***CO Florida 2012  
Visual Basic Program Documentation***  
FDOT Intersection Air Quality (CO) Screening Model

Developed Under Research Sponsored by

The Florida Department of Transportation  
Project # BDK78 TWO 98501

FDOT Project Officer:  
Mr. Mariano Berrios

Developed at the University of Central Florida  
by  
  
C. David Cooper, PhD, PE, QEP  
and  
Mark D. Ritner, EI, EPI

December 15, 2011

TABLE OF CONTENTS

[Background 1](#_Toc313890917)

[Program Flow 2](#_Toc313890918)

[Variable Glossary 3](#_Toc313890919)

[Forms and Module Summary 15](#_Toc313890920)

[Subroutine Summary 16](#_Toc313890921)

[Emission Factor Data Tables 25](#_Toc313890922)

[Program Code 27](#_Toc313890923)

[Module 1 27](#_Toc313890924)

[Run 107](#_Toc313890925)

[About 129](#_Toc313890926)

[Title 132](#_Toc313890927)

[District 145](#_Toc313890928)

[Intersection Type 157](#_Toc313890929)

[Int4X4 182](#_Toc313890930)

[Int4X6 193](#_Toc313890931)

[Int6X4 204](#_Toc313890932)

[Int6X6 215](#_Toc313890933)

[East Tee 226](#_Toc313890934)

[North Tee 236](#_Toc313890935)

[E-W Diamond 264](#_Toc313890936)  
[N-S Diamond 277](#_Toc313890937)  
[Tollbooth 290](#_Toc313890938)

# Background

This document is a companion to the *User’s Guide to COFL2012.* The purpose of this document is to facilitate the examination of the COFL 2012 program code, whether it be for academic research, future program updates, or to satisfy user curiosity. This document is not required for the operation of COFL 2012 for roadway projects.

COFL 2012 was written using the Microsoft Visual Studio 2010 suite with Visual Basic as the programming language. The variable and subroutine glossaries presented in this document are also found on the companion Excel Spreadsheet, “COFL2012ProgramDocumentation.xlsx”. The spreadsheet affords the user the option of custom sorting the variables and/or subroutines.

COFL 2012 has been written and documented with ease of program maintenance and future updates in mind. When future updates to the EPA’s MOVES databases are made, new emission factors may be compiled into the external COFL 2012 look-up tables, without requiring any program modifications to the Visual Basic code. The formatting of these text files is discussed in the *Emission Factor Data Tables* section of this text.

