	1210	> 1210	***
3	**	380	1820	> 1820	***
4	**	500	2420	> 2420	***
*	**	310	1490	> 1490	***
	Motor Vehicle Hourly Volume in Both Directions				
2	**	230	1110	> 1110	***
4	**	460	2200	> 2200	***
6	**	680	3310	> 3310	***
8	**	920	4410	> 4410	***
*	**	560	2710	> 2710	***
	Annual Average Daily Traffic				
2	**	2400	11600	> 11600	***
4	**	4800	23200	> 23200	***
6	**	7200	34800	> 34800	***
8	**	9600	46400	> 46400	***
*	**	5900	28600	> 28600	***

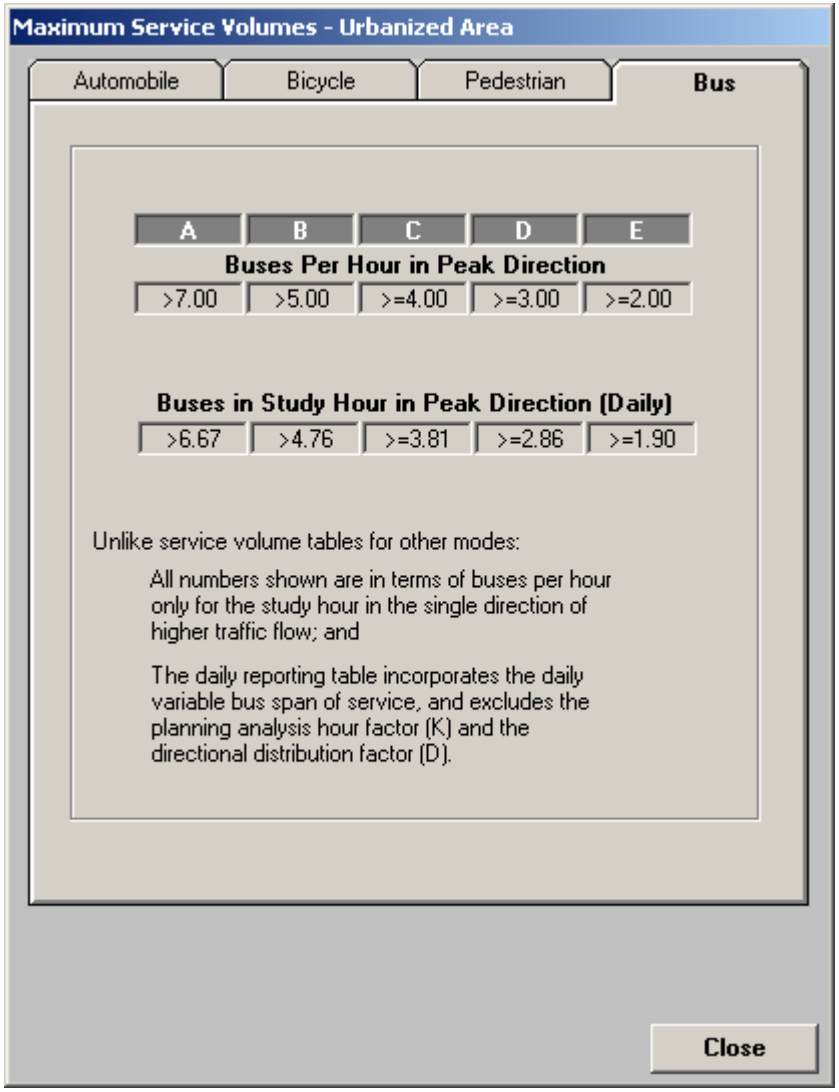
* Service Volumes for the specific facility being analyzed, based on the number of lanes appearing in the segment data screen.

** Cannot be acheived using table input value defaults.

*** Not applicable for that level of service letter grade. See generalized tables notes for more details.

Close

Figure 22. ARTPLAN Maximum Service Volumes Screen (Pedestrian)



Maximum Service Volumes - Urbanized Area

Automobile | Bicycle | Pedestrian | **Bus**

A	B	C	D	E
Buses Per Hour in Peak Direction				
>7.00	>5.00	>=4.00	>=3.00	>=2.00
Buses in Study Hour in Peak Direction (Daily)				
>6.67	>4.76	>=3.81	>=2.86	>=1.90

Unlike service volume tables for other modes:

All numbers shown are in terms of buses per hour only for the study hour in the single direction of higher traffic flow; and

The daily reporting table incorporates the daily variable bus span of service, and excludes the planning analysis hour factor (K) and the directional distribution factor (D).

Close

Figure 23. ARTPLAN Maximum Service Volumes Screen (Bus)

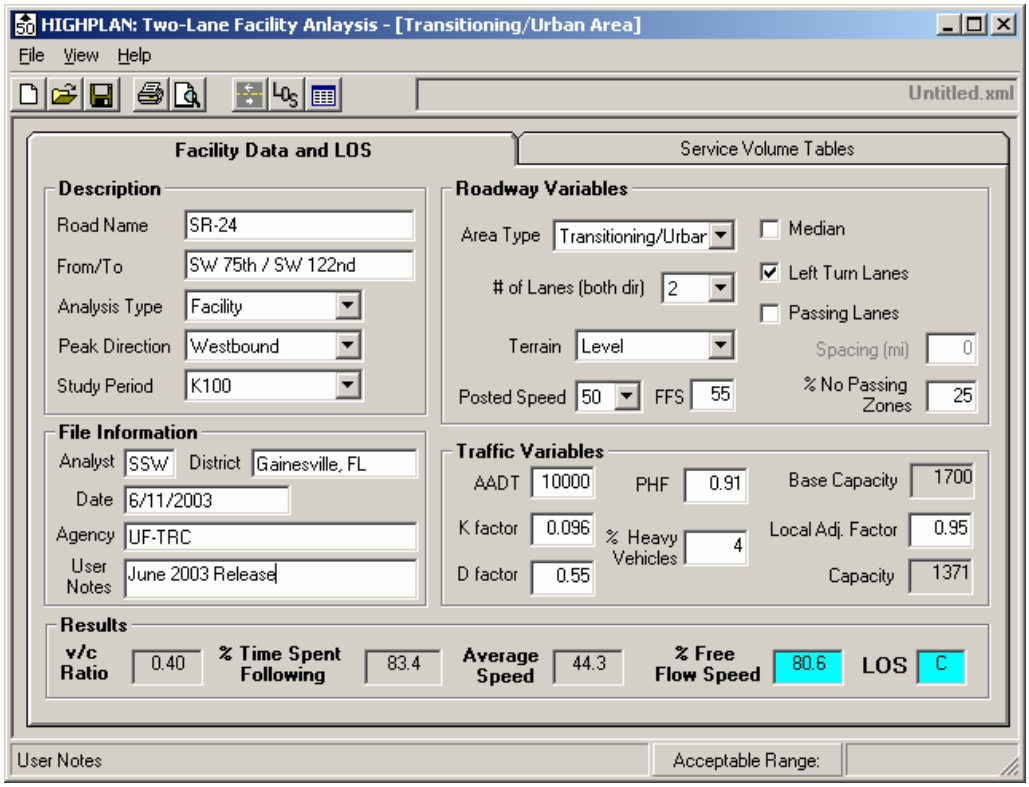


Figure 24. HIGHPLAN Main Screen Showing Two-Lane Inputs and LOS Results

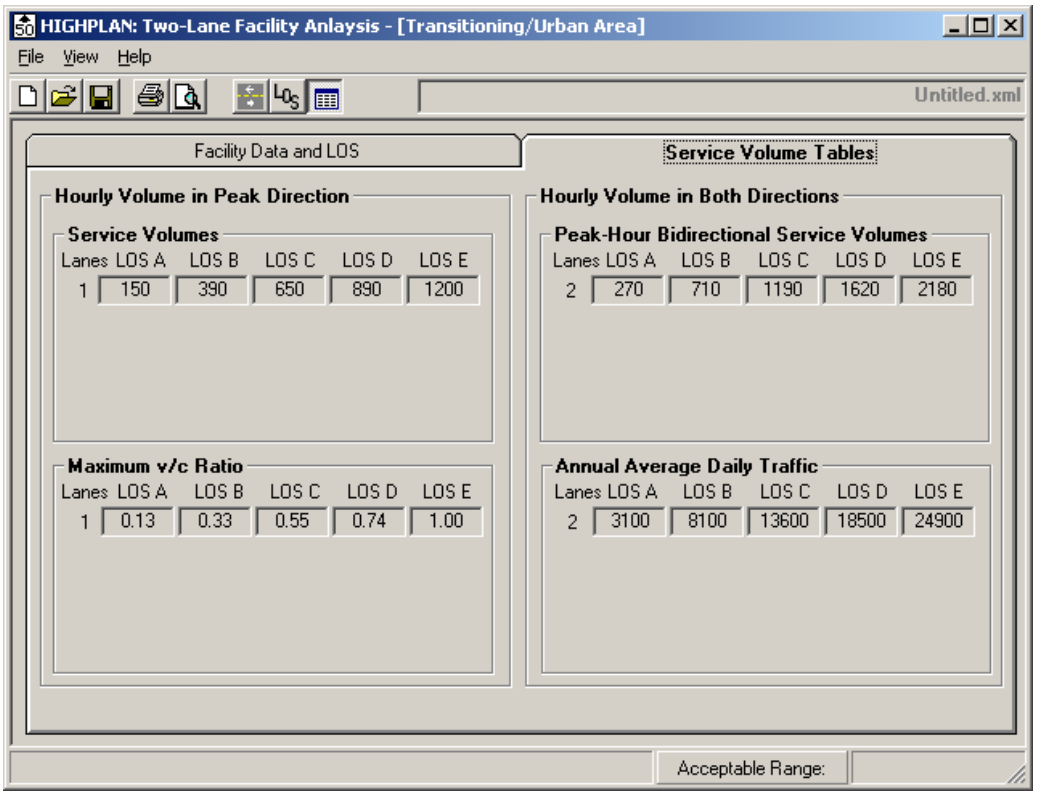


Figure 25. HIGHPLAN Maximum Service Volume Tables Screen (for Two-Lane)

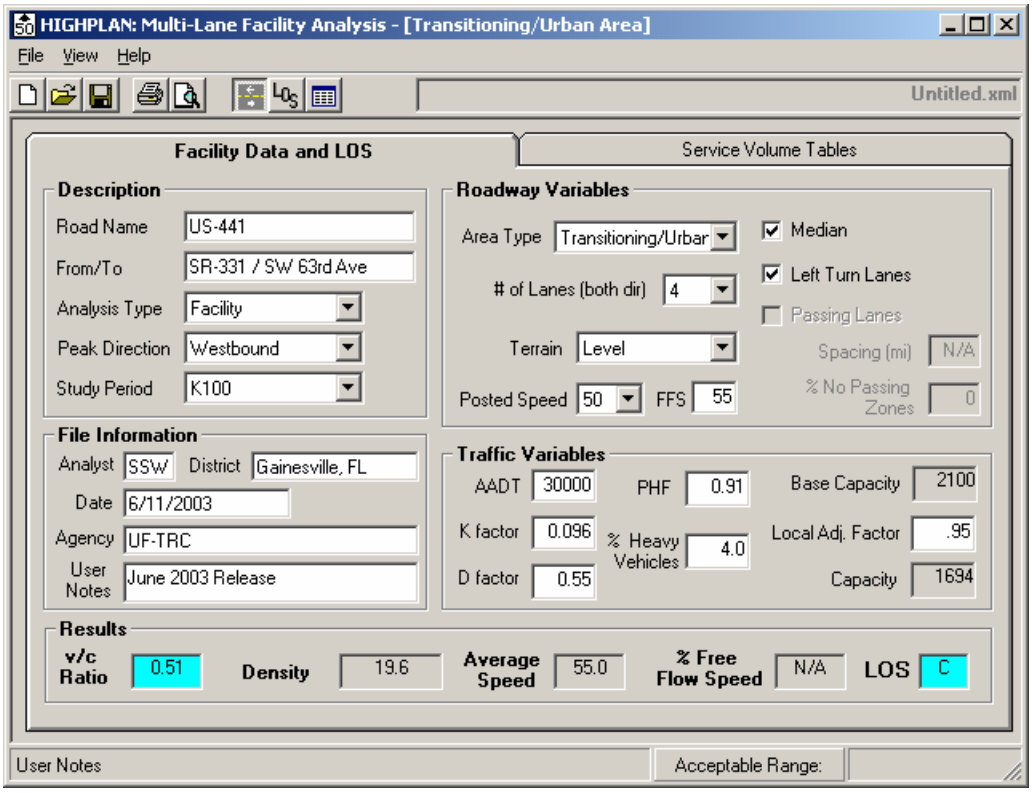


Figure 26. HIGHPLAN Main Screen Showing Multilane Inputs and LOS Results

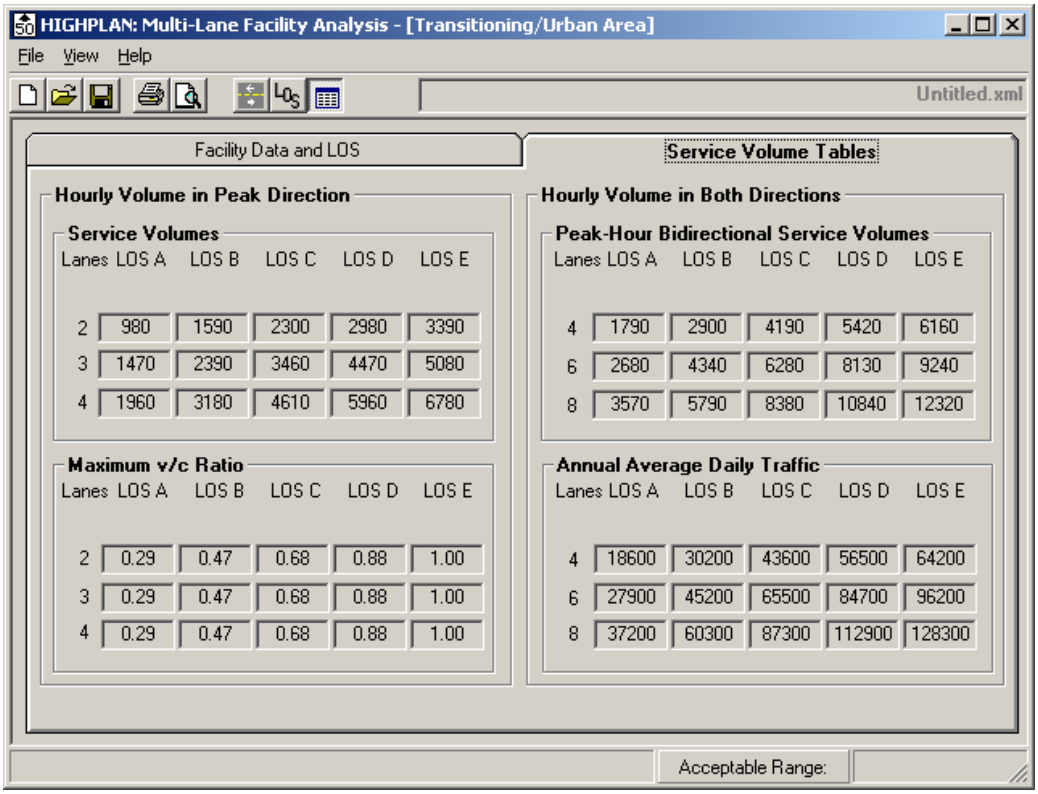


Figure 27. HIGHPLAN Maximum Service Volume Tables Screen (for Multilane)