# Program Flow

End Program

Welcome

Title

Save

New/Open

District

Save

New/Open

Intersection Type

Save

New/Open

New/Open

Save

Intersection Data

New/Open

Run

Save Input

Save Output

Print Output

# Variable Glossary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| AMB | Constant = 0 | Ambient CO = 0 for CAL3QHC input file | Global | Module1 | Initial Declarations |
| AT | Constant = 3 | Arrival Time = average progression for CAL3QHC input file | Global | Module1 | Initial Declarations |
| ATDep | Integer | Traffic volume entering arterial road - Unique to Freeway Diamond configurations | Global | Module1 | Initial Declarations |
| ATEB | String | Approach Traffic East Bound | Global | Module1 | Initial Declarations |
| ATIM | Constant = 60 | Averaging Time for CAL3QHC input file | Global | Module1 | Initial Declarations |
| ATLeft | Integer | Approach Traffic Left-Turning | Global | Module1 | Initial Declarations |
| ATMax | Integer | Approach Traffic Maximum | Global | Module1 | Initial Declarations |
| ATNB | String | Approach Traffic North Bound | Global | Module1 | Initial Declarations |
| ATQ | Integer | Used only in Tee Configurations | Global | Module1 | Initial Declarations |
| ATRight | Integer | Approach Traffic Right-Turning | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| ATSB | String | Approach Traffic South Bound | Global | Module1 | Initial Declarations |
| ATWB | String | Approach Traffic West Bound | Global | Module1 | Initial Declarations |
| BRG | Constant = 0 | Wind Direction = variable for CAL3QHC input file | Global | Module1 | Initial Declarations |
| CALINputLIne | String Array (111) | Array of CAL3QHC lines to be written to CAL3QHC input file | Global | Module1 | Initial Declarations |
| CLAS | Integer | Stability Class Number (1-5) for CAL3QHC input file | Global | Module1 | Initial Declarations |
| CLASLetter | String | Stability Class Letter (A-E) for user outputs | Global | Module1 | Initial Declarations |
| COBack1Hr | String | 1-hr CO Background concentration | Global | Module1 | Initial Declarations |
| COBack8Hr | String | 8-hr CO Background concentration | Global | Module1 | Initial Declarations |
| Conc1 | Single Array (20) | Array of 1-hr CO output concentrations (including background) | Global | Module1 | Initial Declarations |
| Conc8 | Single Array (20) | Array of 8-hr CO output concentrations (including background) | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| DEGR | Constant = 5 | Wind Degree Increment for CAL3QHC input file | Global | Module1 | Initial Declarations |
| Delay | Integer | Holds numeric value that is used as a delay mechanism | Local | Module1 | ReadCAL3qHCOUT |
| DistrictNum | Integer | FDOT District Number | Global | Module1 | Initial Declarations |
| EF10mph | Single | Emission Factor @ 10 mph- The EF used for the staging section approaching a tollbooth stop | Global | Module1 | Initial Declarations |
| EFArray | Single Array (65,40) | Stores the EF values read in from text file | Global | Module1 | Initial Declarations |
| EFFree | Single | Emission Factor Free - The EF used for all free flow links on arterial roads | Global | Module1 | Initial Declarations |
| EFHwy | Single | Emission Factor Highway- The EF used for the cruise links on the freeway diamonds and tollbooths | Global | Module1 | Initial Declarations |
| EFHwyOn | Single | Emission Factor for acceleration onto a highway | Global | Module1 | Initial Declarations |
| EFId | Single | Emission Factor for Idle | Global | Module1 | Initial Declarations |
| EFLt | Single | Emission Factor for Left Turns (20 mph) | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| EFNB | Single | Emission Factor North Bound | Global | Module1 | Initial Declarations |
| EFNEB | Single | Emission Factor East Bound | Global | Module1 | Initial Declarations |
| EFRt | Single | Emission Factor for Right Turns (15 mph) | Global | Module1 | Initial Declarations |
| EFSB | Single | Emission Factor South Bound | Global | Module1 | Initial Declarations |
| EFTollAcc | Single | Emission Factor Tollbooth Acceleration - The EF used for acceleration away from a toll booth | Global | Module1 | Initial Declarations |
| EFWB | Single | Emission Factor West Bound | Global | Module1 | Initial Declarations |
| ETCEB | String | Electronic Toll Collection East Bound - Unique to Freeway Tollbooth configurations | Global | Module1 | Initial Declarations |
| ETCSB | String | Electronic Toll Collection South Bound - Unique to Freeway Tollbooth configurations | Global | Module1 | Initial Declarations |
| ETCSNB | String | Electronic Toll Collection North Bound - Unique to Freeway Tollbooth configurations | Global | Module1 | Initial Declarations |
| ETCWB | String | Electronic Toll Collection West Bound - Unique to Freeway Tollbooth configurations | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| FacilityName | String | "Facility Name" on Title Screen | Global | Module1 | Initial Declarations |
| FDOTin | String | Holds input file name | Global | Module1 | Initial Declarations |
| FDOTout | String | Holds output file name | Global | Module1 | Initial Declarations |
| FinalClose | Boolean | Used when exiting the program | Local | Run | Initial Declarations |
| FTDep | Integer | Traffic volume entering freeway - Unique to Freeway Diamond configurations | Global | Module1 | Initial Declarations |
| FTLeft | Integer | Freeway Offramp Traffic that turns Left | Global | Module1 | Initial Declarations |
| FTMax | Integer | Freeway Approach Traffic Maximum | Global | Module1 | Initial Declarations |
| FTRight | Integer | Freeway Offramp Traffic that turns Right | Global | Module1 | Initial Declarations |
| HL | Constant = 0 | Source Height for CAL3QHC input file | Global | Module1 | Initial Declarations |
| HwySpeed | Integer | Freeway speed that produces the highest EF | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| i | Integer | General Counter Variable | Local | Various | Various |
| IDEBUG | Constant = 1 | Debugging Option for CAL3QHC input file | Global | Module1 | Initial Declarations |
| IFR | String array (15) | Holds variables read in from an existing user input file | Global | Module1 | Initial Declarations |
| InputsCorrect | Boolean Array (4) | Holds the completion status of each of the input screens | Global | Module1 | Initial Declarations |
| IntType | String | Roadway Configuration selected | Global | Module1 | Initial Declarations |
| IOPT | Constant = 1 | Metric to English Conversion Factor for CAL3QHC input file | Global | Module1 | Initial Declarations |
| j | Integer | General Counter Variable | Local | Various | Various |
| Job | String | "Project Title" on Title Screen | Global | Module1 | Initial Declarations |
| LandUse | String | "Land Use" on Title Screen | Global | Module1 | Initial Declarations |
| LIneOfText | String | Holds string to be written to file | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| MIXH | Constant = 1000 | Mixing Height for CAL3QHC input file | Global | Module1 | Initial Declarations |
| MODE | Constant = "'CO'" | Indicates CO Mode for CAL3QHC input file | Global | Module1 | Initial Declarations |
| MultAcc | Single | EF Multiplier for Arterial Roads, 0% Grade | Global | Module1 | Initial Declarations |
| MultHwyOn | Single | EF Multiplier for Freeway Onramps, 2% Grade | Global | Module1 | Initial Declarations |
| MultTollBooth | Single | EF Multiplier for acceleration away from toll booths, 0% grade | Global | Module1 | Initial Declarations |
| NL | Integer | Number of Links for CAL3QHC input file | Global | Module1 | Initial Declarations |
| NM | Constant = 1 | Number of Meteorological Conditions for CAL3QHC input file | Global | Module1 | Initial Declarations |
| NR | Integer | Number of Receptors for CAL3QHC input file | Global | Module1 | Initial Declarations |
| NumInputLines | Integer | Number of lines of to be written to file | Global | Module1 | Initial Declarations |
| OREB | String | On/Off Ramp East Bound - Unique to Freeway Diamond configurations | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| ORNB | String | On/Off Ramp North Bound - Unique to Freeway Diamond configurations | Global | Module1 | Initial Declarations |
| ORSB | String | On/Off Ramp South Bound - Unique to Freeway Diamond configurations | Global | Module1 | Initial Declarations |
| ORWB | String | On/Off Ramp WestBound - Unique to Freeway Diamond configurations | Global | Module1 | Initial Declarations |
| PassFail | Boolean | Holds status of test results | Global | Module1 | Initial Declarations |
| PPMString | String | Stores the line of text from the CAL3QHC output file that has the CO concentrations at each of the programmed receptors | Local | Module1 | ReadCAL3qHCOUT |
| prFont | Font | Holds the printer output font | Local | Run | PrintDocument1\_PrintPage\_1 |
| PrimaryDirectory | String | Used for navigating file structure, relative to where the program is located | Global | Module1 | Initial Declarations |
| PRINT2 | Constant = 0 | Indicates Summary Output Mode for CAL3QHC input file | Global | Module1 | Initial Declarations |
| ProjectYear | String | "Project Year" on Title Screen | Global | Module1 | Initial Declarations |
| QuickVerify | Boolean | Holds the result of the completion status of a form before navigating backwards using the upper navigation icons. It will allow the user to go backwards, without prompting them to complete the form. | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| RandNum | Integer | Holds a random number that is used to display a random image in a picture box | Global | Module1 | Various |
| ReadLIne | String Array (220) | Stores lines read from CAL3QHC output file | Local | Module1 | ReadCAL3qHCOUT |
| RunName | String | "Run Name" on Title Screen | Global | Module1 | Initial Declarations |
| RVP | Constant = 13.3 | Reid Vapor Pressure | Global | Module1 | Initial Declarations |
| sAppPath | String | Holds the directory path for running CAL3QHC | Global | Module1 | Initial Declarations |
| SCAL | Constant = 0.3048 | Scale Conversion Factor for CAL3QHC input file | Global | Module1 | Initial Declarations |
| SEB | String | Speed East Bound | Global | Module1 | Initial Declarations |
| SFR | Constant = 1600 | Saturation Flow Rate for CAL3QHC input file | Global | Module1 | Initial Declarations |
| SFRRamps | Constant = 1500 | Saturation Flow Rate on freeway ramps for CAL3QHC input file | Global | Module1 | Initial Declarations |
| SNB | String | Speed North Bound | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| Speed | Integer | Arterial speed that produces the highest EF | Global | Module1 | Initial Declarations |
| SSB | String | Speed South Bound | Global | Module1 | Initial Declarations |
| ST | Constant = 1 | Signal Type = pretimed for CAL3QHC input file | Global | Module1 | Initial Declarations |
| StreamToDisplay | StreamReader | Variable for re3ading from text files | Global | Module1 | Initial Declarations |
| StringToPrint | String | Holds strings for output printer arguments | Local | Run | PrintDocument1\_PrintPage\_1 |
| SWB | String | Speed West Bound | Global | Module1 | Initial Declarations |
| sWriter | IO.StreamWriter | Used for outputting to text files | Local | Various | Various |
| Tab1 | Integer | Used in formatting output | Local | Run | SaveOutputFile |
| Tab2 | Integer | Used in formatting output | Local | Run | SaveOutputFile |
| Tab3 | Integer | Used in formatting output | Local | Run | SaveOutputFile |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| Temperature | String | January, 7am temperature in selected FDOT district | Global | Module1 | Initial Declarations |
| TempIntType | String | Holds Intersection Type Temporarily | Local | Module1 | ClearAll |
| TempSingle | Single | Holds Values for use in multi-step calculations | Local | Module1 | GetEFs, GetDiamondEFs |
| TestMsg | String | Used for displaying message boxes | Global | Module1 | Initial Declarations |
| TPF | Constant = 0.6 | Total Persistence Factor | Global | Module1 | Initial Declarations |
| TYP | Constant = "'AG'" | Link type "at grade" for CAL3QHC input file | Global | Module1 | Initial Declarations |
| U | Constant = 1 | Wind Speed = 1 mps for CAL3QHC input file | Global | Module1 | Initial Declarations |
| UsersName | String | "User's Name" on Title Screen | Global | Module1 | Initial Declarations |
| VAL1 | Constant = 0 | Initial Wind Increment for CAL3QHC input file | Global | Module1 | Initial Declarations |
| VAL2 | Constant = 71 | Final Wind Increment for CAL3QHC input file | Global | Module1 | Initial Declarations |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Variable** | **Type** | **Description** | **Global or Local** | **Declaration Form** | **Declaration Subroutine** |
| VAR | Constant = "'Y'" | Wind Direction = variable for CAL3QHC input file | Global | Module1 | Initial Declarations |
| VD | Constant = 0 | Deposition Velocity for CAL3QHC input file | Global | Module1 | Initial Declarations |
| VS | Constant = 0 | Settling Velocity for CAL3QHC input file | Global | Module1 | Initial Declarations |
| x | Single | Distance required to accelerate to cruise velocity | Local | Module1 | GetEFs, GetDiamondEFs |
| XR | Integer Array (20) | X-coordinates for receptor locations for selected roadway configuration | Global | Module1 | Initial Declarations |
| y | Integer | Holds the rowposition for printer output | Local | Run | PrintDocument1\_PrintPage\_1 |
| YFAC | Constant = 3 | Clearance Lost Time for CAL3QHC input file | Global | Module1 | Initial Declarations |
| YR | Integer Array (20) | Y-coordinates for receptor locations for selected roadway configuration | Global | Module1 | Initial Declarations |
| ZO | Integer | Surface Roughness Coefficient for CAL3QHC input file | Global | Module1 | Initial Declarations |
| ZR | Constant = 6 | Receptor height | Global | Module1 | Initial Declarations |