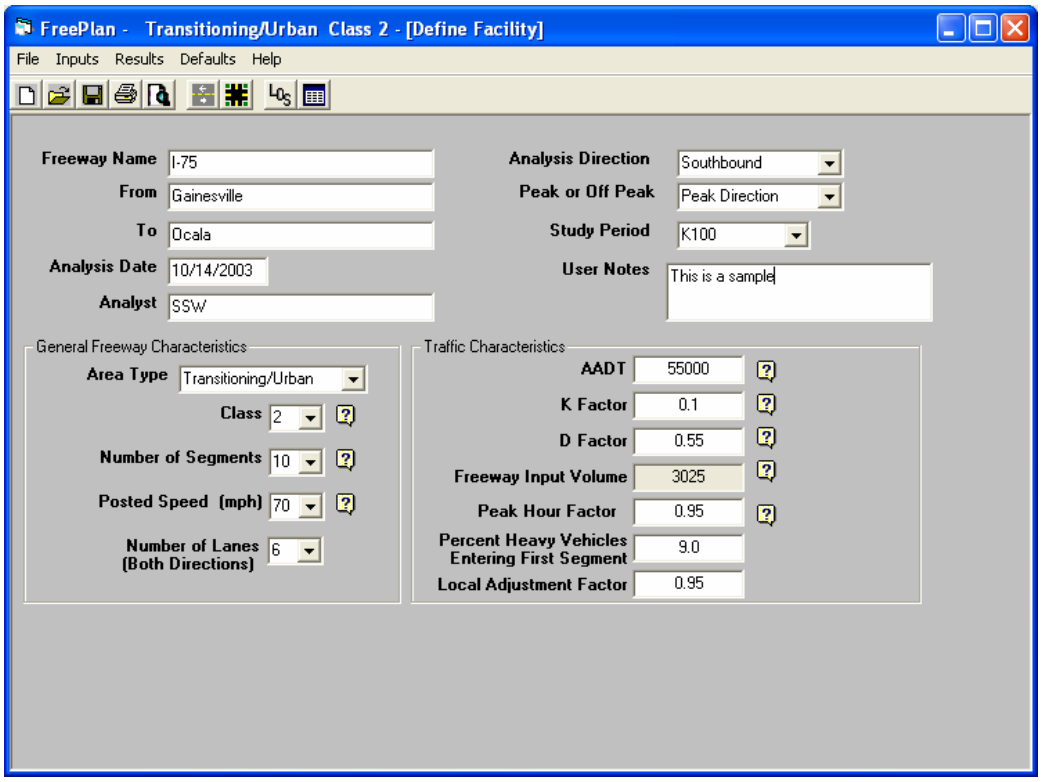


Figure 28. FREEPLAN Facility Data Screen

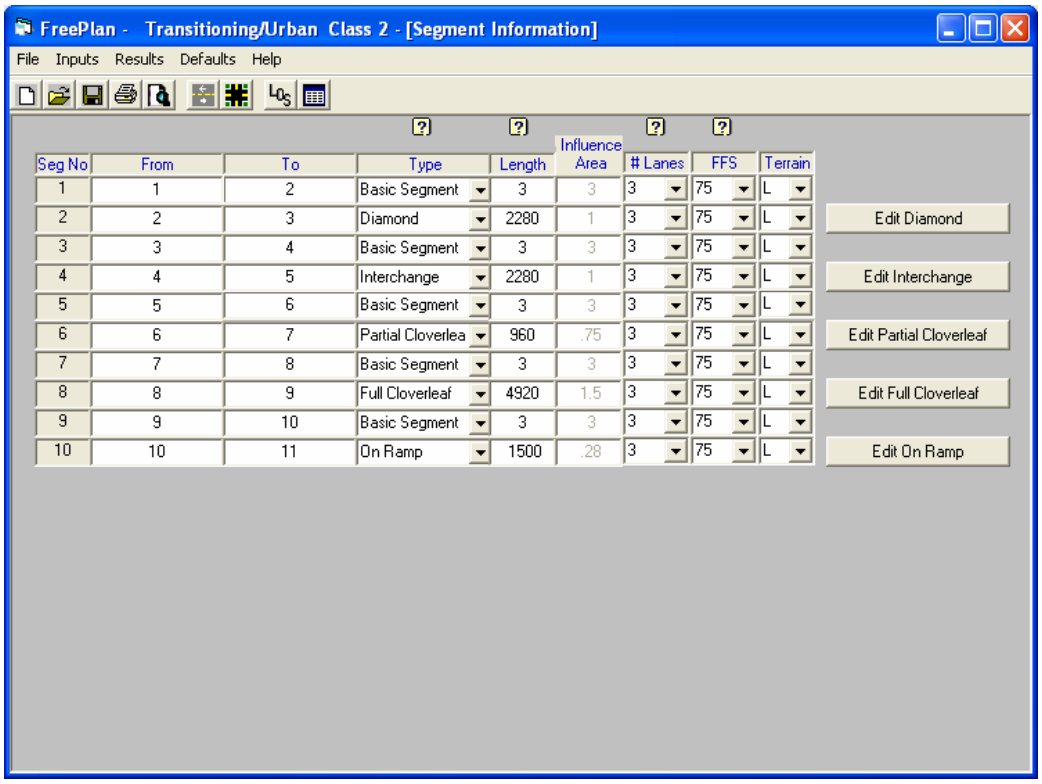


Figure 29. FREEPLAN Segment Data Screen

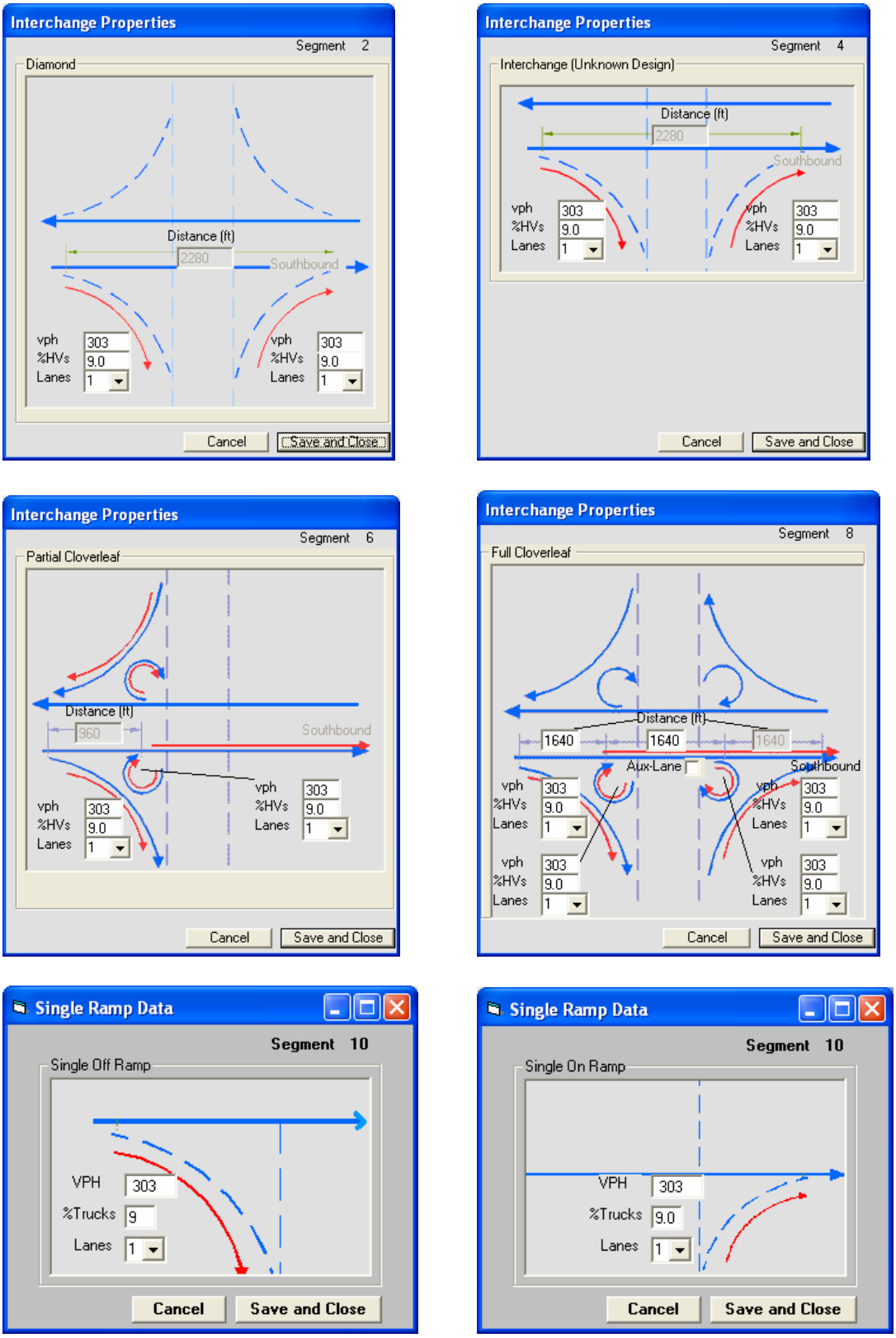


Figure 30. FREEPLAN Interchange Data Specification Screens

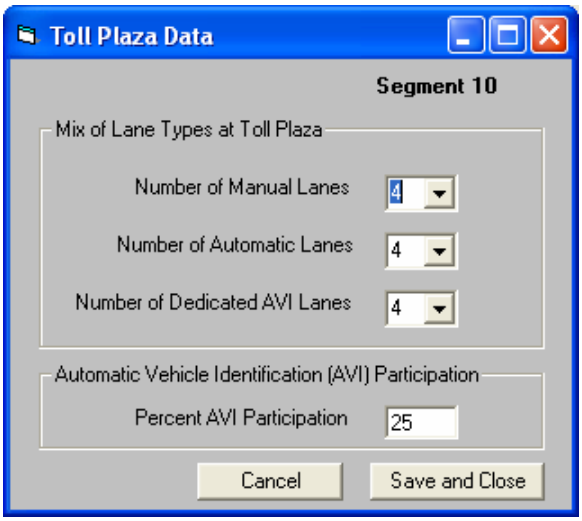


Figure 31. FREEPLAN Toll Plaza Data Specification Screen

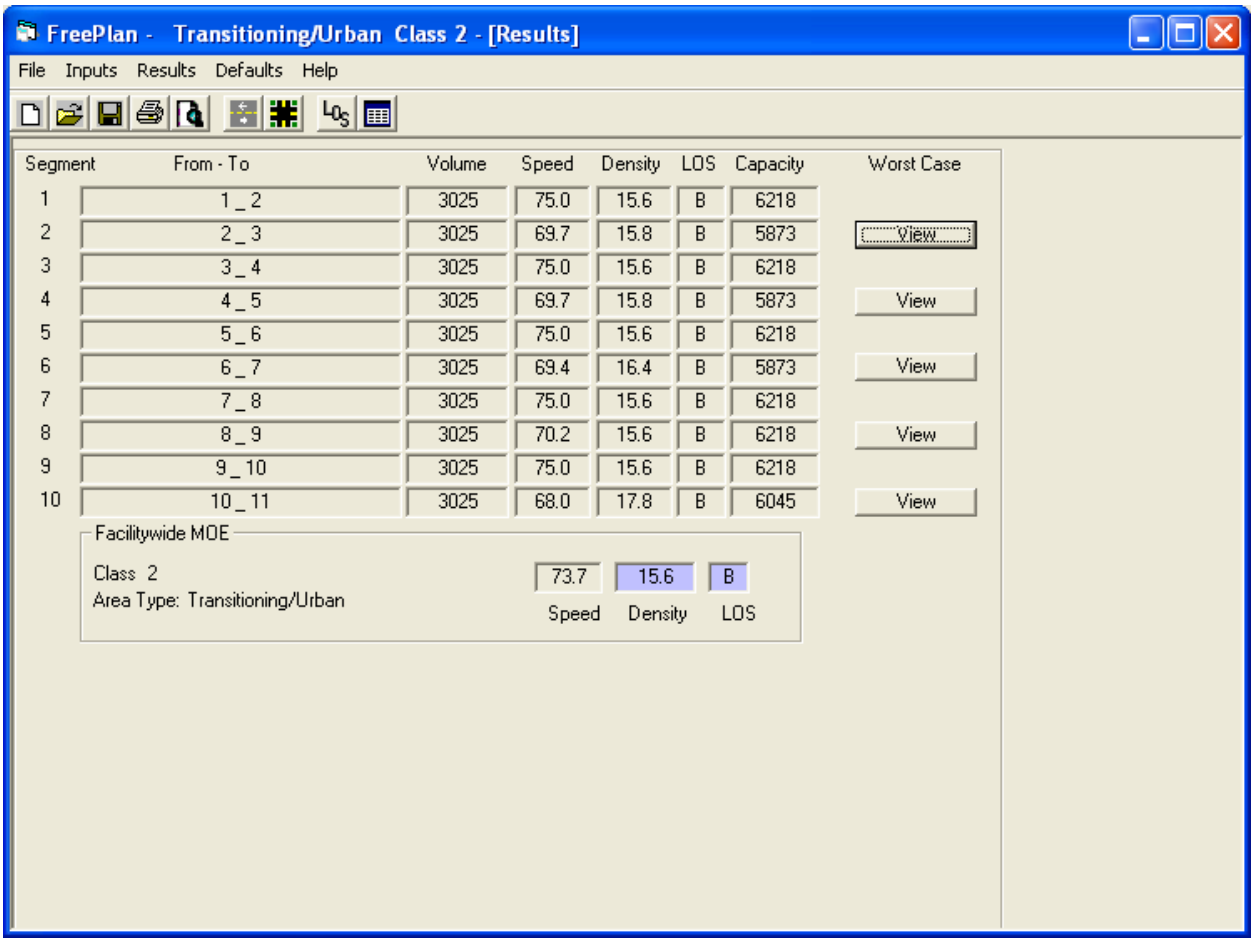


Figure 32. FREEPLAN Results Display Screen

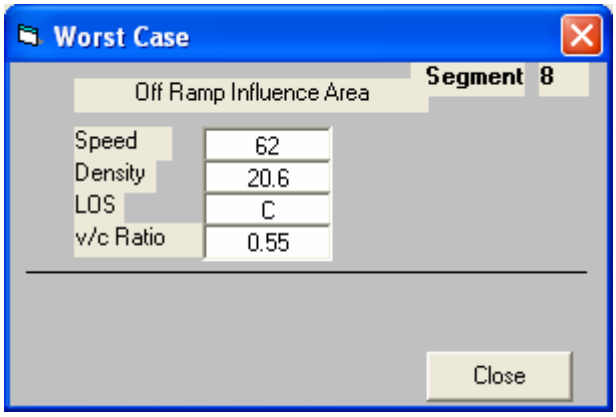
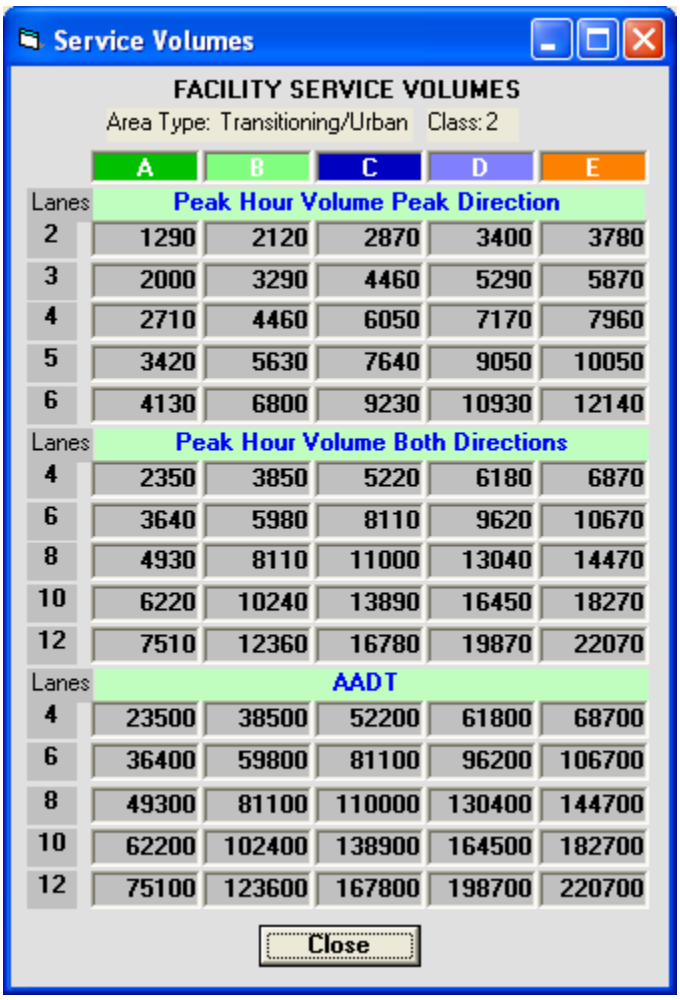


Figure 33. FREEPLAN Worst Case Results Display for Interchange Ramps



Service Volumes

FACILITY SERVICE VOLUMES

Area Type: Transitioning/Urban Class: 2

Lanes	A	B	C	D	E
	Peak Hour Volume Peak Direction				
2	1290	2120	2870	3400	3780
3	2000	3290	4460	5290	5870
4	2710	4460	6050	7170	7960
5	3420	5630	7640	9050	10050
6	4130	6800	9230	10930	12140
	Peak Hour Volume Both Directions				
4	2350	3850	5220	6180	6870
6	3640	5980	8110	9620	10670
8	4930	8110	11000	13040	14470
10	6220	10240	13890	16450	18270
12	7510	12360	16780	19870	22070
	AADT				
4	23500	38500	52200	61800	68700
6	36400	59800	81100	96200	106700
8	49300	81100	110000	130400	144700
10	62200	102400	138900	164500	182700
12	75100	123600	167800	198700	220700

Close

Figure 34. FREEPLAN Maximum Service Volumes Screen

APPENDIX B:

LOSPLAN COMPONENT PROGRAM REVISION HISTORIES

ARTPLAN Revision History

Version 5.1.0 (official release, June 2003)

- Another revision to the sequence of inputs for the segment (auto) screen
  - Added median type input to segment screen
    - For files created with a previous version, the segment median type will be defaulted to that specified for the general facility median type

- Upper limit of % turns from exclusive lanes (PTXL) was changed from 45 to 65 for segment data input screen. Facility data input screen was left at 45%, but the status bar text now indicates that up to 65% can be entered in Segment screen.

- Added tool tip text to 'Adjusted Sat Flow Rate' input field, indicating the factors that influence this value.

- A new entry on the service volume tables form that indicates service volumes corresponding to an arterial with an unbalanced number of lanes between segments
  - This was also added to the formatted report (this required expanding printed report to 5 pages)
  - Added a label indicating the area type to the title bar of the service vols form
  - Some minor cosmetic enhancements were made to service vols form

- Additions/revisions to formatted report
  - Added: Program name, program version, segment median type
  - Re-ordered segment data fields to match with UI
  - Added segment number and ending cross street name for segment in segment input summary section
  - Revisions to formatted report for 'no sidewalk' selection -> 'N/A'
  - Asterisk text from service volumes screen added to print preview/formatted report

- Linking to FDOT website from Help menu (LOS Handbook) and About ARTPLAN dialog box

- Minor revisions to On-line Help file

- Added logic to now determine whether user needs to be queried about saving data

- Added logic to file open/reading code such that a saved file with more than 9 segments will not bomb program—extra segment information just gets ignored.

- Added logic to check for over-capacity conditions for the full hour (1/PHF) on any segment and a corresponding warning message.

- Replaced custom XML reading routine with Microsoft's XML reading utility (MSXML version 4)

- Added remote automation capability (i.e., program can be called remotely and manipulated from another application). This can be accomplished with a command line instruction and parameters. Documentation on this capability will be forthcoming.

- Added batch file processing capability (i.e., many data files can be automatically processed by HIGHPLAN, for LOS results)

Version 5.0.4 (test release)

- Re-ordered individual field restore defaults buttons to match re-ordering of segment input fields from previous version

- Minor revisions to On-line Help file

- Fix to calculations which resulted in program crash when all segment g/C values were set equal to 1.0

- The upper limit of peak hour volume was changed to 5000, regardless of total number of lanes. This will generally keep the AADT value from exceeding its upper limit of 100,000. (Is this the right version?)

  For example, 5000/0.1 (K) = 50,000 -> 50,000/0.5 (D) = 100,000

Version 5.0.3 (test release)

- Rearranged input controls on segment data input page to correspond with sequence of roadway, traffic, and control variables shown on facility data input form

- Re-organized the formatted report so that the automobile inputs and LOS were grouped together, and likewise for multimodal

- The cross-street names were added to the printed report, in lieu of just a segment number

- The signals/mile input on the general facility data input screen and printed report now shows 2 decimal places

Bug Fixes

#1: Wrong error message appeared when a character was entered into the Local Adj. Factor box.

#2: Program allowed you to save a file with invalid data entered for hourly volume (segment data screen).

#3: Can input a value between 1 and 5 in the # of Dir. Lanes box without using arrow buttons, but using arrow buttons, can input a value between 1 and 4. Changed to allow a maximum of 4 directional lanes with arrow button.

#4: NOTE: This bug caused the program to crash. When trying to open a saved file with the Class Type as Signal, a fatal error message appeared.

#5: If Median Type was set to 'Non-Restrictive' or 'None' and then the results were calculated, upon returning to the General Facility Data page the Median Type was reset to Restrictive.

#6: NOTE: This bug caused the program to crash. An error message appeared when trying to open a saved file with either the '&' or the '<' and '>' characters in any of the information boxes. These characters are generally invalid for XML purposes. If you really need to use these characters, use the following syntax, append '&' with '&' and '<' or '>' with '>glt;'.

#7: Could type in any number or character into the Class box and perform a LOS analysis.

#8: NOTE: This bug caused the program to crash when the program was set with these parameters: Area Type as Transitioning/Urban, Median Type as Non-Restrictive, Class as III. Save the file, close, and then reopen it. An error message appeared and clicking OK shuts the program down.

#9: In Multimodal Segment Data, specifying Outside Lane Widths in segments 2 and 3 and then saving, it didn't save the inputs and overrode with values of 8 in these segments.

#10: The Help file isn't running from the pull-down menu. A shortcut on the desktop is created to get the Help file to work properly. This is a problem with the installation routine (fixed in the non NT-compatible version, not in the other version yet).

#11: Signal control type field did not link to appropriate Help file section.

#12: After clicking OK on the error message that appeared after inputting a K-factor out of range, the default value that was in the K-factor box reverted to an incorrect default value.

#13: Print Preview gave an error when a control input values contained the '&' character.

#14: The Print Preview was not displaying the End Intersection value correctly.

#15: Some incorrect default values were displayed when the forms GenInput, SegInput, or MultiModal were displayed.

#16: When an alpha character was inputted into the 'Outside Lane Width' control and then the user tried to tab out, it gave the wrong error message.

#17: After performing an LOS with the Class box on Signal and the Area Type on Transitioning/Urban, an error message appeared when the user clicked back on the Edit Facility Data icon.

#18: When you saved a file with the Class control set to Signal, closed the file and then opened it, the Segment Button was enabled.

#19: When you increased the number of segments to the maximum and then clicked on the Restore Defaults Button, the Add Segments Button was disabled even though the number of segments is less than the maximum. The reverse situation was also true for the Remove Segments button.

#20: When you click on the LOS button, the Service Volumes Button gets enabled. When the Print Preview Form was invoked, the Service Volume Button was still enabled. This button should only be enabled when the user is viewing the LOS results.

#21: Fixed an error related to the sequence of file saving, closing, and then re-opening that resulted from a combination of rural area type and signal class inputs.

Version 5.0.2 (test release)

- Several user interface refinements, particularly with respect to data input validation
- Fixed a bus LOS calculation problem

Version 5.0.1

- Initial test release

HIGHPLAN Revision History

Version 1.1.0 (official release, June 2003)

- Level of Service results no longer appear upon start-up of the application, the LOS button must now be pressed to display results. This resolved an issue with the updating of the service volumes screen, in addition to a couple of other minor user interface issues.

- Added logic to now determine whether user needs to be queried about saving data

- Additions/revisions to formatted report
  - Added: Program name, program version, peak direction, study period, analysis type
  - Fixed problem with the displaying of number of lanes for service volumes
  - Fixed print preview problem for initial display of level of service MOE's

- Moved 'Analysis Type' input from the 'Roadway Variables' section to the 'Description' section

- Changed the label 'Adj. Capacity' to just 'Capacity'

- Added tool tip text to 'Capacity' input field, indicating the factors that influence this value.

- Incorrect tool tip text for 'Base Capacity'. Text Removed.

- Changed 'FFS' field to user editable. Set the allowable range to 40 – 75 mph.

- Changed 'Date' field to user editable

- Re-ordered screen view and LOS toolbar buttons

- Minor revisions to status bar text descriptors

- Minor revisions to On-line Help file, including context specific topic indexing

- Changed the following XML file tag names (Note: Data sets created with previous versions of HIGHPLAN will not read with this version unless the following tag names are changed).

  - <Class_HCM> → <HighwayClass_HCM>
  - <PropPassingZone> → <NoPassZonePct>
  - <HVPcnt> → <HVPct>
  - <AverageTravelSpeed> → <AvgTravelSpeed>

- Linking to FDOT website from Help menu (LOS Handbook) and About ARTPLAN dialog box

- Added remote automation capability (i.e., program can be called remotely and manipulated from another application). This can be accomplished with a command line instruction and parameters. Documentation on this capability will be forthcoming.

- Added batch file processing capability (i.e., many data files can be automatically processed by HIGHPLAN, for LOS results)

- Fixed bug that was defaulting passing lane spacing to 2 miles, even when no passing lanes were present

- Only two digits were being accommodated for % No Passing Zones, so an input value of '100' was being truncated to '10'. Thus, in version 1.0.4, the results were much more optimistic than they should have been for an input of 100% No Passing Zones.

Version 1.0.4 (test release)

- Minor revisions to On-line Help file
- Minor revisions to status bar text descriptors

Version 1.0.3 (test release)

Bug Fixes

#1: Could input any type of character in the % Heavy Vehicles box and % No Passing Zone box with no error message appearing.

#2: You were allowed to save a file with invalid characters in various input controls.

#3: Could get results after performing a LOS with an invalid character in the % Heavy Vehicles box.

#4: Could get different results after performing a LOS with invalid characters in more than one entry box (i.e. % No Passing and % Heavy Vehicles) than with an invalid entry in only one entry box (as above error).

#5: In certain situations, the program hung when changing some parameters after the program was idle for an extended amount of time.

#6: The LOS and Service Volumes icons did not function with 4, 6, or 8 lanes selected, but did work with 2 lanes selected.

#7: If a parameter (such as Terrain) was changed, the Passing Lanes box stayed checked but the Spacing box turned gray. The Passing Lanes box had to be unchecked, then rechecked for the Spacing box to be activated.

#8: The User's Guide under the Help menu did not execute. This is a problem with the installation routine (fixed in the non NT-compatible version, not in the other version yet).

#9: Clicking on the LOS icon immediately after entering a value not within range for any parameters still showed results at the bottom of the screen.

#10: The F LOS value was missing when you manipulated the parameters for multilane (only) facilities for failing conditions.

#11: When an invalid character was input into the % Heavy Vehicle field, the program input a value of zero in the Adj. Capacity field.

#12: A LOS evaluation could be performed with the % No Passing and % Heavy Vehicles boxes empty.

#13: NOTE: This bug caused the program to crash. Characters could be entered after numerical values in the AADT, Base Capacity, % Heavy Vehicles, and % No Passing Zone variable fields.

#14: For multilane, No-passing Zone is 'N/A'. This caused non-numeric error checking problems.

#15: The Left Turn Lanes value (Yes or No) was not being read from a saved data file. The value always defaulted to unchecked (No).

#16: The Posted Speed box sometimes disappeared after opening a saved file.

#17: NOTE: This bug caused the program to crash. An error message appeared when trying to open a saved file with either the '&' or the '<' and '>' characters in any of the information boxes. These characters are generally invalid for XML purposes. If you really need to use these characters, use the following syntax, append '&' with '&' and '<' or '>' with '>glt;'.

#18: Any number up to 999 and any character could be typed in the Posted Speed box. A LOS inquiry could still be performed with the invalid number or character in the box.

#19: The Posted Speed values listing was reversed for the 2–lane and multilane conditions. Due to the 2-lane posted speed values being shown for the multilane situation, this caused the base saturation flow rate to be incorrect for certain speed selections.

#20: NOTE: This bug caused the program to crash when you opened a file after saving and closed the specific file (using File and Close). Upon opening the file, a fatal error message appeared.

#21: Upon closing a file that was opened from disk, the dialog box asking to save changes did not appear.

#22: The default passing lane spacing was changed from ‘0’ to ‘2’.

Version 1.0.2 (test release)

- Several user interface and calculation refinements

Version 1.0.1 (test release)

- Initial release

APPENDIX C:

BATCH PROCESSING INSTRUCTIONS

The Batch Processing utility has been provided to allow users to process multiple ARTPLAN, HIGHPLAN, or FREEPLAN input files. The following sequence of events will be automatically performed on each file in a group of selected files: 1) the inputs from the file will be loaded into the corresponding LOSPLAN program; 2) the results will be calculated for those inputs; and 3) the results will be saved back into the original file.

The Batch Processing utility can be selected from the 'File' menu of any of the three LOSPLAN programs. Figure 35 below illustrates this for the ARTPLAN program.

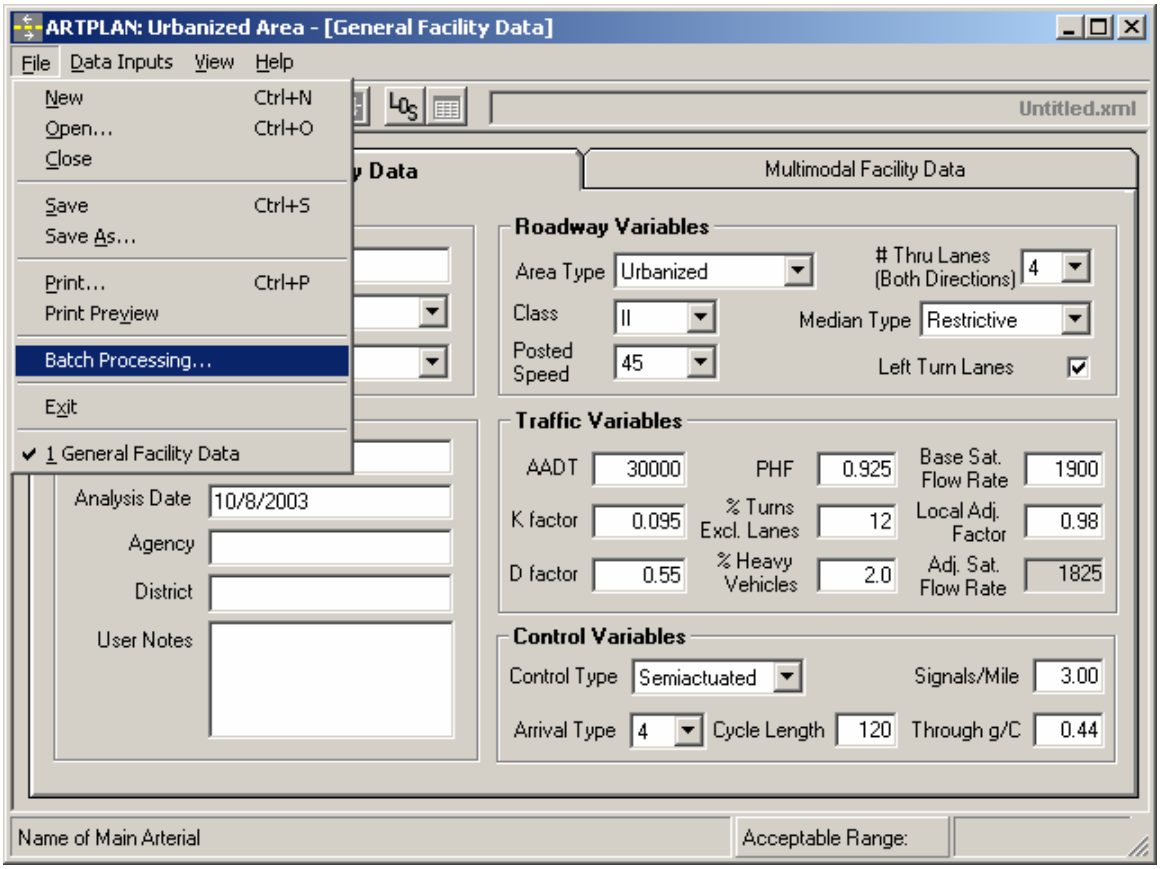


Figure 35. Batch Processing Utility selection from the File menu of ARTPLAN

This action will then launch the Batch Processing utility, as shown in Figure 36 below. LOSPLAN files can be browsed for on connected drives and folders by using the drive and folder list boxes (left side of interface). Files will be displayed according to the program selected (middle of interface). For example, if the ARTPLAN button is selected, files in the selected drive/folder that begin with the 'AP' prefix will be displayed. Likewise, if the HIGHPLAN or FREEPLAN program is selected, files that begin with 'HP' or 'FP' will be displayed, respectively. Figure 37 shows several HIGHPLAN files being displayed. Double-clicking (the left mouse button) on a folder will automatically display files (according to the selected program). Alternatively, the 'Display Files' button can be pressed if a folder was only selected with a single click (the left mouse button).