# Forms and Module Summary

|  |  |
| --- | --- |
| **Name** | **Description** |
| Module1 | Contains global variable declarations and the majority of the program code. |
| Welcome | "Splash" screen. |
| Title | Project title, facility name, user name, run name,project year, land use |
| District | FDOT district |
| Intersection Type | Selection of roadway configuration |
| Int4X4 | Speed and approach traffic inputs for 4X4 intersection |
| Int4X6 | Speed and approach traffic inputs for 4X6 intersection |
| Int6X4 | Speed and approach traffic inputs for 6X4 intersection |
| Int6X6 | Speed and approach traffic inputs for 6X6 intersection |
| East Tee | Speed and approach traffic inputs for East Tee intersection |
| South Tee | Speed and approach traffic inputs for South Tee intersection |
| West Tee | Speed and approach traffic inputs for West Tee intersection |
| North Tee | Speed and approach traffic inputs for North Tee intersection |
| Tollbooth | Speed and approach traffic inputs freeway tollbooth configuration |
| E-WDiamond | Speed and approach traffic inputs E-W freeway diamond configuration |
| N-SDiamond | Speed and approach traffic inputs N-S freeway diamond configuration |
| Run | Runs CAL3QHC, displays resulting output, offers options to save and/or print output |
| About | Displays background information about the project development |