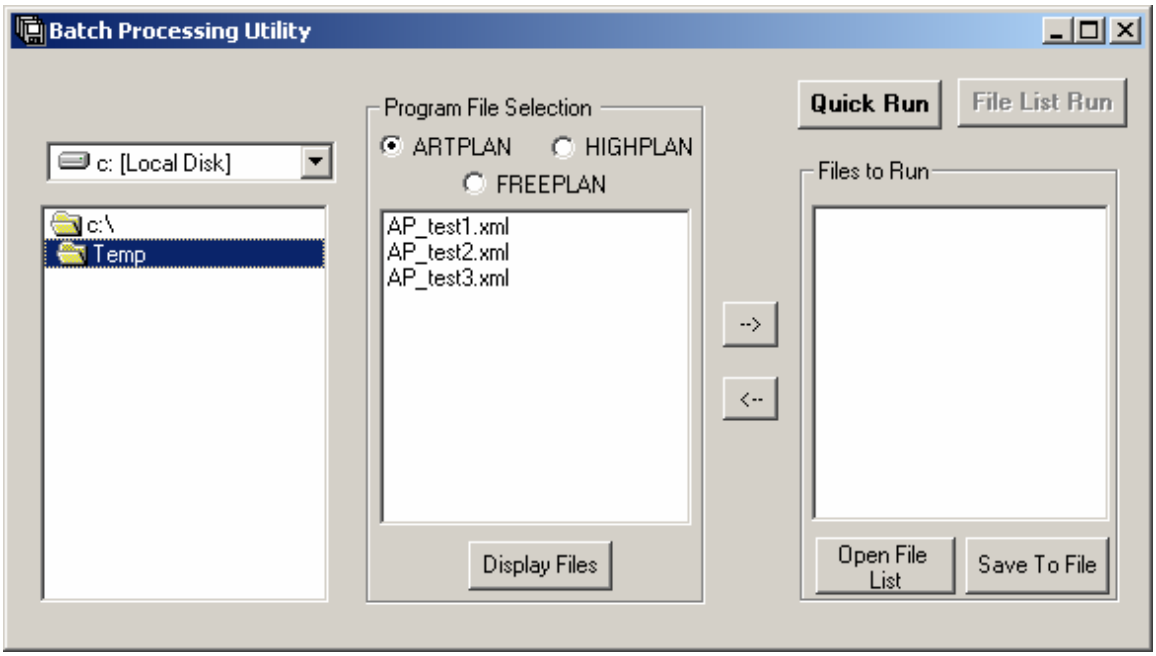


Figure 36. Batch Processing Utility with ARTPLAN Files Displayed

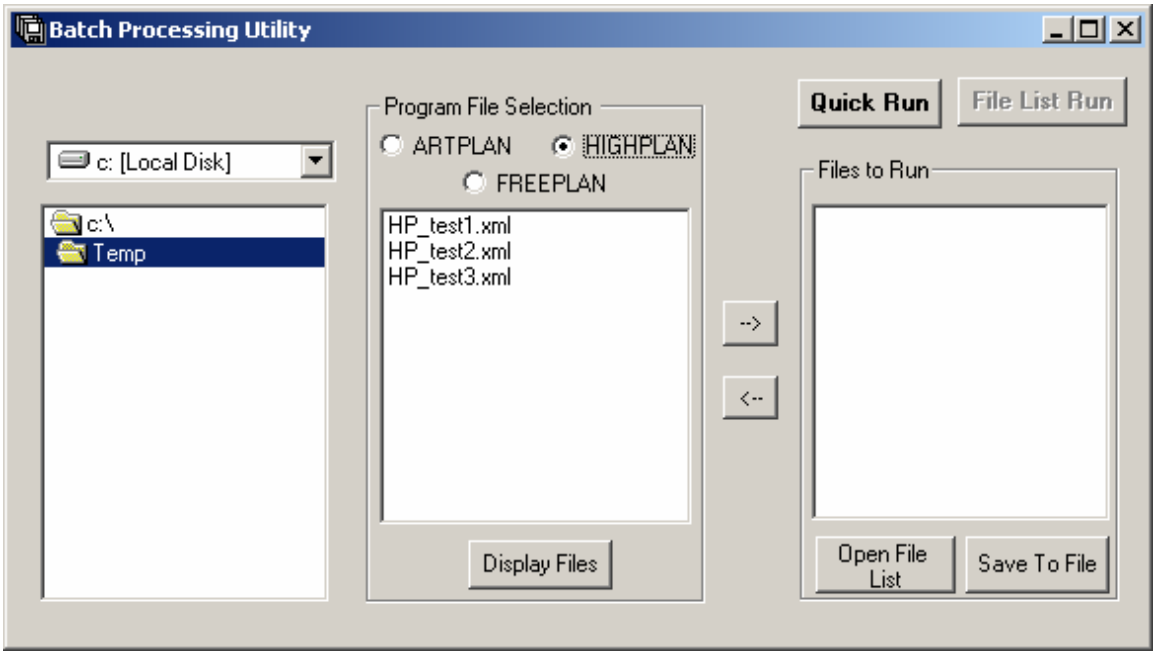


Figure 37. Batch Processing Utility with HIGHPLAN Files Displayed

The following steps describe how to specify the files to be processed:

1. Select the files of interest from the list displayed for the selected drive/folder (middle part of the interface).

   a. Individual files can be selected by just clicking on them with the left mouse button, or the keyboard keys.
   b. Multiple, non-contiguous, files can be selected by using the left mouse button while holding down the 'control' (or 'ctrl') key on the keyboard.
   c. Multiple, contiguous, files can be selected by using the left mouse button while holding down the 'shift' key on the keyboard.
2. Transfer the selected file or files to the 'Files to Run' list box by pressing the arrow button that points to the right. Files can be removed from the 'Files to Run' box by selecting them (this must be done on an individual file basis) and pressing the arrow button that points to the left. Figure 38 shows the selection and transferring of the three listed ARTPLAN files.

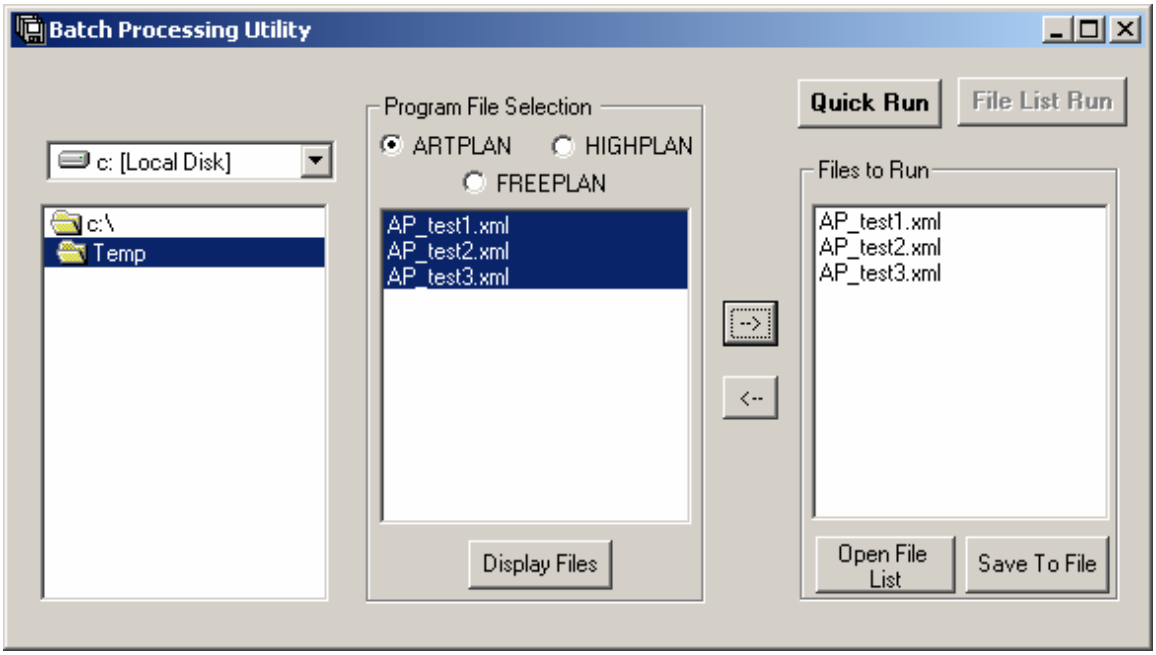


Figure 38. Selection and Transferring of Files to 'Files to Run' List Box.

The last step required before the files can be processed is to save the file list. This is accomplished by pressing the 'Save To File' button. Pressing this button will bring up a file save dialog box as pictured in Figure 39.

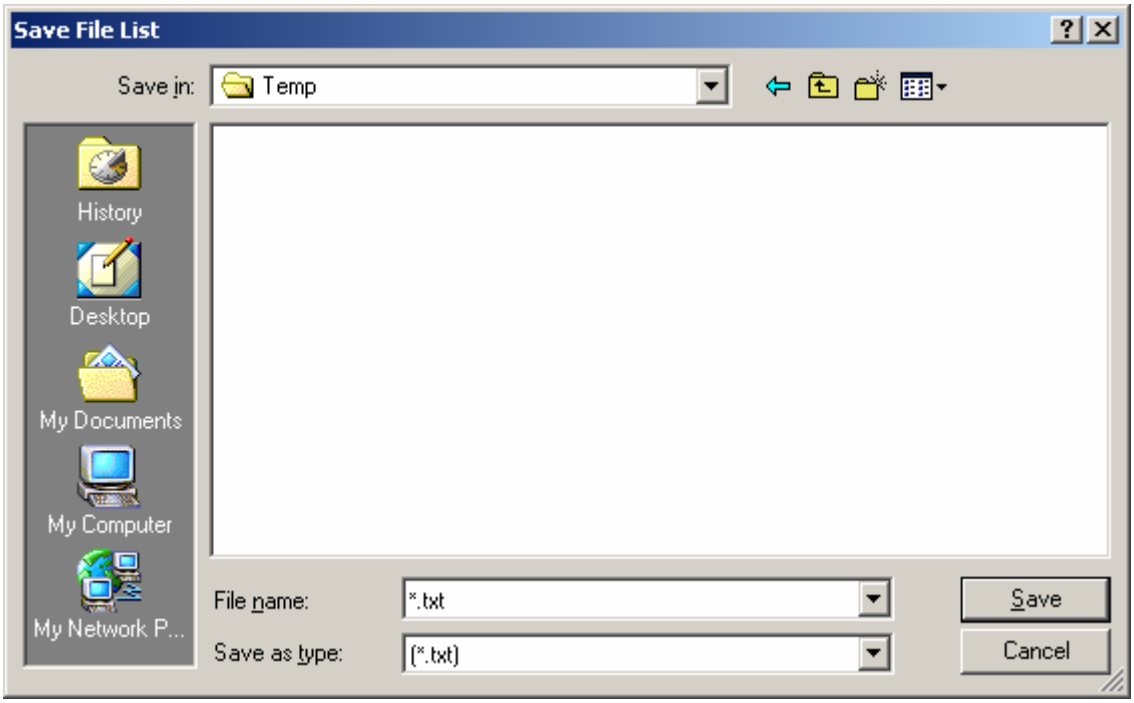


Figure 39. Save Files to Process Dialog Box

Choose the drive/folder location where you would like to save the file, type in any filename of your choice, and then press the 'Save' button. This will save a standard text file, as seen in Figure 40.

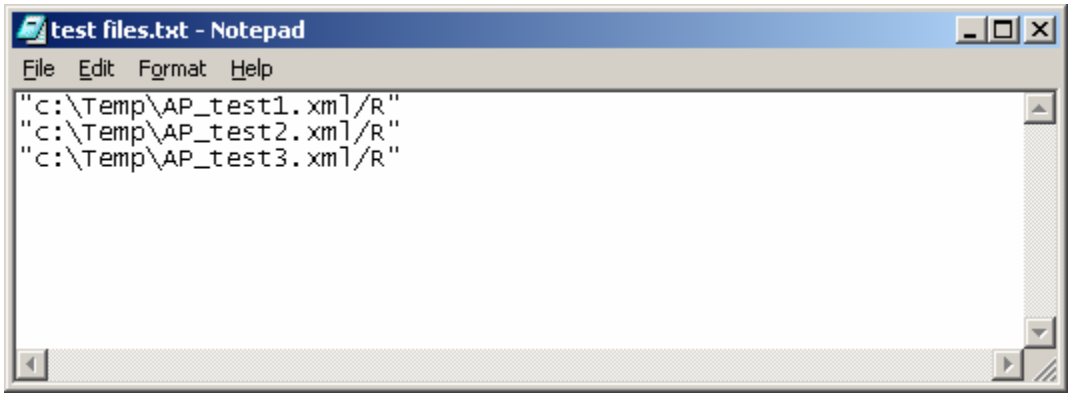


Figure 40. Saved Text File of Files to Process

Once the file list is saved, the 'File List Run' button becomes enabled, as shown in Figure 41. To start the automatic file processing, press this button.

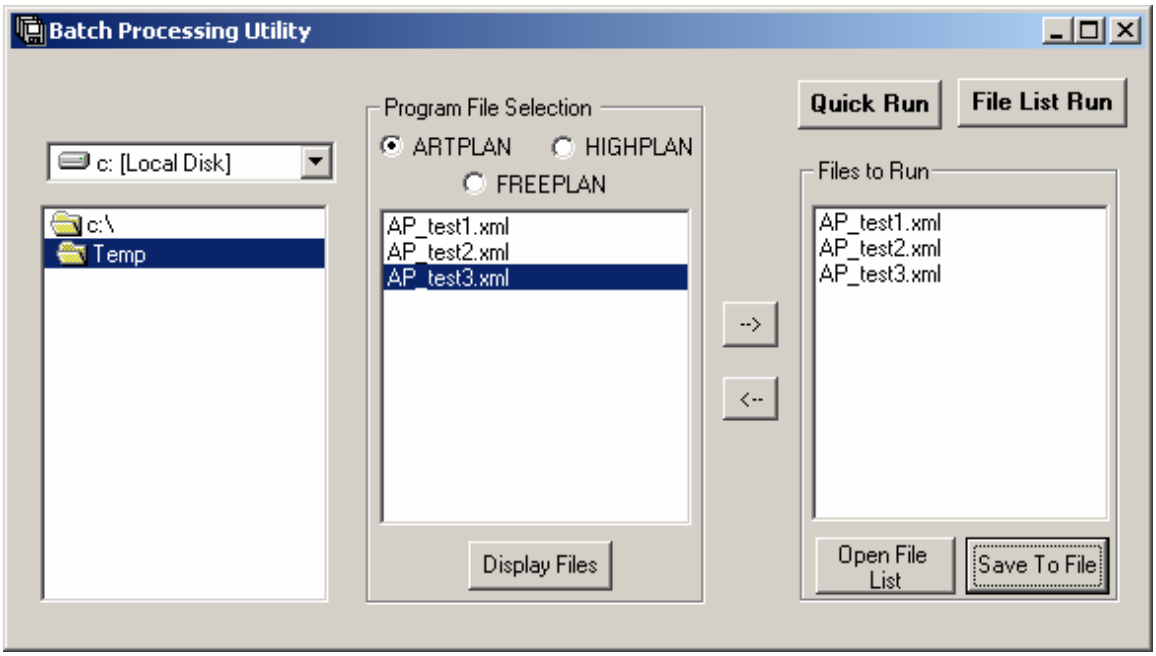


Figure 41. Interface with 'File List Run' Button Enabled after Saving File List

A previously saved file list can also be recalled by pressing the 'Open File List' button. This is illustrated in figures 8 and 9. Again, the 'File List Run' button will become enabled after loading a file list from a previously saved text file.

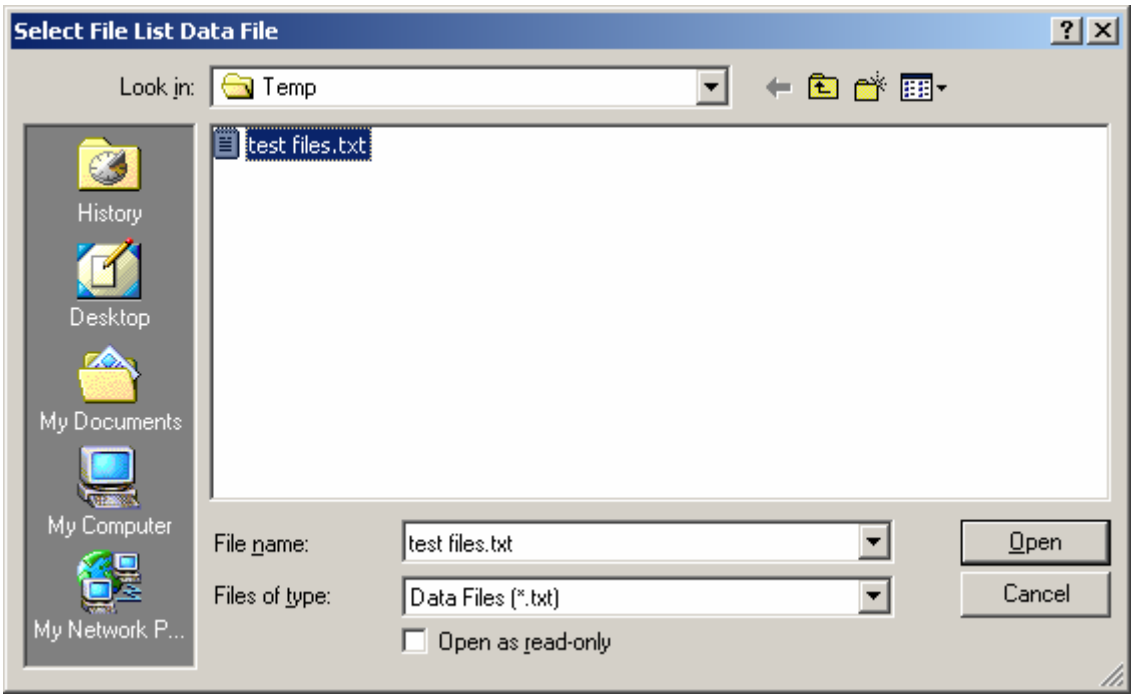


Figure 42. Open File List Dialog Box

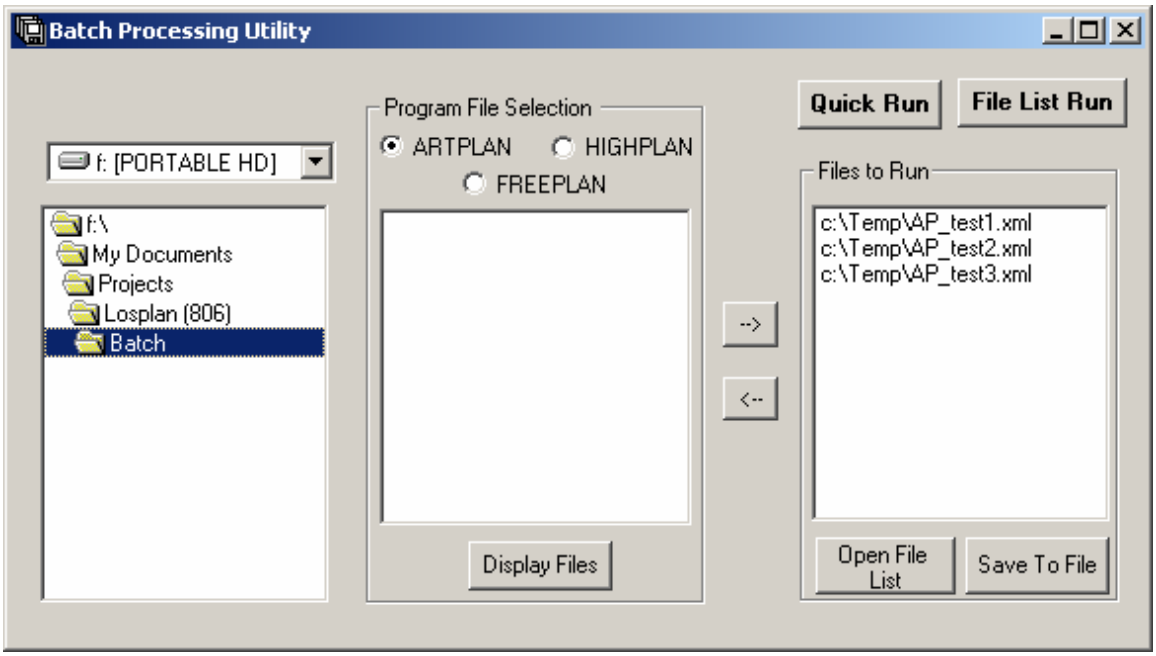


Figure 43. Interface after Loading Filenames from Saved File List

When using this utility, a user will typically want to process multiple files. However, for the instance in which it is desired to process only a single file, the 'Quick Run' feature is also available. Although a single file can be specified for processing in the same manner as for multiple files (i.e., saving to a file list), the 'Quick Run' feature allows a single file to be processed directly, without saving to a file list.

The 'Quick Run' button will process the file that is currently selected in the file list box in the middle part of the interface. It does not reference any files listed in the 'Files to Run' box. For example, in Figure 44, pressing the 'Quick Run' button will process the 'AP_test2.xml' file.

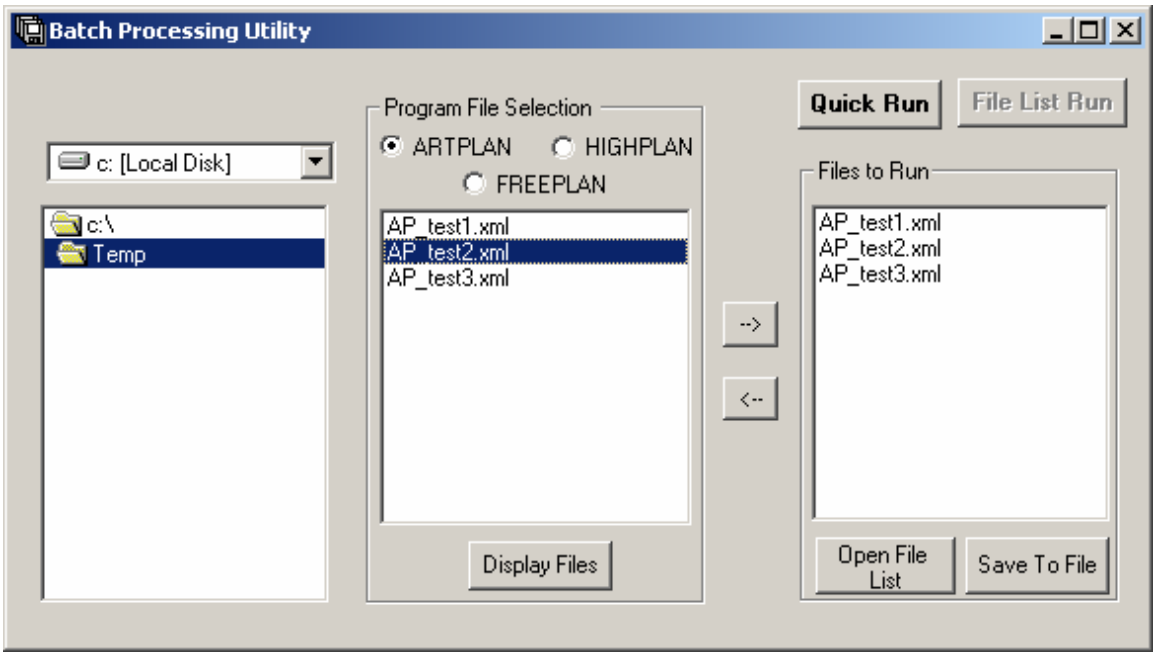


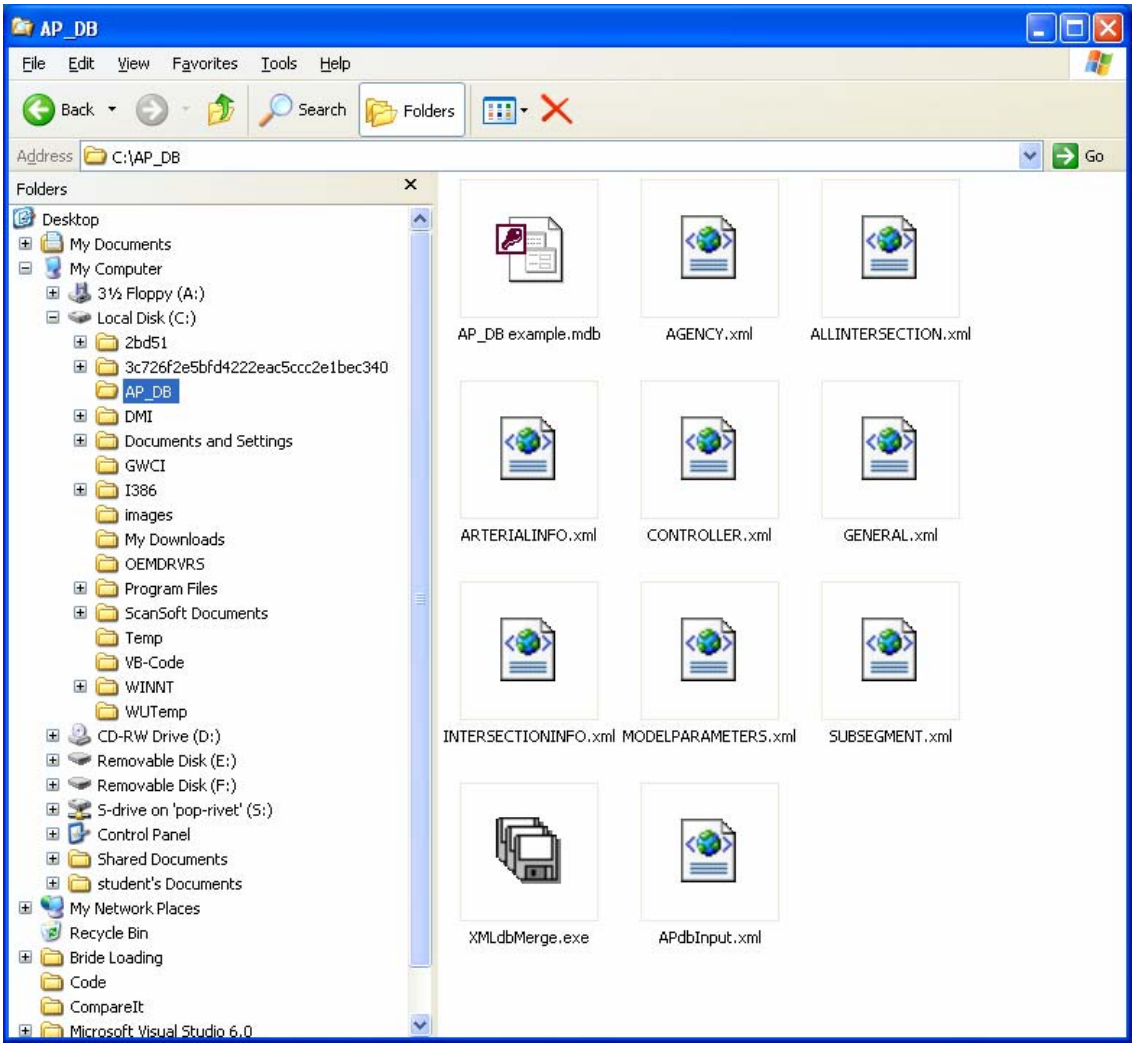
Figure 44. Quick Run Example

APPENDIX D:

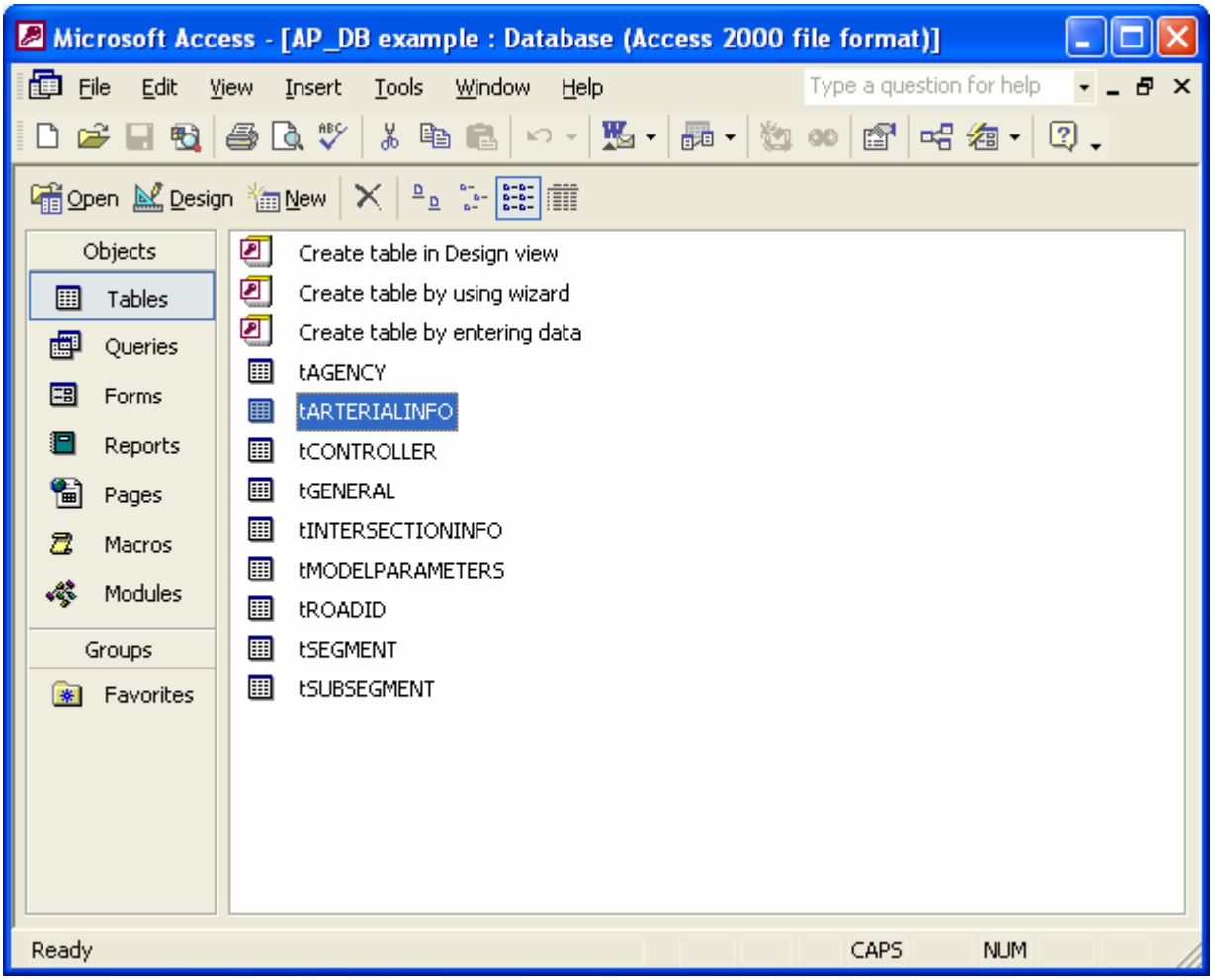
DATABASE INTERFACE EXAMPLE

Database Interfacing with ARTPLAN Example Tutorial

Open the 'AP_DB example.mdb' file, as seen in the Windows Explorer screen shot below.



This brings up the main database view. The 'Tables' view is shown in the figure below. This view shows a listing of all tables defined for the database. As a matter of notational convention, the tables are named with a prefixing 't'.



Double-clicking the name "tARTERIALINFO" shows the following table. Due to the width of the table, it is shown four separate screen shots.

Microsoft Access - [tARTERIALINFO : Table]

RID	ArterialName	FwdDirection	PeriodID	SignalsPerMile	AreaType	ArterialClass_HCM	AADT	KFactor_PLN	DFactor_PLN	PHF
1	Archer Blvd	WestBound	K100	5	Urbanized	4	90000	0.1	0.75	0.8
							0	0	0	0

Record: 2 of 2

Datasheet View

Microsoft Access - [tARTERIALINFO : Table]

BaseSatFlowPerLane	LocalAdjFactor	AdjSatFlowPerLane	HVPct	PctTurnXLane	NumberOfLanes	NumberOfLanesBothDir	PostedSpeed
2000	0.8	1600	3	12	6	6	45
0	0	0	0	0	0	0	0

Record: 1 of 1

Datasheet View

Microsoft Access - [tARTERIALINFO : Table]

NumberOfIntersections	ArrivalType	CycleLength	GCRatio	MedianType	PavementCondition	SidewalkSeparation	OutsideLnWidth	SidewalkBarrier
5	6	90	0	Restrictive	Typical	Typical	Typical	No
0		0	0					

Record: 1 of 1

Datasheet View

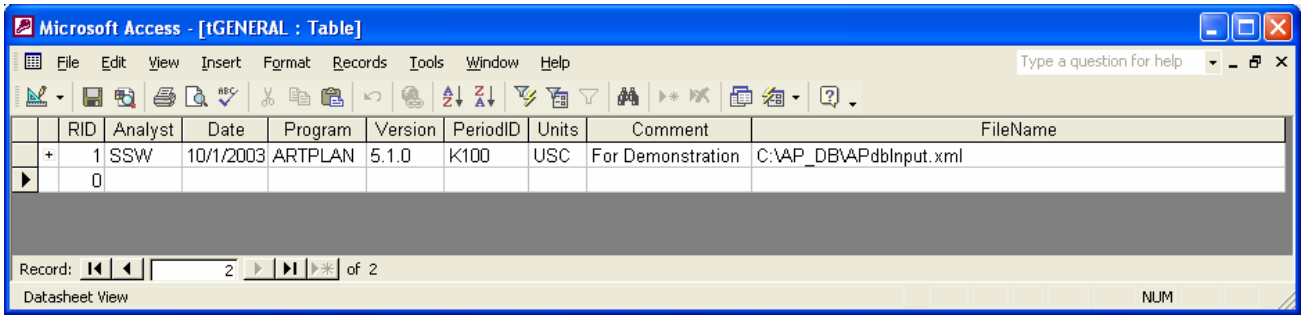
Microsoft Access - [tARTERIALINFO : Table]

SidewalkYN	SidewalkToBusYN	BikeLnYN	NumberOfBuses	SpanOfService
Yes	No	No	0	0
			0	0

Record: 1 of 1

Datasheet View

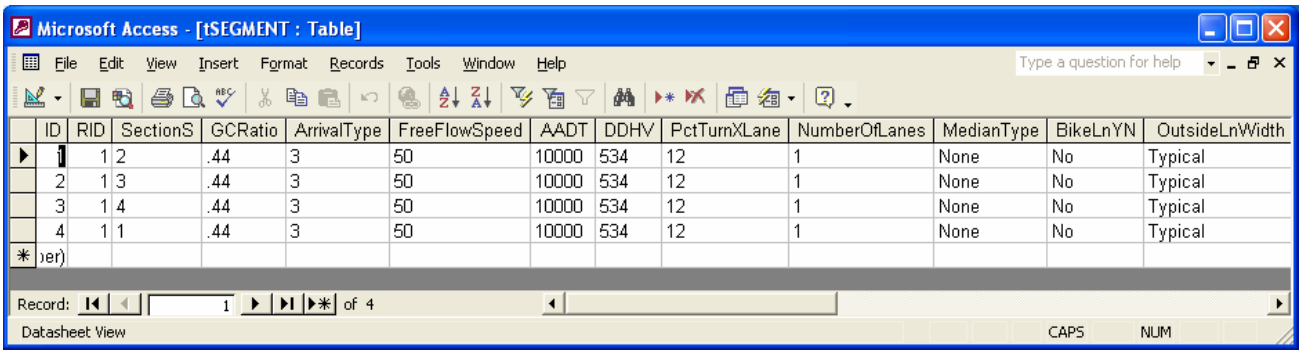
Double-clicking the name “tGENERAL” shows the following table.



Microsoft Access - [tGENERAL : Table]

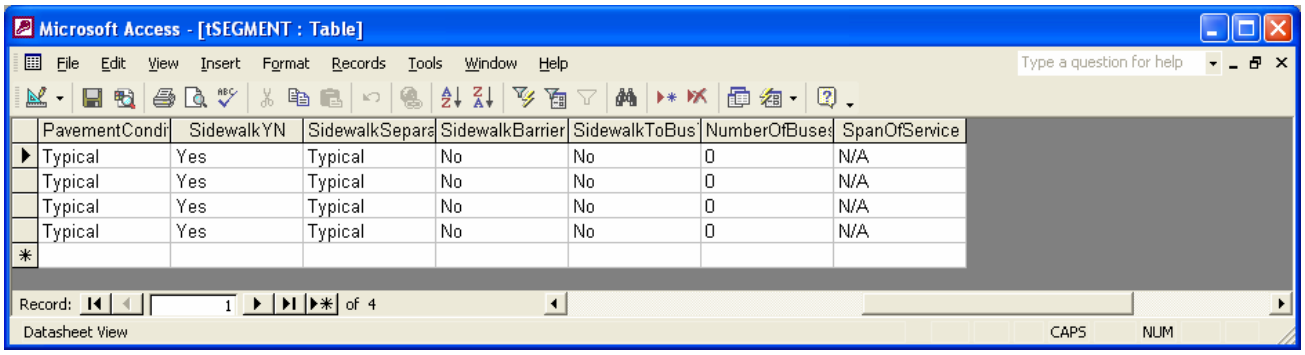
RID	Analyst	Date	Program	Version	PeriodID	Units	Comment	FileName
1	SSW	10/1/2003	ARTPLAN	5.1.0	K100	USC	For Demonstration	C:\AP_DB\APdbInput.xml
0								

Double-clicking the name “tSEGMENT” shows the following table (in two screen shots).



Microsoft Access - [tSEGMENT : Table]

ID	RID	SectionS	GCRatio	ArrivalType	FreeFlowSpeed	AADT	DDHV	PctTurnXLane	NumberOfLanes	MedianType	BikeLnYN	OutsideLnWidth
1	1	2	.44	3	50	10000	534	12	1	None	No	Typical
2	1	3	.44	3	50	10000	534	12	1	None	No	Typical
3	1	4	.44	3	50	10000	534	12	1	None	No	Typical
4	1	1	.44	3	50	10000	534	12	1	None	No	Typical

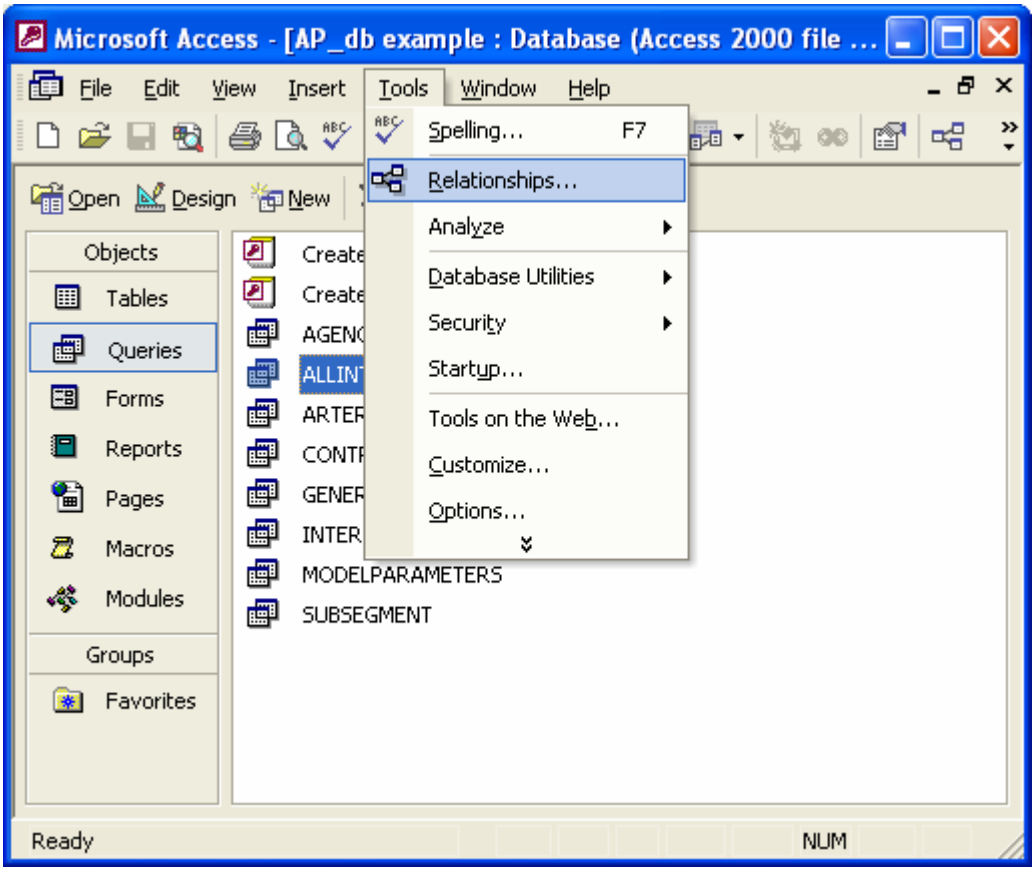


Microsoft Access - [tSEGMENT : Table]

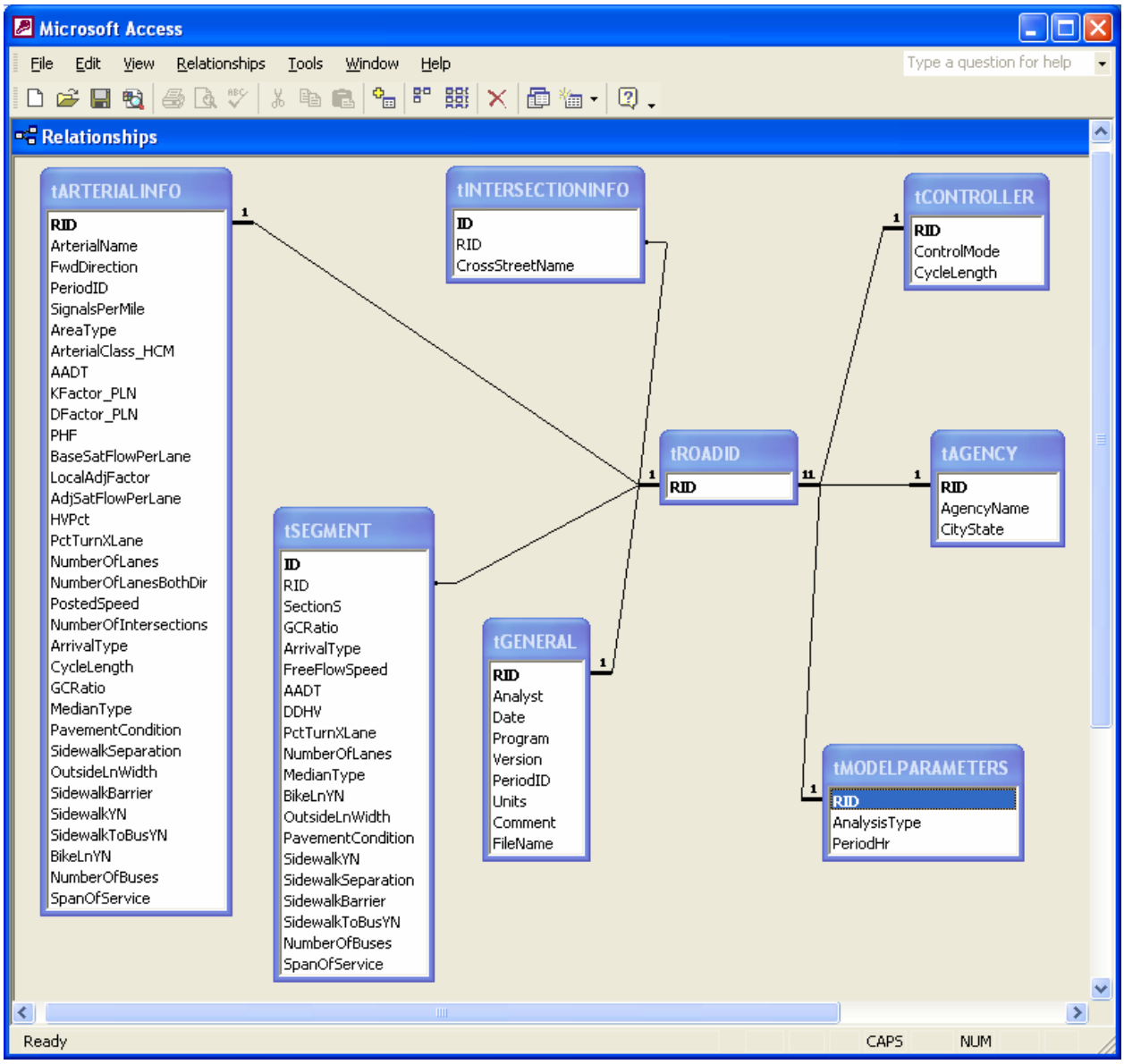
PavementCondi	SidewalkYN	SidewalkSepara	SidewalkBarrier	SidewalkToBus	NumberOfBuses	SpanOfService
Typical	Yes	Typical	No	No	0	N/A
Typical	Yes	Typical	No	No	0	N/A
Typical	Yes	Typical	No	No	0	N/A
Typical	Yes	Typical	No	No	0	N/A

The display of the remaining tables is omitted here, but can be viewed in the same manner as the above tables.

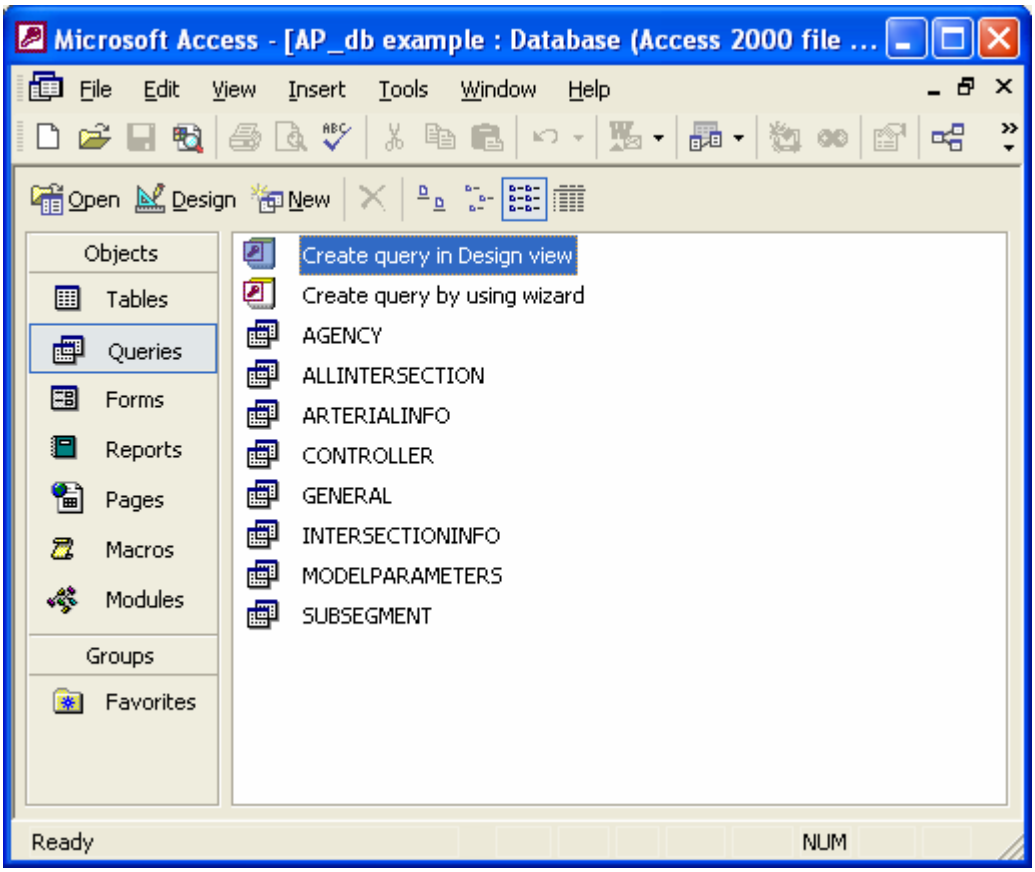
To see how the records contained in the various tables are actually related to one another, select 'Relationships…' from the 'Tools' menu, as shown in the figure below.