# Subroutine Summary

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| AllRegButtons | District | Displays all the "District" buttons in their regular, unselected state. | n/a |
| AllRegButtons | Title | Displays all the "Land Use" buttons in their regular, unselected state. | n/a |
| AllSmallPics | Intersection Type | Sets all eleven roadway configuration pictures to the small format. | n/a |
| Build4X4 | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| Build4X6 | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| Build6X4 | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| Build6X6 | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| BuildEastTee | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| BuildEWDiamond | Module1 | Constructs the CALInputLIne array for the run. | GetDiamondEFs |
| BuildNorthTee | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| BuildNSDiamond | Module1 | Constructs the CALInputLIne array for the run. | GetDiamondEFs |
| BuildSouthTee | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| BuildTollBooth | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |
| BuildWestTee | Module1 | Constructs the CALInputLIne array for the run. | GetEFs |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| ClearAll | Module1 | Clears out all user input variables when a new run is called for or an old run is to be loaded. | If an existing file is being opened, ClearAll calls the appropriate "Init\_\_\_" Subroutine. |
| FillEFArray | Module1 | Reads the selected EF text file into an array. | n/a |
| GenerateOutput | Module1 | Reads the Conc1 array and sends the output of the run to the screen. | ReadCAL3QHCOUT |
| GetDiamondEFs | Module1 | Reads and interpolates the necessary EFs from the EF array for the freeway diamond interchanges. | n/a |
| GetEFs | Module1 | Reads and interpolates the necessary EFs from the EF array for all roadway configurations, except for the freeway diamond interchanges. | n/a |
| Init4X4 | Int4X4 | Loads in the values of existing inputs . | n/a |
| Init4X6 | Int4X6 | Loads in the values of existing inputs . | n/a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| Init6X4 | Int6X4 | Loads in the values of existing inputs . | n/a |
| Init6X6 | Int6X6 | Loads in the values of existing inputs . | n/a |
| InitDistrict | District | Loads in the value of "District" if it already is known and switches that image to the highlighted format. Selects and displays a random Florida picture. | AllRegButtons |
| InitEastTee | East Tee | Loads in the values of existing inputs . | n/a |
| InitEWDiamond | E-Wdiamond | Loads in the values of existing inputs . | n/a |
| InitIntType | Intersection Type | Loads in the value of "Intersection Type" if it already is known and switches that image to the bigger format. Selects and displays a random Florida picture. | AllSmallPics |
| InitNorthTee | North Tee | Loads in the values of existing inputs . | n/a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| InitNSDiamond | NSDiamond | Loads in the values of existing inputs . | n/a |
| InitSouthTee | South Tee | Loads in the values of existing inputs . | n/a |
| InitTitle | Title | Loads in the values of existing inputs . | AllRegButtons |
| InitWestTee | WestTee | Loads in the values of existing inputs . | n/a |
| MakeARun | Module1 | Calls the main procedures required for a COFL 2012 "Run". | WriteCAL3QHCIN, GenerateOutput |
| OpenFile | Module1 | Opens and existing COFL 2012 input file and propogates the values into the user input forms. | n/a |
| PrintDocument1\_PrintPage\_1 | Run | Sends output to the printer | n/a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| ReadCAL3QHCOUT | Module1 | Reads the results of the "outcal3qhc.out" text file in and then extracts the 1-hr CO concentrations into the array Conc1. | n/a |
| SaveFile | Module1 | Saves the existing COFL 2012 user inputs. |  |
| SaveOutputFile | Run | Saves COFL 2012 output to a text file | n/a |
| Select4X4 | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| Select4X6 | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| Select6X4 | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| Select6X6 | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| SelectEastTee | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| SelectEWDiamond | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| SelectNorthTee | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| SelectNSDiamond | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| SelectSouthTee | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |
| SelectTollBooth | Intersection Type | Selects the intersection type. | n/a |
| SelectWestTee | Intersection Type | Selects the intersection type and fills the receptor array with the X and Y coordinates, unique to that type. | n/a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| Verify4X4Form | Int4X4 | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| Verify4X6Form | Int4X6 | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| Verify6X4Form | Int6X4 | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| Verify6X6Form | Int6X6 | Verifies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| VerifyEastTeeForm | East Tee | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| VerifyEWDiamondForm | E-Wdiamond | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| VerifyNorthTee | North Tee | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |

|  |  |  |  |
| --- | --- | --- | --- |
| **Subroutine Name** | **Location** | **Description** | **Other Subs Called** |
| VerifyNSDiamond | NSDiamond | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| VerifySouthTee | South Tee | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| VerifyTitleForm | Title | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| VerifyWestTee | WestTee | Verfies that all inputs are of the appropriate type and within required ranges. Prompts the user to re-enter incorrect inputs. Determines the traffic volumes to be used in the CAL3QHC input file. | n/a |
| WriteCAL3QHCIN | Module1 | Uses the built CALInputLine array to write to the "incal3qhc.in" text file. | n/a |

# Emission Factor Data Tables

COFL 2012 reads emission factors from the .txt files included in the *EFTextFiles* folder for each of the seven FDOT districts. The macro-enabled companion spreadsheet *EFTableGenerator.xlsm* provides a simple method for updating these text files to incorporate future MOVES runs, following EPA updates to their databases. The companion spreadsheet has a total of 14 tabs. The tabs labeled “1” to “7” each have three sections. The top section, as seen below, can be updated the EFs from future runs of MOVES. The middle section is a complete table of EFs that includes values that are the result of a double linear interpolation from the top table. The values in the second section are propagated when the macro “AAA” is run from the Developer Tab. Below the complete table are the same values transposed into a very long column. The long column of values is in the form of the .txt files. This long column is automatically copied to the tabs labeled “Dist1EF.txt” to “Dist7EF.txt”.



The following provides a simple way to update the COFL 2012 EF .txt files, once future runs of MOVES have been made that utilize database updates. Of course, there are other methods of performing these updates that may be preferred.

1. Perform MOVES runs and transfer the results to each of the seven “Values from MOVES Runs” tables on the tabs labeled “1” to “7”.
2. Run the macro “AAA” on the Developer Tab on each of the seven tabs.

* On each of the seven tabs labeled “Dist1EF.txt” to “Dist7EF.txt”, perform a “Save As”. click on “Other Formats” and then scroll down and select “Tab Delimited”.
* Save the file with the same name as found on the tab.
* Excel will prompt “The selected file type does not support workbooks that contain multiple sheets.” Click “OK”.
* Excel will then make a prompt regarding format compatibility. Click “Yes”.

1. The .txt files just created may now be copied into the folder *EFTextFiles* to replace the older versions. No other changes to the COFL 2012 program are required.

# Program Code

## Module 1

Imports System

Imports System.IO

Imports System.Collections

Module Module1

'General Global Variables

Public PassFail As Boolean

Public StreamToDisplay As StreamReader

Public PrimaryDirectory As String

Public TestMsg As String

Public EFArray(65, 40) As Single

Public InputsCorrect(4) As Boolean

Public QuickVerify As Boolean = False

Public EWTollBooth As Boolean = True

Public IFR(150) As String

Public sAppPath As String

Public NumInputLines As Integer

Public LineOfText As String

Public CALInputLine(111) As String

Public RandNum As Integer

'Global Constants

Public Const ZR = 6

Public Const ATIM = 60

Public Const VS = 0

Public Const VD = 0

Public Const SCAL = 0.3048

Public Const IOPT = 1

Public Const IDEBUG = 1

Public Const TYP = "'AG'"

Public Const HL = 0

Public Const YFAC = 3

Public Const SFR = 1600

Public Const SFRRamps = 1500

Public Const ST = 1

Public Const AT = 3

Public Const U = 1

Public Const BRG = 0

Public Const MIXH = 1000

Public Const AMB = 0

Public Const VAR = "'Y'"