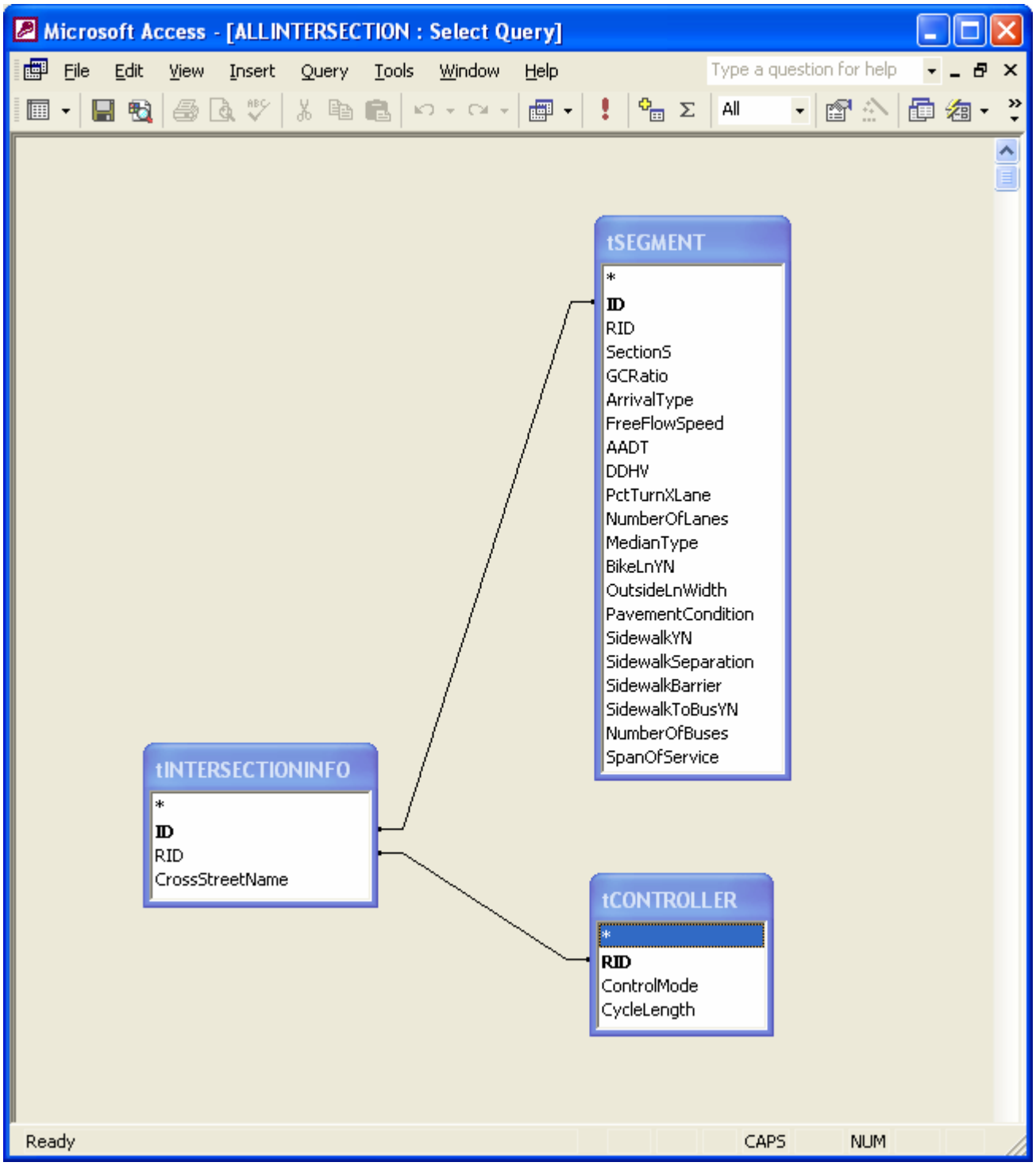
This brings up the 'Relationships' window, as shown below.



Closing this window returns you back to the main database window. Now if we choose the 'Queries' view (left column) instead of 'Tables', a new listing appears, shown in the following figure. Queries provide a way to create new tables based upon a combination of fields established in the main tables.



For example, clicking on the "ALLINTERSECTION" query and choosing to see its design view, the window below will appear, which displays the fields (and corresponding tables) that comprise the query table of "ALLINTERSECTION". Note that the definition of a query only needs to be done once, not every time one wants to export data to an XML file.



The following is the MS-SQL code which gets generated by Access.

```
SELECT      tINTERSECTIONINFO.CrossStreetName,
            tCONTROLLER.ControlMode,
            tCONTROLLER.CycleLength,
            tSEGMENT.SectionS,
            tSEGMENT.GCRatio,
            tSEGMENT.ArrivalType,
            tSEGMENT.FreeFlowSpeed,
            tSEGMENT.AADT,
            tSEGMENT.DDHV,
            tSEGMENT.PctTurnXLane,
            tSEGMENT.NumberOfLanes,
            tSEGMENT.MedianType,
            tSEGMENT.BikeLnYN,
            tSEGMENT.OutsideLnWidth,
            tSEGMENT.PavementCondition,
            tSEGMENT.SidewalkYN,
            tSEGMENT.SidewalkSeparation,
            tSEGMENT.SidewalkBarrier,
            tSEGMENT.SidewalkToBusYN,
            tSEGMENT.NumberOfBuses,
            tSEGMENT.SpanOfService

FROM        (tINTERSECTIONINFO INNER JOIN tCONTROLLER ON
            tINTERSECTIONINFO.RID = tCONTROLLER.RID) INNER JOIN
            tSEGMENT ON tINTERSECTIONINFO.ID = tSEGMENT.ID;
```

Double-clicking the "ALLINTERSECTION" query will display the following table (two screen shots).

Microsoft Access - [ALLINTERSECTION : Select Query]

CrossStreetName	ControlMode	CycleLength	SectionS	GCRatio	ArrivalType	FreeFlowSpeed	AADT	DDHV	PctTurnXLane	NumberOfLanes	MedianType
Second	Pretimed	90	2	.44	3	50	10000	534	12	1	None
Third	Pretimed	90	3	.44	3	50	10000	534	12	1	None
Fourth	Pretimed	90	4	.44	3	50	10000	534	12	1	None
Fifth	Pretimed	90	1	.44	3	50	10000	534	12	1	None

Record: 1 of 4

Datasheet View CAPS NUM

Microsoft Access - [ALLINTERSECTION : Select Query]

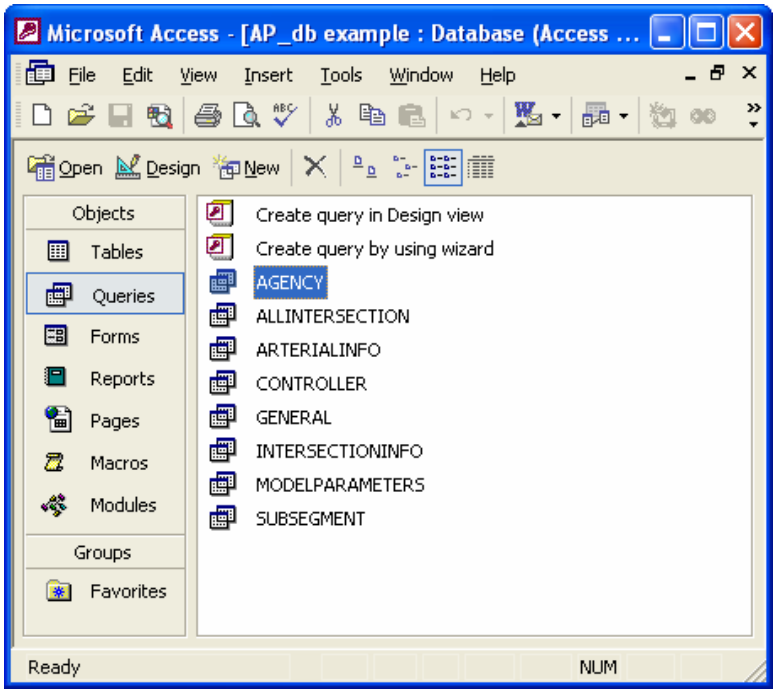
BikeLnYN	OutsideLnWidth	PavementCondition	SidewalkYN	SidewalkSeparation	SidewalkBarrier	SidewalkToBusYN	NumberOfBuses	SpanOfService
No	Typical	Typical	Yes	Typical	No	No	0	N/A
No	Typical	Typical	Yes	Typical	No	No	0	N/A
No	Typical	Typical	Yes	Typical	No	No	0	N/A
No	Typical	Typical	Yes	Typical	No	No	0	N/A

Record: 1 of 4

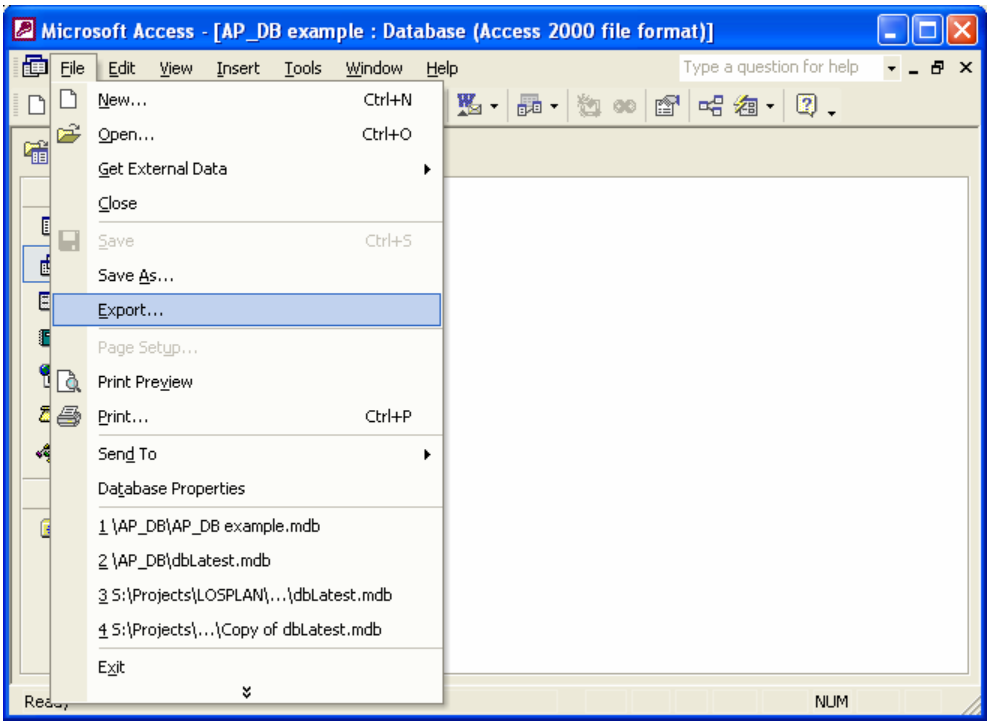
Datasheet View CAPS NUM

The above has been a basic overview of the database structure used in this example. Assuming one has already defined the structure (tables and queries) and populated the fields with data, the following discussion will describe the process for exporting these data to an ARTPLAN compatible input file.

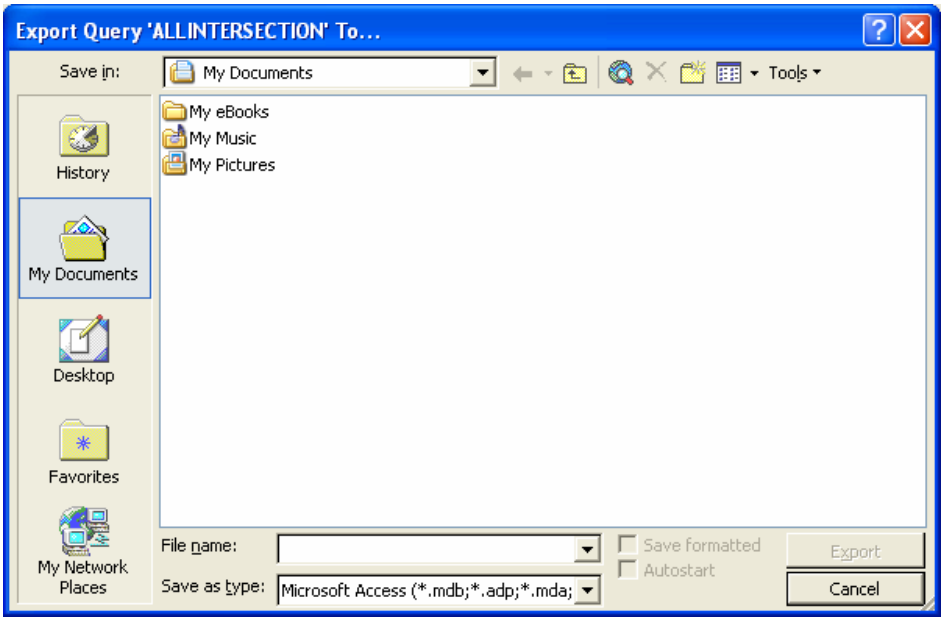
From the ‘Queries’ window, select the ‘Agency’ query.



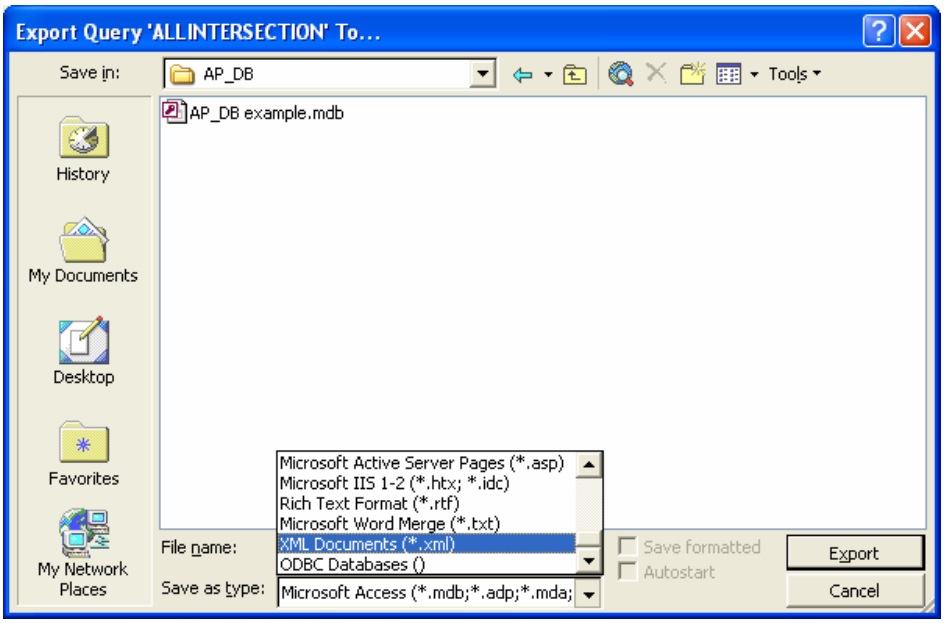
Then select ‘Export…’ from the ‘File’ menu. This can also be selected from a menu shown by clicking the right mouse button.



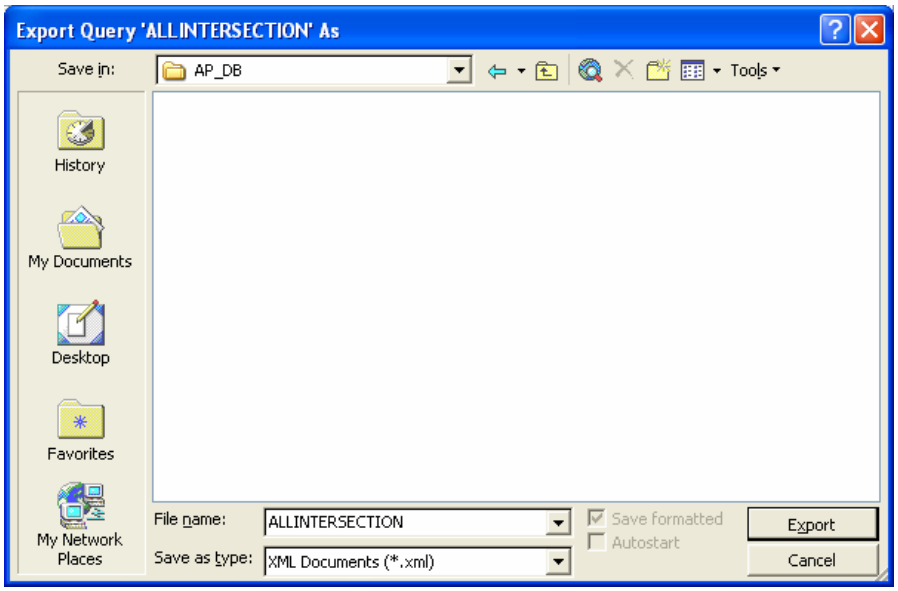
The following dialog box appears.



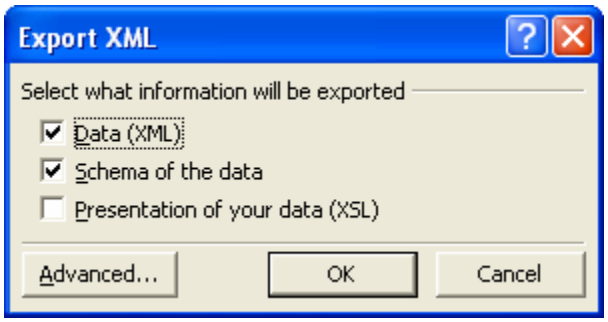
Choose the folder (from the 'Save in:' list box) where you would like to export the query. Type in a filename. In the 'Save as type' list box, choose 'XML Documents (*.xml)'.



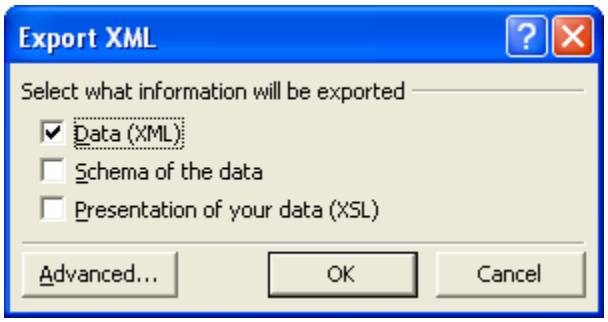
Enter a filename of choice. In this case, we have just left the default filename (same as the name of the query).



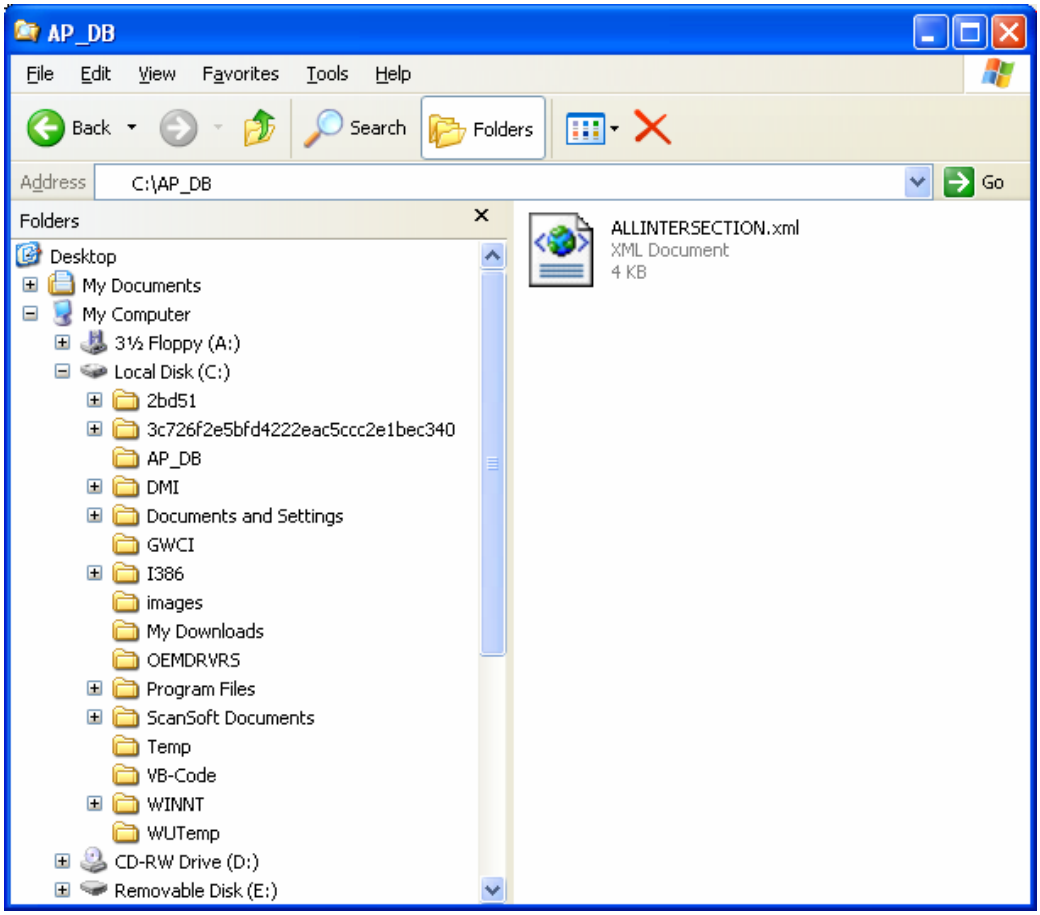
Click the 'Export' button. The following dialog box appears.



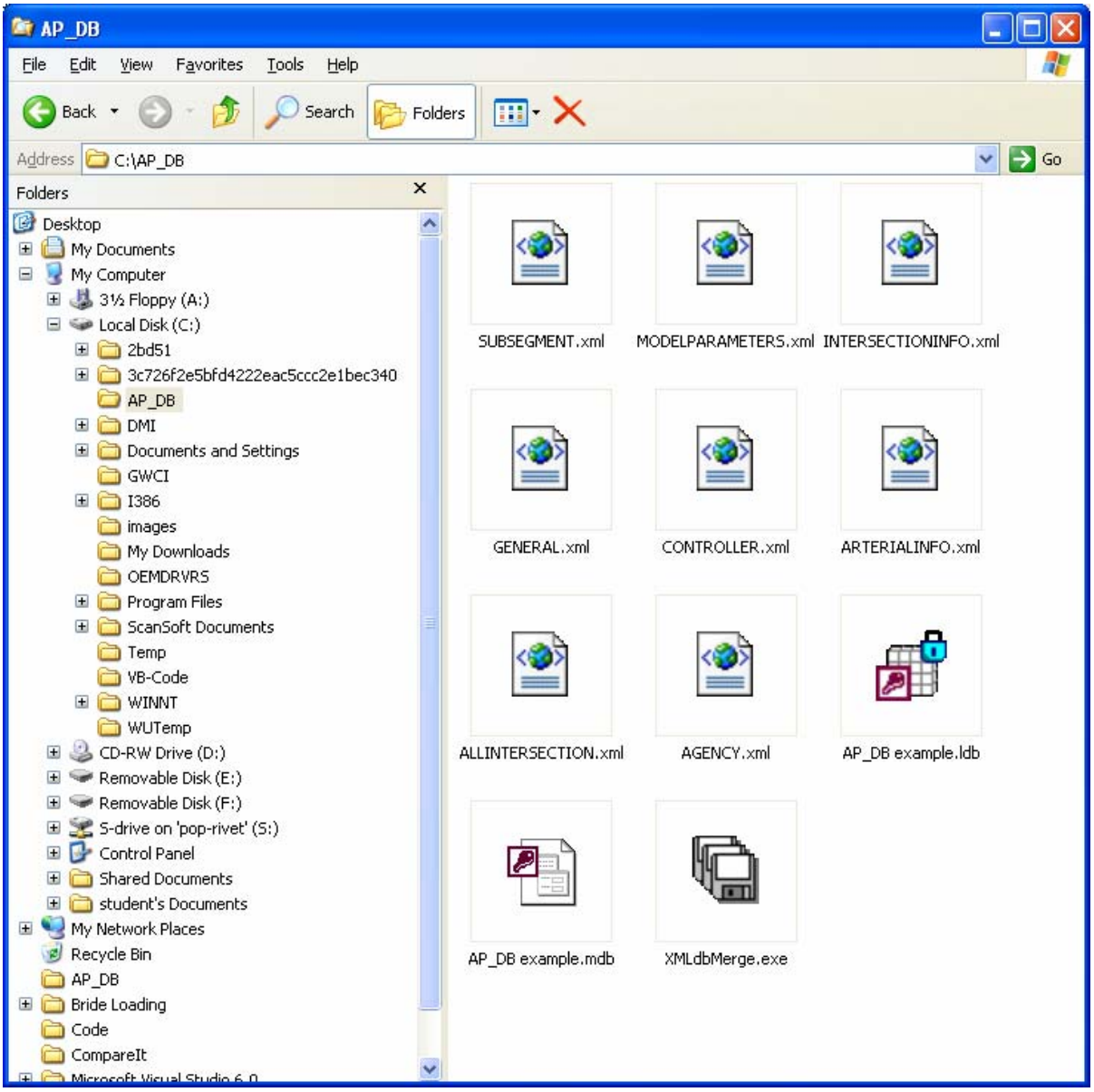
Uncheck "Schema of the data".



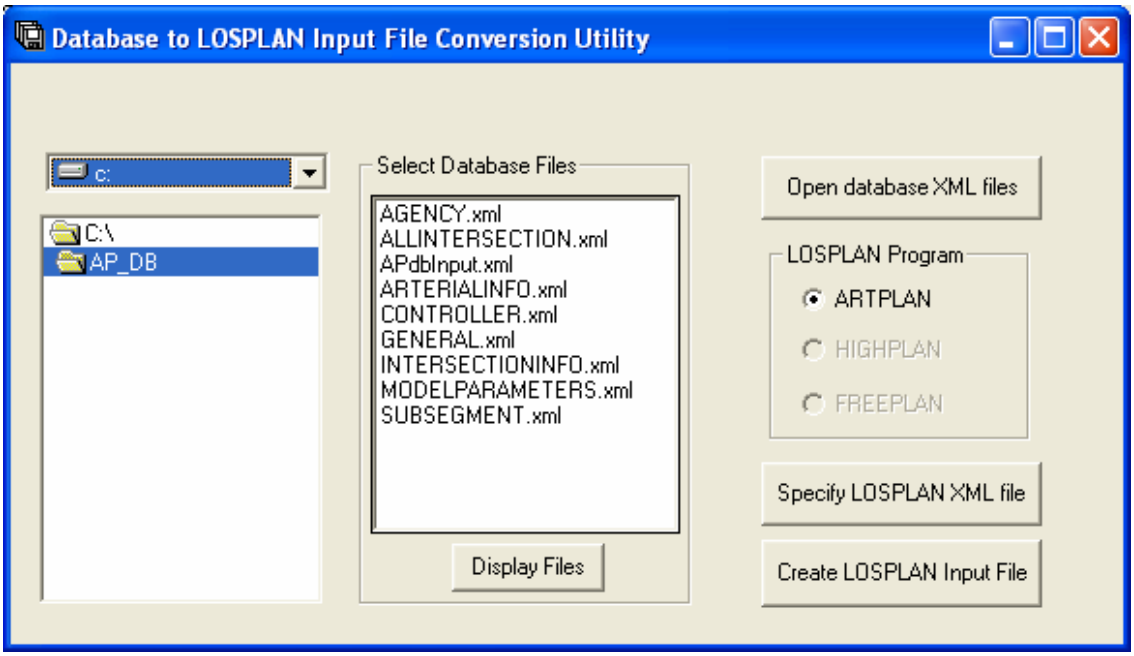
Press the 'OK' button. The file is saved in the indicated folder, as shown below.



Repeat this process for all of the defined queries, shown in the query window. The final result is shown in the following Windows Explorer screen shot. Note that this process can be automated with an Access macro.

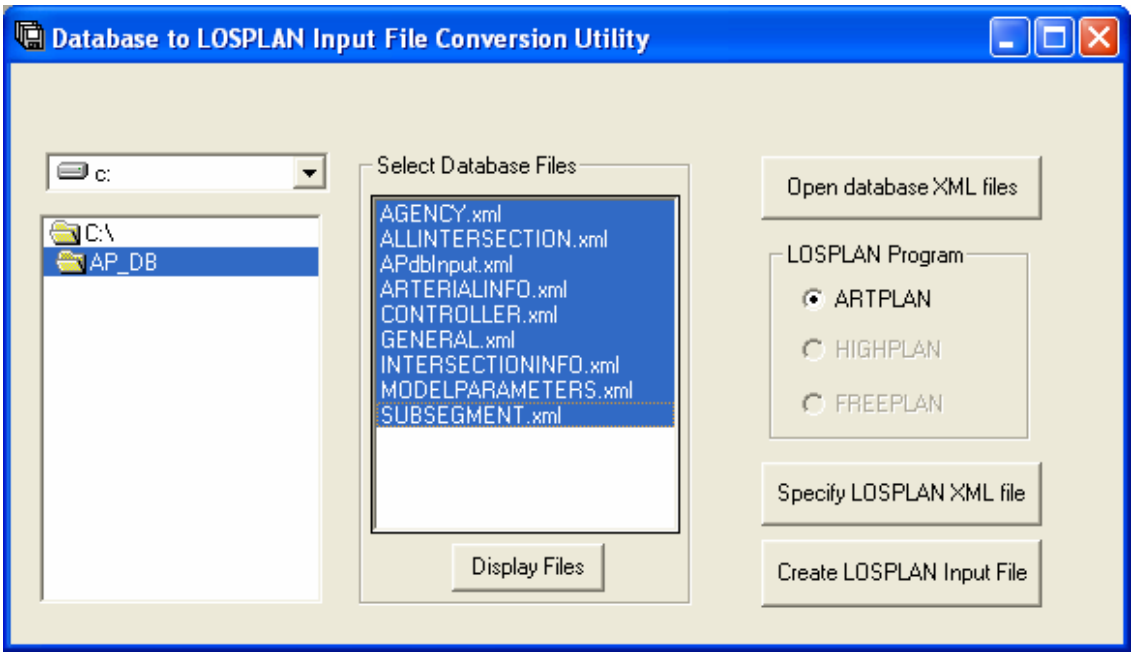

AP_DB
File Edit View Favorites Tools Help
Back Search Folders
Address C:\AP_DB
Go
Folders
Desktop
My Documents
My Computer
3½ Floppy (A:)
Local Disk (C:)
2bd51
3c726f2e5bfd4222eac5ccc2e1bec340
AP_DB
DMI
Documents and Settings
GWCI
I386
images
My Downloads
OEMDRVRS
Program Files
ScanSoft Documents
Temp
VB-Code
WINNT
WUTemp
CD-RW Drive (D:)
Removable Disk (E:)
Removable Disk (F:)
S-drive on 'pop-rivet' (S:)
Control Panel
Shared Documents
student's Documents
My Network Places
Recycle Bin
AP_DB
Bride Loading
Code
CompareIt
SUBSEGMENT.xml
MODELPARAMETERS.xml
INTERSECTIONINFO.xml
GENERAL.xml
CONTROLLER.xml
ARTERIALINFO.xml
ALLINTERSECTION.xml
AGENCY.xml
AP_DB example.ldb
AP_DB example.mdb
XMLdbMerge.exe

Next, run the 'XMLdbMerge.exe' utility (included on the enclosed CD). The main screen for this utility is shown in the following figure.

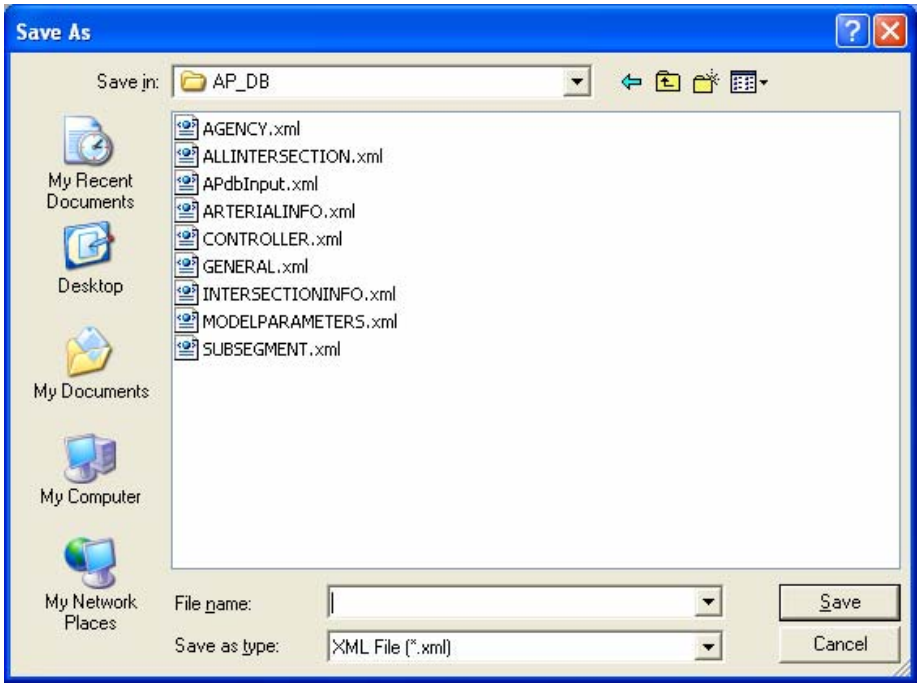


Locate the saved (exported) Access XML files using the drive and folder navigator controls on the left side of the screen. The files contained in the selected drive/folder will be displayed in the list box in the in the center of the screen (as shown above).

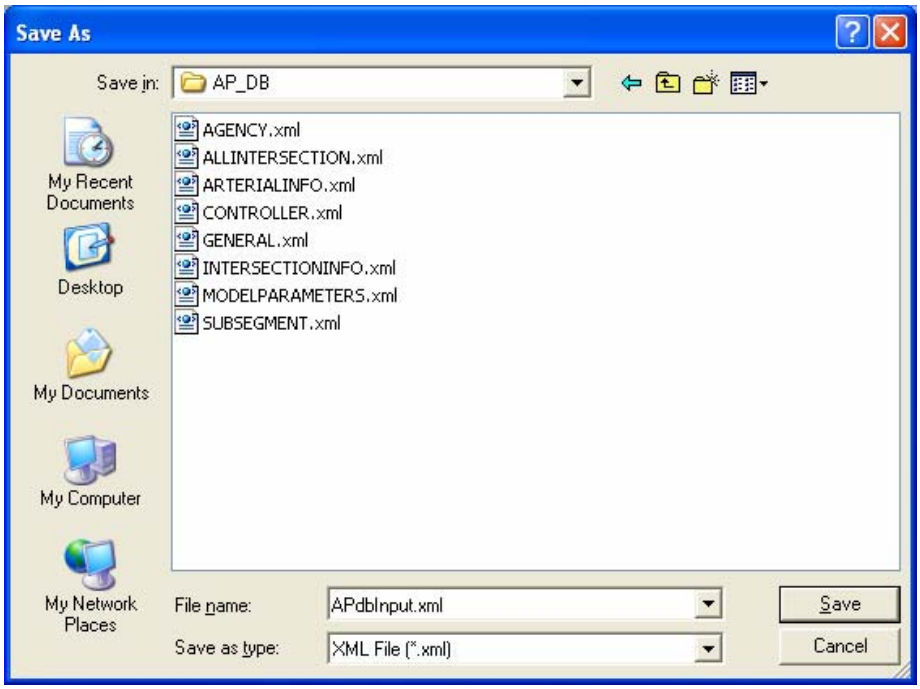
Select all the files, using a combination of the left mouse button and either the 'control' or 'shift' key.



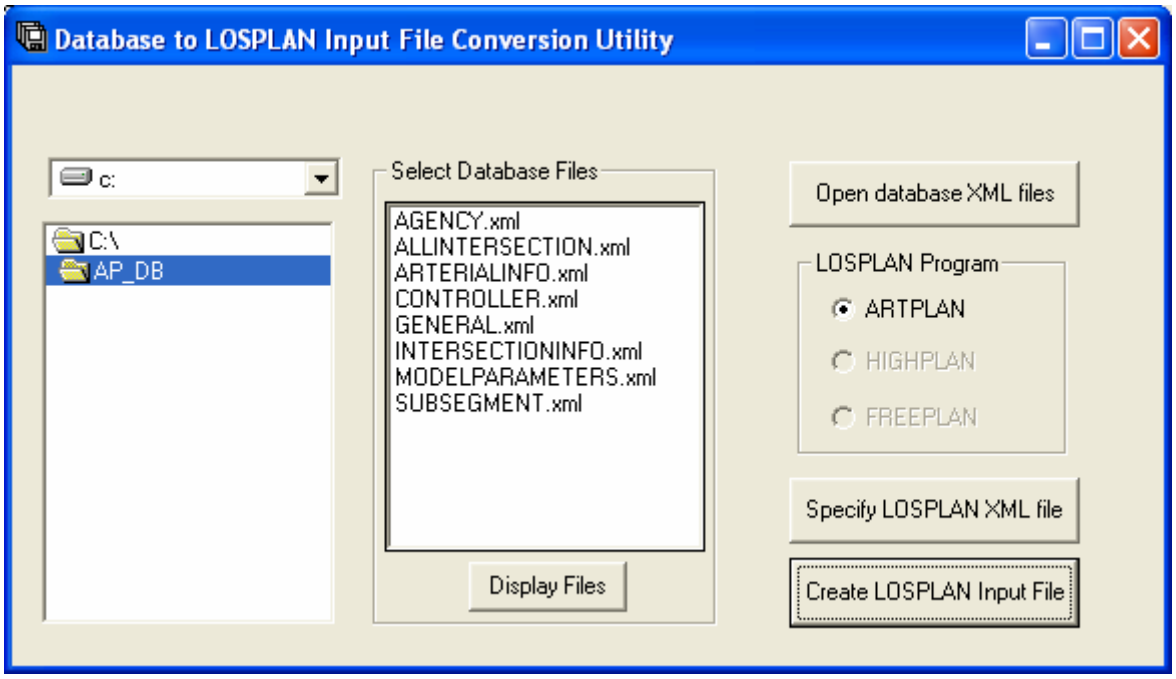
Click on the 'Open database XML files' button. This will create the necessary references to these files. Next, click on the 'Specify LOSPLAN XML file' button. The following dialog box appears.



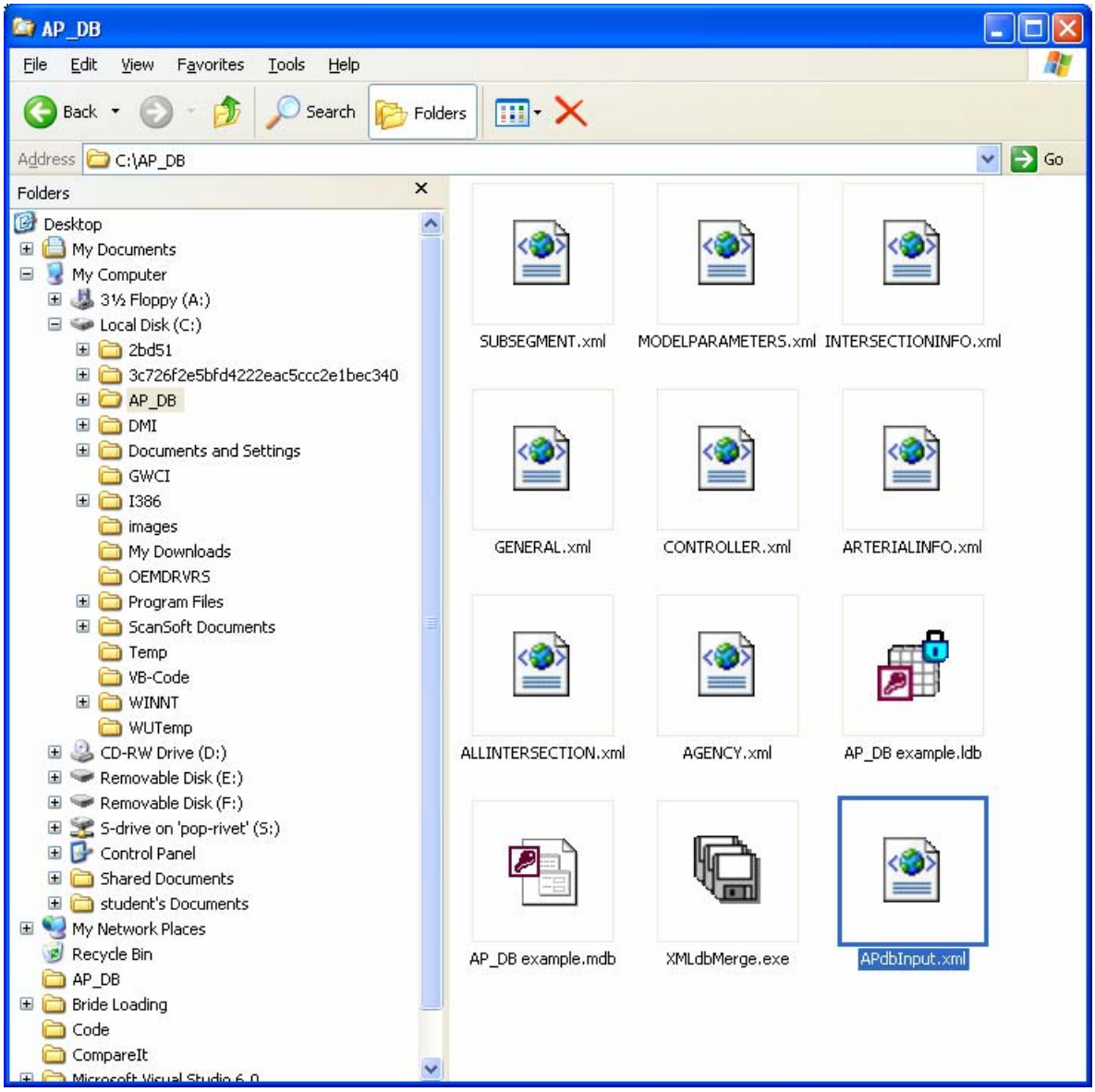
Choose the folder (from the 'Save in:' list box) where you would like to save the converted XML file. Type in a filename. In this example, we have chosen "APdbInput.xml" as the filename.



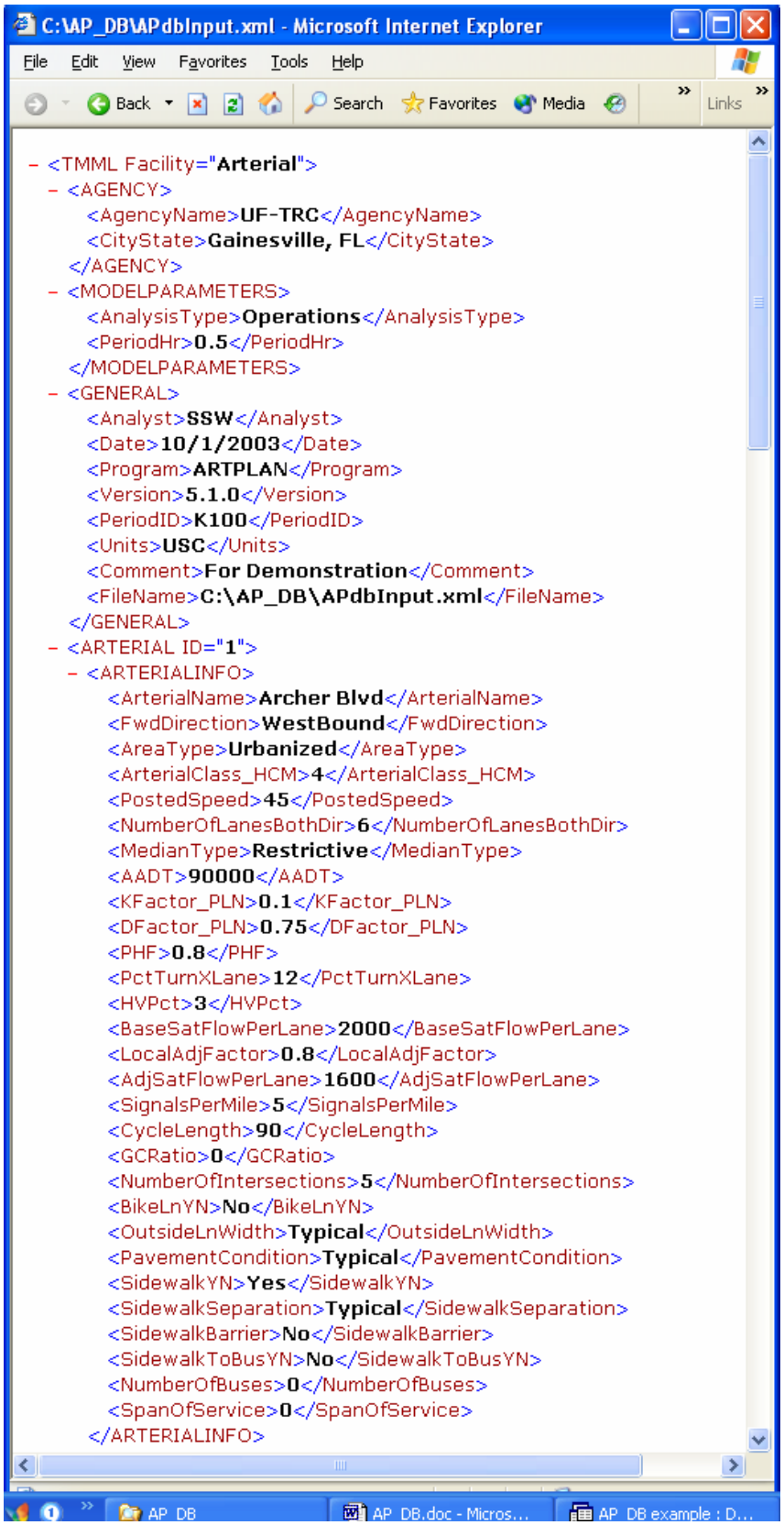
Press the 'Save' button. The main utility screen becomes active again.



Press the 'Create LOSPLAN Input File' button. This will launch the conversion routine. The ARTPLAN compatible input file will now be created, and is shown in the Windows Explorer screen shot (below). The conversion utility can be exited by clicking on the red 'x' in the upper right corner.