Public Const DEGR = 5

Public Const VAL1 = 0

Public Const VAL2 = 71

Public Const NM = 1

Public Const PRINT2 = 0

Public Const MODE = "'CO'"

Public Const TPF = 0.6

Public Const RVP As Single = 13.3

'Multiplier Factors

Public MultAcc As Single

Public MultHwyOn As Single

Public MultTollBooth As Single

'Title Screen Global Variable Declarations

Public Job As String

Public FacilityName As String

Public UsersName As String

Public RunName As String

Public ProjectYear As String

Public LandUse As String

Public ZO As Integer

Public CLAS As Integer

Public CLASLetter As String

Public NL As Integer

Public COBack1Hr As String

Public COBack8Hr As String

Public Conc8(20) As Single

Public Conc1(20) As Single

'District Screen Global Variable Declarations

Public DistrictNum As Integer

Public Temperature As String

'IntersectionType Screen Global Variable Declarations

Public IntType As String

Public XR(20) As Integer

Public YR(20) As Integer

Public NR As Integer

'Intersection Data Global Variable Declarations

Public SNB As String

Public ATNB As String

Public SEB As String

Public ATEB As String

Public SSB As String

Public ATSB As String

Public SWB As String

Public ATWB As String

Public ATMax As Integer

Public ATLeft As Integer

Public ATRight As Integer

Public ATQ As Integer

Public FTMax As Integer

Public FTLeft As Integer

Public FTRight As Integer

Public Speed As Integer

Public HwySpeed As Integer

'Variables Unique to Diamonds

Public ORNB As String

Public OREB As String

Public ORSB As String

Public ORWB As String

Public FTDep As Integer

Public ATDep As Integer

'Variables Unique to Tollbooths

Public ETCSB As String

Public ETCWB As String

Public ETCNB As String

Public ETCEB As String

'CO FL 2012 input and output file names

Public FDOTin As String

Public FDOTout As String

'Variables used to extract and interpolate EF's to be used in CAL3QHC input file

Public EFNB As Single

Public EFEB As Single

Public EFSB As Single

Public EFWB As Single

Public EFFree As Single

Public EFFreeAcc As Single

Public EFTollAcc As Single

Public EF10mph As Single

Public EFHwy As Single

Public EFHwyOn As Single

Public EFId As Single

Public EFLt As Single

Public EFRt As Single

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sub MakeARun\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Public Sub MakeARun()

'Calls WriteCAL3QIN

WriteCAL3QHCIN()

'Calls GenerateOutput

GenerateOutput()

My.Forms.Run.Show()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sub ClearAll - Clears all inputs for a new user run\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Public Sub ClearAll()

Dim i As Integer

Dim TempIntType As String

TempIntType = IntType

Job = ""

FacilityName = ""

UsersName = ""

RunName = ""

ProjectYear = ""

LandUse = ""

ZO = 0

CLAS = 0

NL = 0

COBack1Hr = 0

DistrictNum = 0

IntType = ""

NR = 0

SNB = ""

ATNB = ""

SEB = ""

ATEB = ""

SSB = ""

ATSB = ""

SWB = ""

ATWB = ""

ATMax = 0

ATLeft = 0

ATRight = 0

ATQ = 0

FTMax = 0

FTLeft = 0

FTRight = 0

ORNB = ""

OREB = ""

ORSB = ""

ORWB = ""

ETCSB = ""

ETCWB = ""

ETCNB = ""

ETCEB = ""

For i = 1 To 20

XR(i) = 0

YR(i) = 0

Next

Select Case TempIntType

Case "4 X 4"

Int4X4.Init4X4()

Case "4 X 6"

Int4X6.Init4X6()

Case ("6 X 4")

Int6X4.Init6X4()

Case ("6 X 6")

Int6X6.Init6X6()

Case "East Tee"

EastTee.InitEastTee()

Case ("North Tee")

NorthTee.InitNorthTee()

Case ("Toll Booth")

TollBooth.InitTollBooth()

Case ("West Tee")

WestTee.InitWestTee()

Case ("South Tee")

SouthTee.InitSouthTee()

Case ("N-S Diamond")

NSDiamond.InitNSDiamond()

Case ("E-W Diamond")

I11.InitEWDiamond()

End Select

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sub FillEFArray - Responds to call from District Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub FillEFArray()

Dim i As Integer

Dim j As Integer

'Opens the Appropriate EF File for the Selected District

Directory.SetCurrentDirectory(PrimaryDirectory)

Select Case DistrictNum

Case 1

StreamToDisplay = New StreamReader("EFTextFiles\Dist1EF.txt")

Case 2

StreamToDisplay = New StreamReader("EFTextFiles\Dist2EF.txt")

Case 3

StreamToDisplay = New StreamReader("EFTextFiles\Dist3EF.txt")

Case 4

StreamToDisplay = New StreamReader("EFTextFiles\Dist4EF.txt")

Case 5

StreamToDisplay = New StreamReader("EFTextFiles\Dist5EF.txt")

Case 6

StreamToDisplay = New StreamReader("EFTextFiles\Dist6EF.txt")

Case 7

StreamToDisplay = New StreamReader("EFTextFiles\Dist7EF.txt")

End Select

'Reads EF's from the Appropriate District Input File

LineOfText = StreamToDisplay.ReadLine() 'First input line contains unused variable, District Name

Do Until StreamToDisplay.EndOfStream

For i = 0 To 13

For j = 0 To 40

LineOfText = StreamToDisplay.ReadLine()

EFArray(i, j) = LineOfText

Next j

Next i

Loop

StreamToDisplay.Close()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sub GetEFs - Extracts EFs for the given district from the appropriate text file\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub GetEFs()

Dim x As Single

Dim TempSingle As Single

'Gets EFId, EF10mph, EFRt, and EFLt

EFId = EFArray(0, (ProjectYear - 2010))

EF10mph = EFArray(10 / 5, (ProjectYear - 2010))

EFRt = EFArray(15 / 5, (ProjectYear - 2010))

EFLt = EFArray(20 / 5, (ProjectYear - 2010))

'Gets other needed EFs

If IntType <> "South Tee" Then

EFSB = EFArray((CInt(SSB) / 5), (ProjectYear - 2010))

End If

If IntType <> "West Tee" Then

EFWB = EFArray((CInt(SWB) / 5), (ProjectYear - 2010))