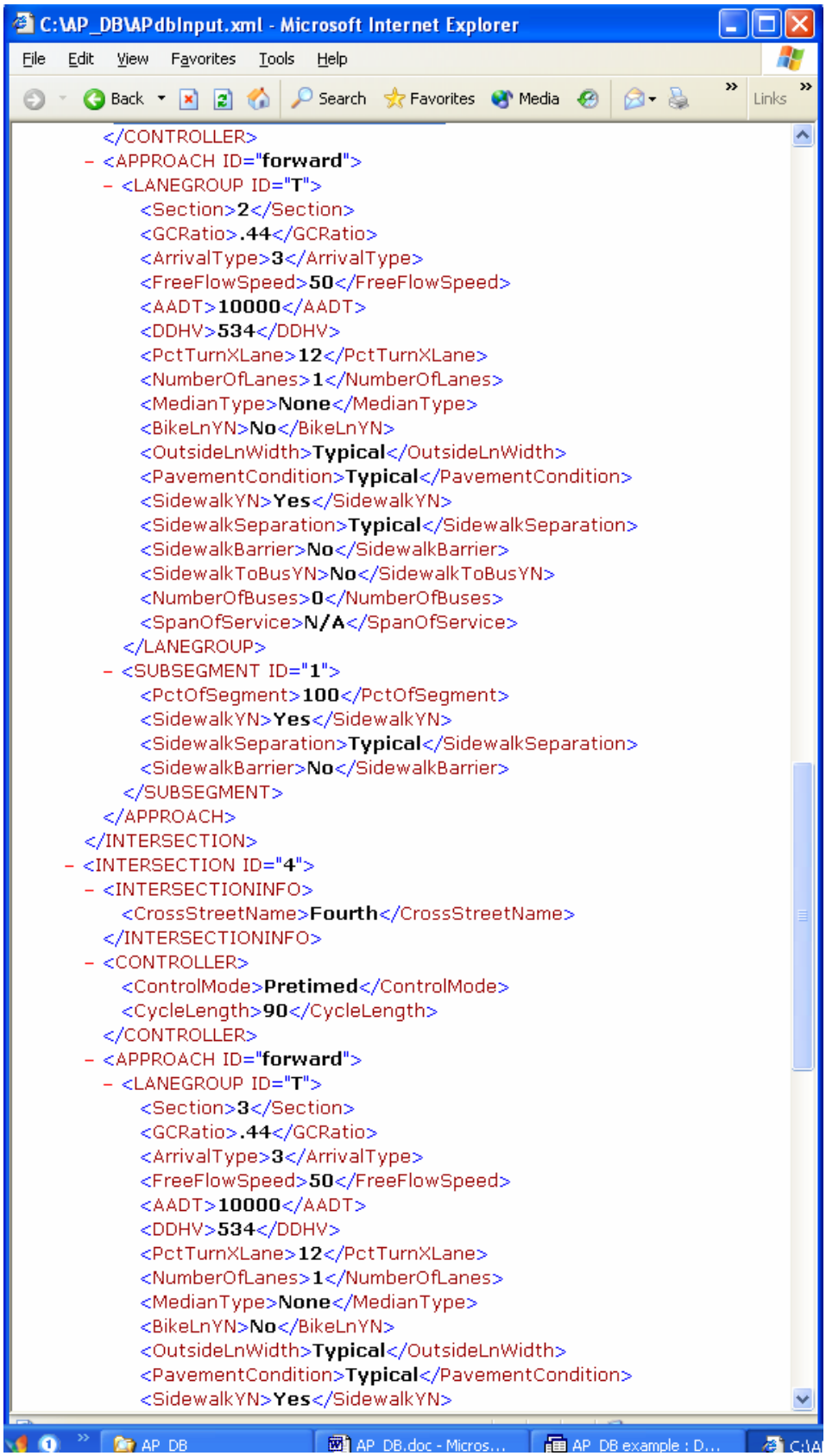


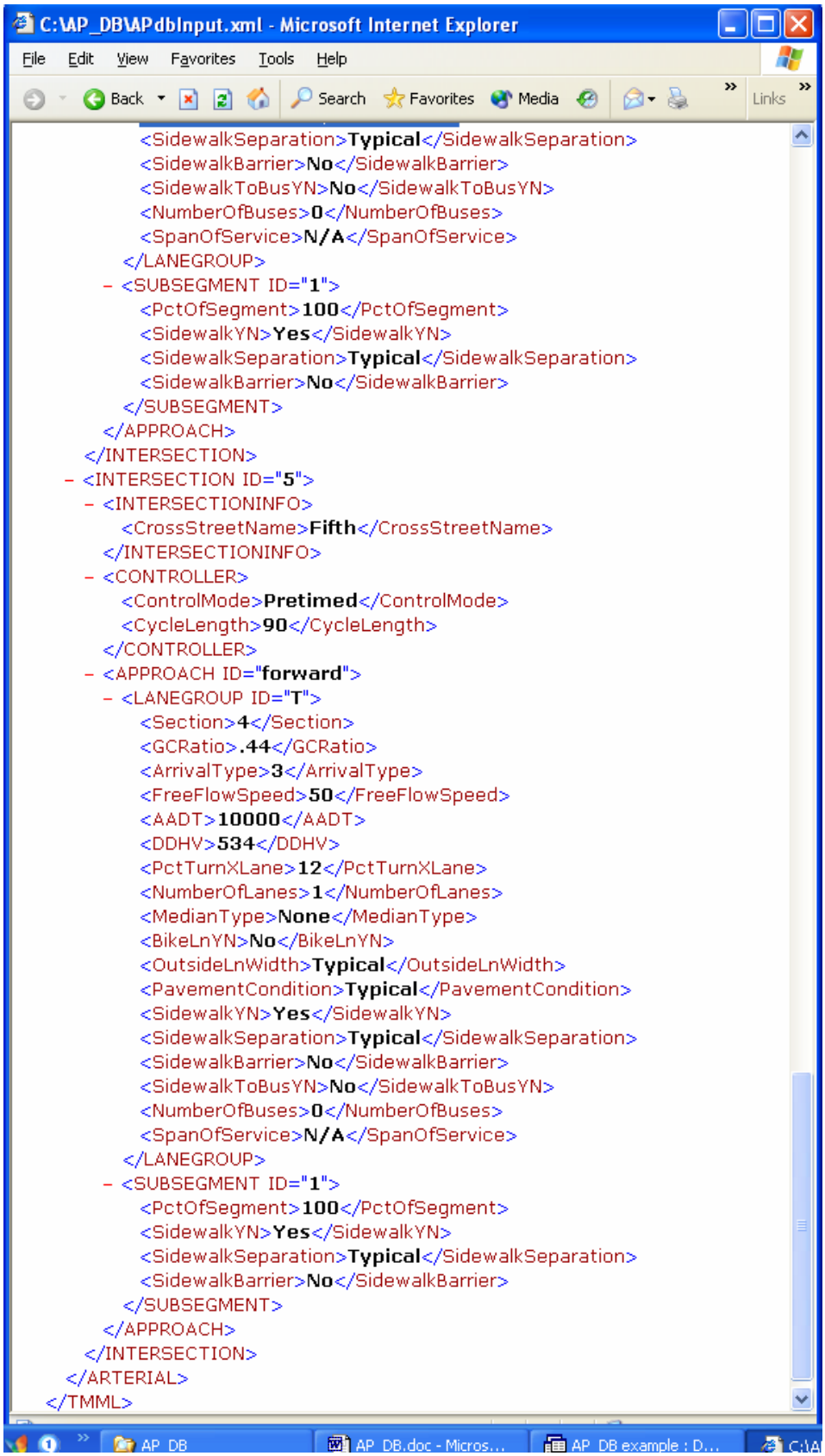
To view the contents of this file, simply double-click the file. It will automatically open in Internet Explorer. The screen shots below show the contents of this file (multiple screen shots due to the length of the file). This file can now be loaded into ARTPLAN by simply starting ARTPLAN, selecting 'Open…' from the 'File' menu, and then specifying this file for loading.


C:\AP_DB\APdbInput.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Links
- <TMML Facility="Arterial">
- <AGENCY>
<AgencyName>UF-TRC</AgencyName>
<CityState>Gainesville, FL</CityState>
</AGENCY>
- <MODELPARAMETERS>
<AnalysisType>Operations</AnalysisType>
<PeriodHr>0.5</PeriodHr>
</MODELPARAMETERS>
- <GENERAL>
<Analyst>SSW</Analyst>
<Date>10/1/2003</Date>
<Program>ARTPLAN</Program>
<Version>5.1.0</Version>
<PeriodID>K100</PeriodID>
<Units>USC</Units>
<Comment>For Demonstration</Comment>
<FileName>C:\AP_DB\APdbInput.xml</FileName>
</GENERAL>
- <ARTERIAL ID="1">
- <ARTERIALINFO>
<ArterialName>Archer Blvd</ArterialName>
<FwdDirection>WestBound</FwdDirection>
<AreaType>Urbanized</AreaType>
<ArterialClass_HCM>4</ArterialClass_HCM>
<PostedSpeed>45</PostedSpeed>
<NumberOfLanesBothDir>6</NumberOfLanesBothDir>
<MedianType>Restrictive</MedianType>
<AADT>90000</AADT>
<KFactor_PLN>0.1</KFactor_PLN>
<DFactor_PLN>0.75</DFactor_PLN>
<PHF>0.8</PHF>
<PctTurnXLane>12</PctTurnXLane>
<HVPct>3</HVPct>
<BaseSatFlowPerLane>2000</BaseSatFlowPerLane>
<LocalAdjFactor>0.8</LocalAdjFactor>
<AdjSatFlowPerLane>1600</AdjSatFlowPerLane>
<SignalsPerMile>5</SignalsPerMile>
<CycleLength>90</CycleLength>
<GCRatio>0</GCRatio>
<NumberOfIntersections>5</NumberOfIntersections>
<BikeLnYN>No</BikeLnYN>
<OutsideLnWidth>Typical</OutsideLnWidth>
<PavementCondition>Typical</PavementCondition>
<SidewalkYN>Yes</SidewalkYN>
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
<SidewalkToBusYN>No</SidewalkToBusYN>
<NumberOfBuses>0</NumberOfBuses>
<SpanOfService>0</SpanOfService>
</ARTERIALINFO>
AP_DB
AP_DB.doc - Micros...
AP_DB example : D...

C:\AP_DB\APdbInput.xml - Microsoft Internet Explorer


```xml
- <INTERSECTION ID="1">
  - <INTERSECTIONINFO>
      <CrossStreetName>First</CrossStreetName>
    </INTERSECTIONINFO>
  - <CONTROLLER>
      <ControlMode>Pretimed</ControlMode>
      <CycleLength>90</CycleLength>
    </CONTROLLER>
  </INTERSECTION>
- <INTERSECTION ID="2">
  - <INTERSECTIONINFO>
      <CrossStreetName>Second</CrossStreetName>
    </INTERSECTIONINFO>
  - <CONTROLLER>
      <ControlMode>Pretimed</ControlMode>
      <CycleLength>90</CycleLength>
    </CONTROLLER>
  - <APPROACH ID="forward">
    - <LANEGROUP ID="T">
        <Section>1</Section>
        <GCRatio>.44</GCRatio>
        <ArrivalType>3</ArrivalType>
        <FreeFlowSpeed>50</FreeFlowSpeed>
        <AADT>10000</AADT>
        <DDHV>534</DDHV>
        <PctTurnXLane>12</PctTurnXLane>
        <NumberOfLanes>1</NumberOfLanes>
        <MedianType>None</MedianType>
        <BikeLnYN>No</BikeLnYN>
        <OutsideLnWidth>Typical</OutsideLnWidth>
        <PavementCondition>Typical</PavementCondition>
        <SidewalkYN>Yes</SidewalkYN>
        <SidewalkSeparation>Typical</SidewalkSeparation>
        <SidewalkBarrier>No</SidewalkBarrier>
        <SidewalkToBusYN>No</SidewalkToBusYN>
        <NumberOfBuses>0</NumberOfBuses>
        <SpanOfService>N/A</SpanOfService>
      </LANEGROUP>
    - <SUBSEGMENT ID="1">
        <PctOfSegment>100</PctOfSegment>
        <SidewalkYN>Yes</SidewalkYN>
        <SidewalkSeparation>Typical</SidewalkSeparation>
        <SidewalkBarrier>No</SidewalkBarrier>
      </SUBSEGMENT>
    </APPROACH>
  </INTERSECTION>
- <INTERSECTION ID="3">
  - <INTERSECTIONINFO>
      <CrossStreetName>Third</CrossStreetName>
    </INTERSECTIONINFO>
  - <CONTROLLER>
      <ControlMode>Pretimed</ControlMode>
      <CycleLength>90</CycleLength>
```


C:\AP_DB\APdbInput.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Links
</CONTROLLER>
- <APPROACH ID="forward">
- <LANEGROUP ID="T">
<Section>2</Section>
<GCRatio>.44</GCRatio>
<ArrivalType>3</ArrivalType>
<FreeFlowSpeed>50</FreeFlowSpeed>
<AADT>10000</AADT>
<DDHV>534</DDHV>
<PctTurnXLane>12</PctTurnXLane>
<NumberOfLanes>1</NumberOfLanes>
<MedianType>None</MedianType>
<BikeLnYN>No</BikeLnYN>
<OutsideLnWidth>Typical</OutsideLnWidth>
<PavementCondition>Typical</PavementCondition>
<SidewalkYN>Yes</SidewalkYN>
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
<SidewalkToBusYN>No</SidewalkToBusYN>
<NumberOfBuses>0</NumberOfBuses>
<SpanOfService>N/A</SpanOfService>
</LANEGROUP>
- <SUBSEGMENT ID="1">
<PctOfSegment>100</PctOfSegment>
<SidewalkYN>Yes</SidewalkYN>
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
</SUBSEGMENT>
</APPROACH>
</INTERSECTION>
- <INTERSECTION ID="4">
- <INTERSECTIONINFO>
<CrossStreetName>Fourth</CrossStreetName>
</INTERSECTIONINFO>
- <CONTROLLER>
<ControlMode>Pretimed</ControlMode>
<CycleLength>90</CycleLength>
</CONTROLLER>
- <APPROACH ID="forward">
- <LANEGROUP ID="T">
<Section>3</Section>
<GCRatio>.44</GCRatio>
<ArrivalType>3</ArrivalType>
<FreeFlowSpeed>50</FreeFlowSpeed>
<AADT>10000</AADT>
<DDHV>534</DDHV>
<PctTurnXLane>12</PctTurnXLane>
<NumberOfLanes>1</NumberOfLanes>
<MedianType>None</MedianType>
<BikeLnYN>No</BikeLnYN>
<OutsideLnWidth>Typical</OutsideLnWidth>
<PavementCondition>Typical</PavementCondition>
<SidewalkYN>Yes</SidewalkYN>
AP_DB
AP_DB.doc - Micros...
AP_DB.example : D...


C:\AP_DB\APdbInput.xml - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media Links
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
<SidewalkToBusYN>No</SidewalkToBusYN>
<NumberOfBuses>0</NumberOfBuses>
<SpanOfService>N/A</SpanOfService>
</LANEGROUP>
- <SUBSEGMENT ID="1">
<PctOfSegment>100</PctOfSegment>
<SidewalkYN>Yes</SidewalkYN>
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
</SUBSEGMENT>
</APPROACH>
</INTERSECTION>
- <INTERSECTION ID="5">
- <INTERSECTIONINFO>
<CrossStreetName>Fifth</CrossStreetName>
</INTERSECTIONINFO>
- <CONTROLLER>
<ControlMode>Pretimed</ControlMode>
<CycleLength>90</CycleLength>
</CONTROLLER>
- <APPROACH ID="forward">
- <LANEGROUP ID="T">
<Section>4</Section>
<GCRatio>.44</GCRatio>
<ArrivalType>3</ArrivalType>
<FreeFlowSpeed>50</FreeFlowSpeed>
<AADT>10000</AADT>
<DDHV>534</DDHV>
<PctTurnXLane>12</PctTurnXLane>
<NumberOfLanes>1</NumberOfLanes>
<MedianType>None</MedianType>
<BikeLnYN>No</BikeLnYN>
<OutsideLnWidth>Typical</OutsideLnWidth>
<PavementCondition>Typical</PavementCondition>
<SidewalkYN>Yes</SidewalkYN>
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
<SidewalkToBusYN>No</SidewalkToBusYN>
<NumberOfBuses>0</NumberOfBuses>
<SpanOfService>N/A</SpanOfService>
</LANEGROUP>
- <SUBSEGMENT ID="1">
<PctOfSegment>100</PctOfSegment>
<SidewalkYN>Yes</SidewalkYN>
<SidewalkSeparation>Typical</SidewalkSeparation>
<SidewalkBarrier>No</SidewalkBarrier>
</SUBSEGMENT>
</APPROACH>
</INTERSECTION>
</ARTERIAL>
</TMML>
AP_DB
AP_DB.doc - Micros...
AP_DB example : D...

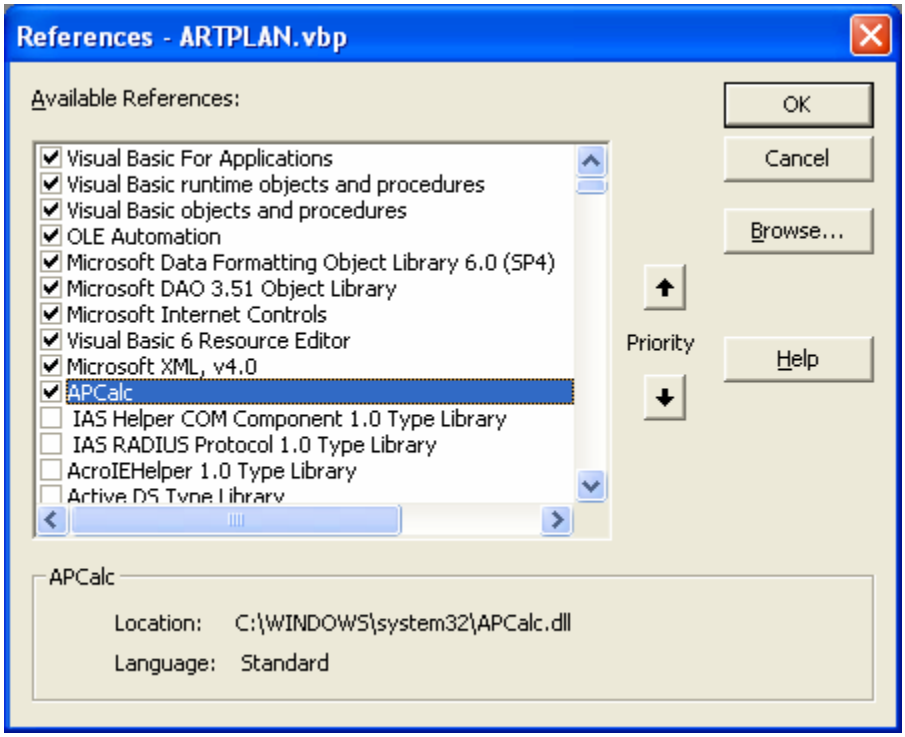
APPENDIX E:

DLL DOCUMENTATION

Guidelines for adding the ARTPLAN calculations engine (APCalc.dll) to your Visual Basic project

1. Add a Reference to the 'APCalc.dll' to Your Visual Basic Project

1.1 Place 'APCalc.dll' file in project or 'Windows\System' directory
1.2 Go to the 'Project' menu item
1.3 Click on 'References'
1.4 Then click on 'Browse…' to search for the APCalc.dll file
1.5 Then use the 'Look in:' Box to search for the APCalc.dll in your directory structure.
1.6 The 'References' dialog will now show APCalc.dll (as shown in the figure below)
1.7 Click 'OK'. The DLL is now properly referenced to your VB project.



2. Declare DLL

Add the following statement to your VB project's declarations section:

```
Public dll As APCalcs
```

3. Initialize DLL

Initialize the DLL in your project's startup subroutine, with the following statement:

```
Set dll = New APCalcs
```

4. DLL Function Calls

Add the 'DllCalc' subroutine to your VB project (included below). This subroutine contains all the code and DLL function calls necessary to retrieve all auto and multimodal calculated performance measures and levels of service. The subroutine shown below makes use of the data structure in FDOT's ARTPLAN. However, it is making explicit reference to each of the variable elements within the data structure so that you can easily substitute your own project's variable names. Shown immediately below the subroutine listing is the specification for each of the variables used in the calculations DLL.

```
Public Sub DllCalc()

   Dim DllPkHrVolRet() As Single
   Dim DllBothDirVolRet() As Single
   Dim DllAADTfinal() As Single
   Dim DllAADTflag() As Boolean

   Dim DllServiceBike() As Single
   Dim DllServicePed() As Long
   Dim DllServiceBus() As Single
   Dim DllDailyServiceBus() As Single

   Dim i As Integer, ii As Integer, j As Integer

'********************** Transfer input values to DLL **********************

     Call dll.InitialSysInfo(ArtInfo.NumSegments, SystemInfo.ArtClass, SystemInfo.AreaType, _
          SystemInfo.AADT, SystemInfo.Dfactor, SystemInfo.gCratio, SystemInfo.Kfactor, _
          SystemInfo.PHF, SystemInfo.PTXL, SystemInfo.BaseSatFlow, SystemInfo.FwdLanes, _
          SystemInfo.Cycle, SystemInfo.SigControl, SystemInfo.MedianType, _
          SystemInfo.PctHeavyVeh)

     For i = 1 To ArtInfo.NumSegments
          Call dll.InitialIntInfo(i, IntInfo(1, i).CrossName, IntInfo(1, i).Cycle, IntInfo(1, i).gCratio, _
            IntInfo(1, i).SegLength, IntInfo(1, i).SegAADT, IntInfo(1, i).PTXL, _
            IntInfo(1, i).NumLanes, IntInfo(1, i).ArvType, IntInfo(1, i).FFSpeed, _
            IntInfo(1, i).SegVol, Intinfo(1, i).MedianType)
     Next i
```

```
  For i = 0 To ArtInfo.NumSegments – 1
      Call dll.InitialMMsegment(i, MMSegment(i).BikeLaneYN, MMSegment(i).OutLaneWid, _
        MMSegment(i).NumOutLaneWid, MMSegment(i).PavCond, MMSegment(i).SidewalkSep, _
        MMSegment(i).SidewalkYN, MMSegment(i).SidewalkBarr, MMSegment(i).SidewalkToBusYN, _
        MMSegment(i).BusesPerHour, MMSegment(i).SpanOfService)

      For ii = 0 To 2
        Call dll.PedSubSegInfo(i, ii, PedSubSeg(i).NumSubSeg, _
          PedSubSeg(i).PctOfSegment(ii), PedSubSeg(i).SidewalkYN(ii), _
          PedSubSeg(i).SidewalkSep(ii), PedSubSeg(i).SidewalkBarr(ii))
      Next ii
  Next i


'******************** Get Adjusted Saturation Flow Value ******************

SystemInfo.AdjSatFlow = dll.AdjSatFlowRate(SystemInfo.PctHeavyVeh, SystemInfo.MedianType, _
                        SystemInfo.BaseSatFlow, SystemInfo.LTBayYN, SystemInfo.LocAdjFactor)


'******************** Call the Calculations Routines *********************

Call dll.RunCalc

'******************** Get Auto Performance Measure Values and LOS ******************

For i = 1 To ArtInfo.NumSegments
   IntInfo(1, i).ThruVol = dll.ThruVolRet(i)
   IntInfo(1, i).vcRatio = dll.VcRatioRet(i)
   IntInfo(1, i).CtrlDelay = dll.CtrlDelayRet(i)
   IntInfo(1, i).IntLOS = dll.IntLosRet(i)
   SegInfo(1, i).AvgSpd = dll.SegAvgSpdRet(i)
   SegInfo(1, i).LOS = dll.SegLOSRet(i)
Next i

'************** Get Service Volume Table Values ******************

DllPkHrVolRet = dll.PkHrVolRet
DllBothDirVolRet = dll.BothDirVolRet
DllAADTfinal = dll.AADTfinalRet
DllAADTflag = dll.AADTflagRet
DllServicePed = dll.ServicePedRet
DllServiceBike = dll.ServiceBikeRet
DllServiceBus = dll.ServiceBusRet
DllDailyServiceBus = dll.DailyServiceBusRet

For j = 1 To 4          'lanes 2,4,6,8
   For i = 1 To 5         'LOS A - E
      PkHrVol(j, i) = DllPkHrVolRet(j, i)
      BothDirVol(j, i) = DllBothDirVolRet(j, i)
      AADTfinal(j, i) = DllAADTfinal(j, i)
      AADTflag(j, i) = DllAADTflag(j, i)
      ServiceBike(j, i) = DllServiceBike(j, i)
      ServicePed(j, i) = DllServicePed(j, i)
   Next i
Next j
```

```
   For i = 1 To 6
      ServiceBus(i) = DllServiceBus(i)
      DailyServiceBus(i) = DllDailyServiceBus(i)
   Next i

‘ check for oversaturation for entire hour of analysis (i.e., 1/PHF)
   For i = 1 To ArtInfo.NumSegments
      OverVC(i) = dll.OverVCRet(i)
   Next i

'************** Get Multimodal Performance Measure Values *****************

   For i = 1 To ArtInfo.NumSegments
       BkLosVal(i) = dll.BikeLosValRet(i)
       PedLosVal(i) = dll.PedLosValRet(i)
       TransitLosVal(i) = dll.TransitLosValRet(i)
   Next i

   For i = 1 To ArtInfo.NumSegments
       Arterial.segresult(i).BikeLOS = dll.segBikeLosRet(i)
       Arterial.segresult(i).PedLOS = dll.SegPedLosRet(i)
       Arterial.segresult(i).TransitLOS = dll.SegTransitLosRet(i)
    Next i

   For i = 1 To 8
      For j = 1 To 3          ‘# of ped subsegments
          Arterial.segresult(i).SubSegPedLOS(j) = dll.PedSubSegLosRet (i, j)
      Next j
   Next i


'************** Get Overall Arterial LOS Results *****************

    ArtInfo.AvgSpd = dll.ArtInfoAvgSpdRet
    ArtInfo.LOS = dll.ArtInfoLOSRet
    ArtInfo.SumLength = dll.ArtSumLengthRet
    Arterial.TotPedLOS = dll.TotPedLosRet
    Arterial.TotPedPts = dll.TotPedPtsRet
    Arterial.TotBikeLOS = dll.TotBikeLosRet
    Arterial.TotBikePts = dll.TotBikePtsRet
    Arterial.TotBusLOS = dll.TotBusLosRet
    Arterial.TotBusPts = dll.TotBusPtsRet


End Sub
```