End If

If IntType <> "North Tee" Then

EFNB = EFArray((CInt(SNB) / 5), (ProjectYear - 2010))

End If

If IntType <> "East Tee" Then

EFEB = EFArray((CInt(SEB) / 5), (ProjectYear - 2010))

End If

'Choses largest EF of free flow link values

EFFree = 0

If IntType <> "South Tee" Then

If EFFree < EFSB Then

EFFree = EFSB

Speed = CInt(SSB)

End If

End If

If IntType <> "West Tee" Then

If EFFree < EFWB Then

EFFree = EFWB

Speed = CInt(SWB)

End If

End If

If IntType <> "North Tee" Then

If EFFree < EFNB Then

EFFree = EFNB

Speed = CInt(SNB)

End If

End If

If IntType <> "East Tee" Then

If EFFree < EFEB Then

EFFree = EFEB

Speed = CInt(SEB)

End If

End If

'Calculates Multipliers

'First calculates as a function of speed

MultAcc = (-0.0007 \* Speed ^ 3) + (0.0417 \* Speed ^ 2) + (0.4407 \* Speed) - 23.493

MultTollBooth = (0.0015 \* Speed ^ 3) - (0.2408 \* Speed ^ 2) + (12.408 \* Speed) - 191.41

'Then adjusts for the year

If CInt(ProjectYear) < 2025 Then

TempSingle = MultAcc

MultAcc = TempSingle \* ((0.014 \* ProjectYear) - 27.213)

TempSingle = MultTollBooth

MultTollBooth = TempSingle \* ((0.014 \* ProjectYear) - 27.213)

Else

TempSingle = MultAcc

MultAcc = TempSingle \* 1.137

TempSingle = MultTollBooth

MultTollBooth = TempSingle \* 1.137

End If

'Then adjusts for fraction of traffic on arterial links

x = (1 / 2) \* (6 \* 5280 / 3600) \* (Speed / 6) ^ 2

TempSingle = MultAcc

MultAcc = (75 / 120) \* (x / 1000) \* (TempSingle) + (1) \* (1 - (75 / 120) \* (x / 1000))

If MultTollBooth < 1 Then MultTollBooth = 1

If MultAcc < 1 Then MultAcc = 1

'Adjusts EF's with multiplier factors

EFFreeAcc = EFFree \* MultAcc

EFTollAcc = EFFree \* MultTollBooth

'Rounds EF's to the .1 Place

EFFree = (Fix(EFFree \* 10 + 0.5)) / 10

EFFreeAcc = (Fix(EFFreeAcc \* 10 + 0.5)) / 10

EFTollAcc = (Fix(EFTollAcc \* 10 + 0.5)) / 10

EFId = (Fix(EFId \* 10 + 0.5)) / 10

EFLt = (Fix(EFLt \* 10 + 0.5)) / 10

EFRt = (Fix(EFRt \* 10 + 0.5)) / 10

EF10mph = (Fix(EF10mph \* 10 + 0.5)) / 10

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*Sub GetDiamondEFs - Extracts EFs for the given district from the appropriate text file\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub GetDiamondEFs()

Dim x As Single

Dim TempSingle As Single

'Gets EFId, EFRt, and EFLt

EFId = EFArray(0, (ProjectYear - 2010))

EFRt = EFArray(15 / 5, (ProjectYear - 2010))

EFLt = EFArray(20 / 5, (ProjectYear - 2010))

'Gets other needed EFs

EFSB = EFArray((CInt(SSB) / 5), (ProjectYear - 2010))

EFWB = EFArray((CInt(SWB) / 5), (ProjectYear - 2010))

EFNB = EFArray((CInt(SNB) / 5), (ProjectYear - 2010))

EFEB = EFArray((CInt(SEB) / 5), (ProjectYear - 2010))

'Chooses the largest EF for FreeFlow and HwyFlow

EFFree = 0

EFHwy = 0

If IntType = "N-S Diamond" Then

If EFFree < EFWB Then

EFFree = EFWB

Speed = CInt(SWB)

End If

If EFFree < EFEB Then

EFFree = EFEB

Speed = CInt(SEB)

End If

If EFHwy < EFSB Then

EFHwy = EFSB

HwySpeed = CInt(SSB)

End If

If EFHwy < EFNB Then

EFHwy = EFNB

HwySpeed = CInt(SNB)

End If

Else

If EFHwy < EFWB Then

EFHwy = EFWB

HwySpeed = CInt(SWB)

End If

If EFHwy < EFEB Then

EFHwy = EFEB

HwySpeed = CInt(SEB)

End If

If EFFree < EFSB Then

EFFree = EFSB

Speed = CInt(SSB)

End If

If EFFree < EFNB Then

EFFree = EFNB

Speed = CInt(SNB)

End If

End If

'Calculates Multipliers

'First calculates as a function of speed

MultAcc = (-0.0007 \* Speed ^ 3) + (0.0417 \* Speed ^ 2) + (0.4407 \* Speed) - 23.493

MultHwyOn = (0.0013 \* HwySpeed ^ 3) - (0.2099 \* HwySpeed ^ 2) + (10.74 \* HwySpeed) - 161.77

'Then adjusts for the year

If CInt(ProjectYear) < 2025 Then

TempSingle = MultAcc

MultAcc = TempSingle \* ((0.014 \* ProjectYear) - 27.213)

TempSingle = MultHwyOn

MultHwyOn = TempSingle \* ((0.014 \* ProjectYear) - 27.213)

Else

TempSingle = MultAcc

MultAcc = TempSingle \* 1.137

TempSingle = MultHwyOn

MultHwyOn = TempSingle \* 1.137

End If

'Then adjusts for fraction of traffic on arterial links

x = (1 / 2) \* (6 \* 5280 / 3600) \* (Speed / 6) ^ 2

TempSingle = MultAcc

MultAcc = (75 / 120) \* (x / 1000) \* (TempSingle) + (1) \* (1 - (75 / 120) \* (x / 1000))

If MultHwyOn < 1 Then MultHwyOn = 1

If MultAcc < 1 Then MultAcc = 1

'Adjusts EF's with multiplier factors

EFFreeAcc = EFFree \* MultAcc

EFHwyOn = EFHwy \* MultHwyOn

'Rounds EF's to the .1 Place

EFFreeAcc = (Fix(EFFreeAcc \* 10 + 0.5)) / 10

EFLt = (Fix(EFLt \* 10 + 0.5)) / 10

EFRt = (Fix(EFRt \* 10 + 0.5)) / 10

EFHwy = (Fix(EFHwy \* 10 + 0.5)) / 10

EFHwyOn = (Fix(EFHwyOn \* 10 + 0.5)) / 10

EFId = (Fix(EFId \* 10 + 0.5)) / 10

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Writes CAL3QHC Input File\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub WriteCAL3QHCIN()

Directory.SetCurrentDirectory(PrimaryDirectory)

'Writes CALInputLine Array out to "incal3qhc.in"

Directory.SetCurrentDirectory(PrimaryDirectory)

Dim sWriter As IO.StreamWriter = New IO.StreamWriter("incal3qhc.in")

Dim i As Integer

For i = 1 To NumInputLines

LineOfText = CALInputLine(i)

sWriter.WriteLine(LineOfText)

Next

sWriter.Flush()

sWriter.Close()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Reads CAL3QHC Output File\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub ReadCAL3QHCOUT()

Dim i As Integer

Dim Delay As Integer = 1000000000

Dim ReadLine(220) As String

Dim PPMString As String

PassFail = True

Directory.SetCurrentDirectory(PrimaryDirectory)

'Opens the CAL3QHC output file

For i = 1 To Delay

Next

Try

StreamToDisplay = New StreamReader("outcal3qhc.out")

Catch

For i = 1 To Delay

Next i

Try

StreamToDisplay = New StreamReader("outcal3qhc.out")

Catch

Exit Sub

End Try

End Try

'Reads the CAL3QHC output file into the array ReadLine

i = 1

Do Until StreamToDisplay.EndOfStream

ReadLine(i) = StreamToDisplay.ReadLine()

i = i + 1

Loop

StreamToDisplay.Close()

PPMString = ReadLine(i - 4)

'Reads 1 hr. concentration values from outcal3qhc.out and adds in background CO

If (IntType <> "East Tee" And IntType <> "North Tee" And IntType <> "West Tee" And IntType <> "South Tee") Then

For i = 20 To 18 Step -1

Conc1(i) = CSng(Microsoft.VisualBasic.Right(PPMString, 6))

Conc1(i) = Conc1(i) + CSng(COBack1Hr)

Conc1(i) = Format(Conc1(i), "#0.0")

PPMString = Microsoft.VisualBasic.Left(PPMString, (i \* 6 + 3))

Next i

End If

For i = 17 To 1 Step -1

Conc1(i) = CSng(Microsoft.VisualBasic.Right(PPMString, 6))

Conc1(i) = Conc1(i) + CSng(COBack1Hr)

Conc1(i) = Format(Conc1(i), "#0.0")

PPMString = Microsoft.VisualBasic.Left(PPMString, (i \* 6 + 3))

Next i

'Calculations on receptors 1-17

For i = 1 To 17

'Rounds to .1

Conc1(i) = (Fix(Conc1(i) \* 10 + 0.5)) / 10

'Checks for 1 hr exceedances

If Conc1(i) >= 35 Then PassFail = False

'Calculates 8 hr concentrations via Total Persistence Factor

Conc8(i) = (Fix((Conc1(i) \* TPF) \* 10 + 0.5)) / 10

'Checsk for 8 hr exceedances

If Conc8(i) >= 9 Then PassFail = False

Next i

If (IntType = "East Tee" Or IntType = "South Tee" Or IntType = "West Tee" Or IntType = "North Tee") Then

Exit Sub

Else

'Calculations on receptors 18-20

For i = 18 To 20

'Rounds to .1

Conc1(i) = (Fix(Conc1(i) \* 10 + 0.5)) / 10

'Checks for 1 hr exceedancesF

If Conc1(i) >= 35 Then PassFail = False

'Calculates 8 hr concentrations via Total Persistence Factor

Conc8(i) = (Fix((Conc1(i) \* TPF) \* 10 + 0.5)) / 10

'Checsk for 8 hr exceedances

If Conc8(i) >= 9 Then PassFail = False

Next i

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Runs CAL3QHC and then Reads and Formats Output to Screen\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub GenerateOutput()

'Runs CAL3QHC

Directory.SetCurrentDirectory(PrimaryDirectory)

'sAppPath = "CAL3QHC incal3qhc" & RunNumber & ".in outcal3qhc" & RunNumber & ".out"

sAppPath = "CAL3QHC incal3qhc.in outcal3qhc.out"

Shell(sAppPath, vbMaximizedFocus)

'Reads resulting concentrations from outcal3qhc.out

ReadCAL3QHCOUT()

'Fills out onscreen output screen

Run.TextBox2.Text = Format$(Now, "Long Date")

Run.TextBox10.Text = Job

Run.TextBox8.Text = FacilityName

Run.TextBox6.Text = UsersName

Run.TextBox4.Text = RunName

Run.TextBox26.Text = DistrictNum

Run.TextBox28.Text = ProjectYear

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

Run.TextBox30.Text = "E-W Freeway " & IntType

Else

Run.TextBox30.Text = "N-S Freeway " & IntType

End If

Else

Run.TextBox30.Text = IntType

End If

Run.TextBox38.Text = "Speed"

Run.TextBox37.Text = "Approach Traffic"

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

Run.TextBox30.Text = "E-W Freeway " & IntType

Else

Run.TextBox30.Text = "N-S Freeway " & IntType

End If

Else

Run.TextBox30.Text = IntType

End If

Run.TextBox38.Text = "Speed"

Run.TextBox37.Text = "Approach Traffic"

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

Run.TextBox38.Text = "East Bound"

Run.TextBox37.Text = "EB Stopping"

Run.TextBox40.Text = SEB & " mph"

Run.TextBox39.Text = CInt(ATEB \* (100 - ETCEB) / 100) & " vph"

Run.TextBox34.Text = "West Bound"

Run.TextBox42.Text = "WB Stopping"

Run.TextBox36.Text = SWB & " mph"

Run.TextBox41.Text = CInt(ATWB \* (100 - ETCWB) / 100) & " vph"

If ETCEB > 0 Then

Run.TextBox32.Text = "EB ETC-only"

Run.TextBox35.Text = CInt(ATEB \* ETCEB / 100) & " vph"

Else

Run.TextBox32.Text = ""

Run.TextBox35.Text = ""

End If

If ETCWB > 0 Then

Run.TextBox44.Text = "WB ETC-only"