---

5. Calculations Subroutine Call

Insert the following statement in your VB project code at the point where you want to retrieve calculations results:

Call DllCalc

Variable Descriptions

Facility (SystemInfo. & ArtInfo.) Variables

Variable Name	Data Type	Allowable Values	Units / Notes
NumSegments	Integer	0, 1, …, 9	
ArtClass	Integer	1, 2, 3, 4, 5	5 = Signal
AreaType	String	"Urbanized", "Transitioning/Urban", "Rural"	
AADT	Long	0 – 100,000	vehicles
Dfactor	Single	0.5 – 1	
Kfactor	Single	0.01 - 0.15	
PHF	Single	0.75 – 1.0	
PTXL	Single	0 – 45	% Turns from Exclusive Lanes
BaseSatFlow	Long	100 to 2400	FDOT default = 1850 pc/hr/ln
gCratio	Single	0.1 – 1	
FwdLanes	Integer	1 – 4	
Cycle	Integer	40 – 240	seconds
SigControl	String	"Pretimed", "Semiactuated", "Actuated"	
MedianType	String	"None", "Non-Restrictive", "Restrictive"	
PctHeavyVeh	Single	0 – 100	%
OverVC	Boolean	True, False	v/c > 1/PHF indicator

Segment (IntInfo.) Variables

Variable Name	Data Type	Allowable Values	Units / Notes
CrossName	String	Any String Value	
Cycle	Integer	40 – 240	seconds
gCratio	Single	0.1 – 1	
SegLength	Integer	100 – 15840	feet
SegAADT	Long	1000 – 100000	Vehicles (optional, see below)
SegVol	Integer	0 – 5000	vehicles (analysis direction), if equal to zero, SegVol = SegAADT * Kfactor * Dfactor
PTXL	Single	0 – 65	%
NumLanes	Single	1 – 4	analysis direction
ArvType	Integer	1 – 6	
FFSpeed	Integer	25 – 60	mi/hr

MultiModal (MMSegment.) Variables

Variable Name	Data Type	Allowable Values	Units / Notes
BikeLaneYN	Integer	0, 1	no, yes
OutLaneWid	Integer	0, 1, 2, 3	Narrow, Typical, Wide, Specify Width
NumOutLaneWid	Integer	8 – 16	Feet (applies to selection of 3 for above)
PavCond	Integer	0, 1, 2	Undesirable, Typical, Desirable
SidewalkSep	Integer	0, 1, 2	Adjacent, Typical, Wide
SidewalkYN	Integer	0, 1	no, yes
SidewalkBarr	Integer	0, 1	no, yes
SidewalkToBusYN	Integer	0, 1	no, yes
BusesPerHour	Integer	0 – 60	buses/hr
SpanOfService	Integer	0 – 24	hours

PedSubsegment (PedSubSeg.) Variables

Variable Name	Data Type	Allowable Values	Units / Notes
NumSubSeg	Integer	1, 2, 3	
PctOfSegment	Integer	0 – 100	%
SidewalkYN	Integer	0, 1	no, yes
SidewalkSep	Integer	0, 1, 2	Adjacent, Typical, Wide
SidewalkBarr	Integer	0, 1	no, yes

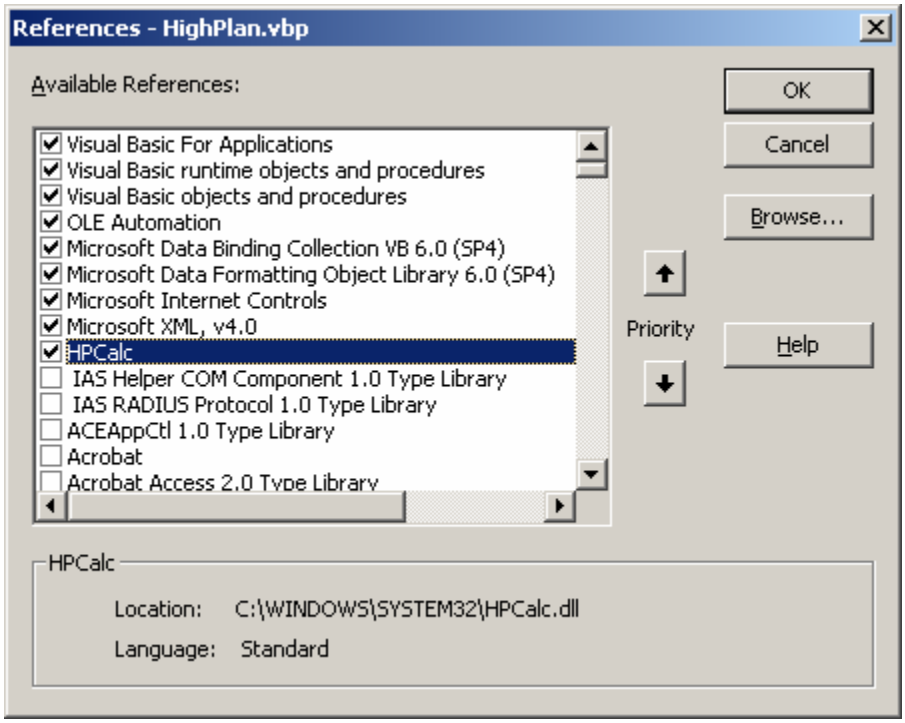
Adjusted Saturation Flow Variables

Variable Name	Data Type	Allowable Values	Units / Notes
PctHeavyVeh	Single	0 – 100	%
MedianType	String	“None”, “Non-Restrictive”, “Restrictive”	
BaseSatFlow	Long	100 to 2400	FDOT default = 1850 pc/hr/ln
LTBayYN	Integer	0, 1	no, yes
LocAdjFactor	Single	.75 – 1.0	

Guidelines for adding the HIGHPLAN calculations engine (HPCalc.dll) to your Visual Basic project

1. Add a Reference to the 'HPCalc.dll' to Your Visual Basic Project

1.1 Place 'HPCalc.dll' file in project or 'Windows\System32' directory
1.2 Go to the 'Project' menu item
1.3 Click on 'References'
1.4 Then click on 'Browse…' to search for the HPCalc.dll file
1.5 Then use the 'Look in:' Box to search for the HPCalc.dll in your directory structure.
1.6 The 'References' dialog will now show HPCalc.dll (as shown in the figure below)
1.7 Click 'OK'. The DLL is now properly referenced to your VB project.



2. Declare DLL

Add the following statement to your VB project's declarations section:

```
Public dll As TwoLaneCalc
Public Mdll As MultiLaneCalc
```

3. Initialize DLL

Initialize the DLL in your project's startup subroutine, with the following statements:

```
Set dll = New TwoLaneCalc
```

```
Set Mdll = New MultiLaneCalc
```

4. DLL Function Calls

Add the 'TwoLaneDLLCall' and 'MultiLaneDLLCall' subroutines to your VB project (included below). These subroutines contain all the code and DLL function calls necessary to retrieve all calculated performance measures and levels of service for either two-lane or multilane highways. The subroutines shown below make use of the data structure in FDOT's HIGHPLAN program. However, it is making explicit reference to each of the variable elements within the data structure so that you can easily substitute your own project's variable names. Shown immediately below the subroutine listings is the specification for each of the variables used in the calculations DLLs.

Two-Lane Highways

```
Public Sub TwoLaneDLLCall()

Dim i As Integer
Dim DDHV As Single
Dim DllLOSTabRet() As TabStuff


'********************** Transfer input values to DLL **********************

Call dll.InitialHighInfo(HighwayInfo.AreaType, HighwayInfo.NumberOfLns, _
    HighwayInfo.Terrain, HighwayInfo.FreeFlowSpeed, _
    HighwayInfo.MedianYN, HighwayInfo.LTBaysYN, _
    HighwayInfo.PassLnYN, HighwayInfo.PassingLnSpacing, _
    HighwayInfo.PctNoPassingZone, _
    HighwayInfo.AADT, HighwayInfo.KFactor, HighwayInfo.Dfactor, _
    HighwayInfo.PHF, HighwayInfo.HVPct, HighwayInfo.BaseCapPerLn, _
    HighwayInfo.LocalAdj, _
    HighwayInfo.AnalysisType, HighwayInfo.Class)


Call dll.Setup2Lane (HP2LaneParPath)

DDHV = dll.RetDDHV
Call dll.TwolaneMOEs(DDHV, 0)

'************** Get LOS Results ****************

HighwayInfo.VCRatio = dll.RetVCRatio
HighwayInfo.PTSF = dll.RetPTSF
```

```
HighwayInfo.ATS = dll.RetATS
HighwayInfo.PctFFS = dll.RetPctFFS
HighwayInfo.LOSMOE = dll.RetLOSMOE
HighwayInfo.LOS = dll.RetLOS

For i = 0 To 4
    HighwayInfo.MOE(i) = dll.RetMOE(i)
Next i
HighwayInfo.MOE(1) = HighwayInfo.VCRatio

Call dll.TwoLaneServiceVols
DllLOSTabRet = dll.LOSTabRet

For i = 1 To 5          'LOS A - E
    LOSTable(1).PkServiceVol(i) = DllLOSTabRet(1).PkServiceVol(i)
    LOSTable(1).BothServiceVol(i) = DllLOSTabRet(1).BothServiceVol(i)
    LOSTable(1).MaxVCRatio(i) = DllLOSTabRet(1).MaxVCRatio(i)
    LOSTable(1).MaxAADT(i) = DllLOSTabRet(1).MaxAADT(i)
Next i

End Sub
```

Mulitlane Highways

```
Public Sub MultiLaneDLLCall()

Dim i As Integer, j As Integer
Dim DllLOSTabMulRet() As TabStuff


'********************** Transfer input values to DLL **********************

Call Mdll.InitialMulLaneHighInfo(HighwayInfo.AreaType, HighwayInfo.Terrain, _
    HighwayInfo.NumberOfLns, HighwayInfo.FreeFlowSpeed, _
    HighwayInfo.MedianYN, HighwayInfo.LTBaysYN, _
    HighwayInfo.AADT, HighwayInfo.KFactor, HighwayInfo.Dfactor, _
    HighwayInfo.PHF, HighwayInfo.HVPct, HighwayInfo.BaseCapPerLn, _
    HighwayInfo.LocalAdj, HighwayInfo.AnalysisType)

    Call Mdll.MultilaneMOEs


'************* Get Intermediate Calculation Values ***************

   FacAdjFactor = dll.retFacAdjFact
```

```
    Adj_Med_LTB = dll.retAdjMedLT
        HighwayInfo.AdjCapPerLn = Mdll.AdjCapPerLnMulRet


'************** Get LOS Results *****************

        HighwayInfo.VCRatio = Mdll.RetVCRatioML
        HighwayInfo.ATS = Mdll.RetATSML
        HighwayInfo.Density = Mdll.RetDensityML
        HighwayInfo.LOSMOE = dll.RetLOSMOE
        HighwayInfo.LOS = Mdll.RetLOSML

    For i = 0 To 4
      HighwayInfo.MOE(i) = dll.RetMOE(i)
    Next i
    HighwayInfo.MOE(1) = HighwayInfo.VCRatio

    DllLOSTabMulRet = Mdll.LOSTabRet

    For i = 1 To 3    'for 4, 6, and 8 lanes
      For j = 1 To 5  'LOS A - E
        LOSTable(i).PkServiceVol(j) = DllLOSTabMulRet(i).PkServiceVol(j)
        LOSTable(i).BothServiceVol(j) = DllLOSTabMulRet(i).BothServiceVol(j)
        LOSTable(i).MaxVCRatio(j) = DllLOSTabMulRet(i).MaxVCRatio(j)
        LOSTable(i).MaxAADT(j) = DllLOSTabMulRet(i).MaxAADT(j)
     Next j
  Next i

  End Sub
```

5. Calculations Subroutine Call

Insert the following statement in your VB project code at the point where you want to retrieve calculations results:

Call TwoLaneDLLCall (for two-lane calculations)
Call MultiLaneDLLCall (for multilane calculations)

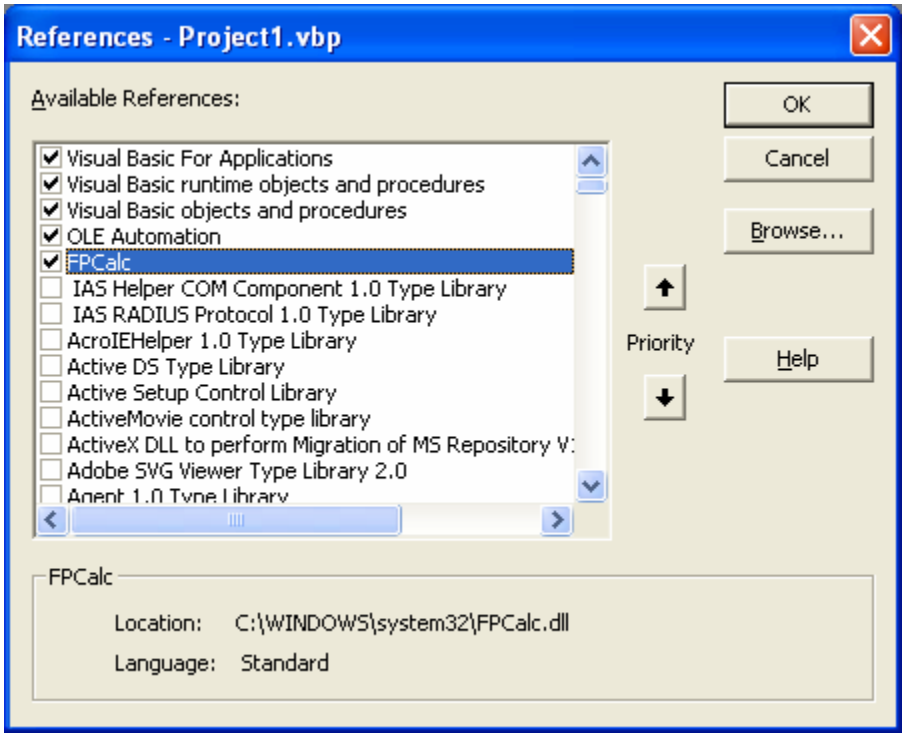
Variable Specifications

Variable Name	Data Type	Allowable Values	Units / Notes
AnalysisType	Integer	0, 1	"Facility", "Segment"
AreaType	String	"Urbanized", "Transitioning /Urban", "Rural developed", "Rural undeveloped"	
NumberOfLns	Integer	2 (2 lane) 4 – 8 (multilane)	
Terrain	Integer	0, 1	"Level", "Rolling"
FreeFlowSpeed	Integer	40 – 70 (2 lane) 45 – 70 (multilane)	FDOT default = posted speed + 5 mi/h
MedianYN	Boolean	True / False	
LTBaysYN	Boolean	True / False	
AADT	Long	1000 – 100000	vehicles
Kfactor	Single	0.01 – 0.2	
Dfactor	Single	0.5 – 1	
PHF	Single	0.75 – 0.98	
HVPct	Integer	0 – 100	%
BaseCapPerLn	Integer	1400 – 2400	FDOT default = 1850 pc/h/ln
LocalAdj	Single	0.8 – 1.0	
Two-Lane Highways Only			
Class	String	"I", "II", "III", "IV"	
PassLnYN	Boolean	True / False	
PassingLnSpacing	String	0 (if PassLnYN = False) 2 – 30 (if PassLnYN = True)	mi
PctNoPassingZone	String	0 – 100	%
HP2LaneParPath	String		Path of 'HP2Lane.par' file

Guidelines for adding the FREEPLAN calculations engine (FPCalc.dll) to your Visual Basic project

1. Add a Reference to the 'FPCalc.dll' to Your Visual Basic Project

1.1 Place 'FPCalc.dll' file in project or 'Windows\System32' directory
1.2 Go to the 'Project' menu item
1.3 Click on 'References'
1.4 Then click on 'Browse…' to search for the FPCalc.dll file
1.5 Then use the 'Look in:' Box to search for the FPCalc.dll in your directory structure.
1.6 The 'References' dialog will now show FPCalc.dll (as shown in the figure below)
1.7 Click 'OK'. The DLL is now properly referenced to your VB project.



2. Declare DLL

Add the following statement to your VB project's declarations section:

```
Public dll As FPCalcs
```

3. Initialize DLL

Initialize the DLL in your project's startup subroutine, with the following statement:

```
Set dll = New FPCalcs
```

4. DLL Function Calls

Add the 'CalcsDLL' subroutine to your VB project (included below). This subroutine contains all the code and DLL function calls necessary to retrieve all facility and segment calculated performance measures and levels of service. The subroutine shown below makes use of the data structure in FDOT's FREEPLAN. However, it is making explicit reference to each of the variable elements within the data structure so that you can easily substitute your own project's variable names. Shown immediately below the subroutine listing is the specification for each of the variables used in the calculations DLL.

```
Public Sub DllCalc()    'FPCalc DLL

Dim i As Integer, j As Integer, k As Integer

' ************** Initialize the Facility Variables *****************

Call dll.InitialFacility(facility.AreaType, facility.FreewayClass, facility.numseg, facility.AADT, _
        facility.KFactor, facility.DFactor, facility.PeakhourFactor, facility.LocalAdjFactor, _
        facility.timePeriod, facility.direction, facility.peak, _
        facility.speedLimit, facility.default.Lanes, _
        facility.default.offrampDecelLength, facility.default.onRampAccelLength, _
        facility.default.PercentTrucks, facility.default.percenttrucksonoff, _
        facility.default.rampFFS, facility.default.percentrampvolume, _
        facility.HCManalysis)


' ************** Initialize the Segment Variables ***************

  For i = 0 To facility.numseg – 1       'Segment
    For j = 0 To 2        '* see note at bottom
      Call dll.InitialSegment(i, j, facility.segarray(i).ChangeInts, _
                facility.segarray(i).offRampPercentTrucks(j), _
                facility.segarray(i).onRampPercentTrucks(j), _
                facility.segarray(i).onRampVPH(j), facility.segarray(i).offRampVPH(j), _
                facility.segarray(i).onramplanes(j), facility.segarray(i).offramplanes(j), _
                facility.segarray(i).Length, _
                facility.segarray(i).InbetweenLength, _
                facility.segarray(i).AuxLaneBetweenInterchanges, _
                facility.segarray(i).auxlane, facility.segarray(i).Lanes, _
                facility.segarray(i).Terrain, facility.segarray(i).ffs, _
                facility.segarray(i).Type, _
                facility.segarray(i).Distance(j), facility.segarray(i).typenum)
    Next j
  Next i
```

* This index is for accommodating the full cloverleaf interchange. The first on (loop)- and off (directional)-ramps use index 0 and the second on-(directional) and off-(loop) ramps use index 1. Three distances are also specified for full-cloverleaf: 1) first off-ramp gore to first on-ramp gore; 2) first on-ramp gore to second off-ramp gore; and 3) second off-ramp gore to second on-ramp gore. Thus, indexes 0,1, and 2 would be used, respectively. For all other interchange types, the zero index should be used to specify on-/off-ramp values and gore-to-gore distances.
It is necessary to enter a segment length for all basic segment entries (i.e., indexes 0,1,2)

```
' *********** Call the Calculation Module ************

  dll.DoCalcs

' ************ Return Variables for the Result Screen ************

  For k = 0 To facility.numseg - 1
        facility.segresult(k).los = dll.segresultlos(k)
        facility.segresult(k).speed = dll.segresultspeed(k)
        facility.segresult(k).density = dll.segresultdensity(k)
        facility.segresult(k).capacity = dll.segresultcapacity(k)
        Involume(k) = dll.Volume(k)
        facility.segworst(k).ramp = dll.segworstramp(k)
        facility.segworst(k).los = dll.segworstlos(k)
        facility.segworst(k).density = dll.segworstdensity(k)
        facility.segworst(k).speed = dll.segworstspeed(k)
        facility.segworst(k).vcratio = dll.segworstvcratio(k)
  Next k

‘ ************* Overall Values on the Result Screen **************

  facility.overallDensity = dll.overallDensity
  facility.overallLOS = dll.overallLOS
  facility.overallSpeed = dll.overallSpeed

End Sub
```

---

5. Calculations Subroutine Call

Insert the following statement in your VB project code at the point where you want to retrieve calculations results:

Call DllCalc

Variable Specifications

Variable Name	Data Type	Allowable Values	Units / Notes
numseg	String	1 – 20	
AADT	Long	5,000 – 300,000	vehicles
KFactor	Single	0.01 – 0.40	
DFactor	Single	0.4 – 1.0	
PHF	Single	0.5 – 1.0	
LocalAdjFactor	Single	0.75 – 1.0.	
timePeriod	String	K30, K100, K5-6pm, Kp/d, Kother	
direction	String	Eastbound, Westbound, Northbound, Southbound	
peak	String	Peak Direction, Off-Peak Direction	
AreaType	String	Urbanized, Transitioning/Urban, Rural	
FreewayClass	String	1 – 4	
speedLimit	Integer	50 – 70	mi/h
defaultLanes	Integer	4 – 12	lanes both direction
offrampDecelLength	Integer	Default value – 500	
onRampAccelLength	Integer	Default value – 1200	
PercentTrucks	Integer	Default value – 6	Entering first segment
percenttrucksonoff	Integer	Default value – 6	
rampFFS	Integer	Default value – 40	
percentrampvolume	Integer	Default value – 10	
HCManalysis	Boolean	True / False	

Segment Variables

Variable Name	Data Type	Allowable Values	Units / Notes
ChangeInts	Boolean	True, False	
offRampPercentTrucks	Single	0 – 100	%
onRampPercentTrucks	Single	0 – 100	%
onRampVPH	Integer	0 – 2200	veh/hr/ln
offRampVPH	Integer	0 – 2200	veh/hr/ln
onramplanes	Integer	1 – 2	
offramplanes	Integer	1 – 2	
Length	Integer	> 200, if back-to-back interchanges 0, otherwise	ft
InbetweenLength	Integer	2000 – 7200 default = 3000	ft, dist between back-to-back interchanges
AuxLaneBetweenInterchanges	Boolean	True, False	
auxlane	Integer	2000 – 7200	ft, distance of aux lane
Lanes	Integer	2 – 6	

Terrain	String	L, R, M	level, rolling, mountainous
ffs	Integer	55 – 75	mi/h
Type	String	Basic Segment, Interchange, Diamond, Partial Cloverleaf, Full Cloverleaf, On Ramp, Off Ramp, Toll Plaza	
Distance	Single		ft, ramp gore-to-gore distance
typenum	Integer	Basic Segment = 0 Interchange = 1 Diamond = 2 Partial Cloverleaf = 5 Full Cloverleaf = 6 On Ramp = 7 Off Ramp = 8 Toll Plaza = 11	