Run.TextBox43.Text = CInt(ATWB \* ETCWB / 100) & " vph"

Else

Run.TextBox44.Text = ""

Run.TextBox43.Text = ""

End If

Else

Run.TextBox38.Text = "North Bound"

Run.TextBox37.Text = "NB Stopping"

Run.TextBox40.Text = SNB & " mph"

Run.TextBox39.Text = CInt(ATNB \* (100 - ETCNB) / 100) & " vph"

Run.TextBox34.Text = "South Bound"

Run.TextBox42.Text = "SB Stopping"

Run.TextBox36.Text = SSB & " mph"

Run.TextBox41.Text = CInt(ATSB \* (100 - ETCSB) / 100) & " vph"

If ETCNB > 0 Then

Run.TextBox32.Text = "NB ETC-only"

Run.TextBox35.Text = CInt(ATNB \* ETCNB / 100) & " vph"

Else

Run.TextBox32.Text = ""

Run.TextBox35.Text = ""

End If

If ETCSB > 0 Then

Run.TextBox44.Text = "SB ETC-only"

Run.TextBox43.Text = CInt(ATSB \* ETCSB / 100) & " vph"

Else

Run.TextBox44.Text = ""

Run.TextBox43.Text = ""

End If

End If

ElseIf (IntType = "N-S Diamond" Or IntType = "E-W Diamond") Then

Run.TextBox38.Text = "Arterial"

Run.TextBox34.Text = "Freeway"

Run.TextBox37.Text = "Arterial"

Run.TextBox42.Text = "Freeway"

Run.TextBox40.Text = Speed & " mph"

Run.TextBox36.Text = HwySpeed & " mph"

Run.TextBox39.Text = ATMax & " vph"

Run.TextBox41.Text = FTMax & " vph"

Run.TextBox32.Text = ""

Run.TextBox35.Text = ""

Run.TextBox44.Text = ""

Run.TextBox43.Text = ""

Else

Run.TextBox38.Text = "Arterial"

Run.TextBox40.Text = Speed & " mph"

Run.TextBox37.Text = "Arterial"

Run.TextBox39.Text = ATMax & " vph"

Run.TextBox34.Text = ""

Run.TextBox36.Text = ""

Run.TextBox44.Text = ""

Run.TextBox42.Text = ""

Run.TextBox41.Text = ""

Run.TextBox32.Text = ""

Run.TextBox35.Text = ""

Run.TextBox44.Text = ""

Run.TextBox43.Text = ""

End If

Run.TextBox14.Text = Temperature & " °F"

Run.TextBox16.Text = RVP & " psi"

Run.TextBox18.Text = LandUse

Run.TextBox20.Text = CLASLetter

Run.TextBox22.Text = ZO & " cm"

Run.TextBox24.Text = Format(CSng(COBack1Hr), "#0.0") & " ppm"

Run.TextBox173.Text = Format(CSng(COBack8Hr), "#0.0") & " ppm"

'Outputs for intersections with 20 receptors or leaves blanks for the Tee Intersections with 17 receptors

If (IntType = "East Tee" Or IntType = "North Tee" Or IntType = "West Tee" Or IntType = "South Tee") Then

Run.TextBox125.Text = ""

Run.TextBox124.Text = ""

Run.TextBox123.Text = ""

Run.TextBox115.Text = ""

Run.TextBox114.Text = ""

Run.TextBox113.Text = ""

Run.TextBox105.Text = ""

Run.TextBox104.Text = ""

Run.TextBox103.Text = ""

Else

Run.TextBox125.Text = "18"

Run.TextBox124.Text = "19"

Run.TextBox123.Text = "20"

Run.TextBox115.Text = Format(Conc1(18), "#0.0")

Run.TextBox114.Text = Format(Conc1(19), "#0.0")

Run.TextBox113.Text = Format(Conc1(20), "#0.0")

Run.TextBox105.Text = Format(Conc8(18), "#0.0")

Run.TextBox104.Text = Format(Conc8(19), "#0.0")

Run.TextBox103.Text = Format(Conc8(20), "#0.0")

End If

Run.TextBox152.Text = Format(Conc1(1), "#0.0")

Run.TextBox151.Text = Format(Conc1(2), "#0.0")

Run.TextBox150.Text = Format(Conc1(3), "#0.0")

Run.TextBox149.Text = Format(Conc1(4), "#0.0")

Run.TextBox148.Text = Format(Conc1(5), "#0.0")

Run.TextBox147.Text = Format(Conc1(6), "#0.0")

Run.TextBox146.Text = Format(Conc1(7), "#0.0")

Run.TextBox145.Text = Format(Conc1(8), "#0.0")

Run.TextBox144.Text = Format(Conc1(9), "#0.0")

Run.TextBox143.Text = Format(Conc1(10), "#0.0")

Run.TextBox122.Text = Format(Conc1(11), "#0.0")

Run.TextBox121.Text = Format(Conc1(12), "#0.0")

Run.TextBox120.Text = Format(Conc1(13), "#0.0")

Run.TextBox119.Text = Format(Conc1(14), "#0.0")

Run.TextBox118.Text = Format(Conc1(15), "#0.0")

Run.TextBox117.Text = Format(Conc1(16), "#0.0")

Run.TextBox116.Text = Format(Conc1(17), "#0.0")

Run.TextBox142.Text = Format(Conc8(1), "#0.0")

Run.TextBox141.Text = Format(Conc8(2), "#0.0")

Run.TextBox140.Text = Format(Conc8(3), "#0.0")

Run.TextBox139.Text = Format(Conc8(4), "#0.0")

Run.TextBox138.Text = Format(Conc8(5), "#0.0")

Run.TextBox137.Text = Format(Conc8(6), "#0.0")

Run.TextBox136.Text = Format(Conc8(7), "#0.0")

Run.TextBox135.Text = Format(Conc8(8), "#0.0")

Run.TextBox134.Text = Format(Conc8(9), "#0.0")

Run.TextBox133.Text = Format(Conc8(10), "#0.0")

Run.TextBox112.Text = Format(Conc8(11), "#0.0")

Run.TextBox111.Text = Format(Conc8(12), "#0.0")

Run.TextBox110.Text = Format(Conc8(13), "#0.0")

Run.TextBox109.Text = Format(Conc8(14), "#0.0")

Run.TextBox108.Text = Format(Conc8(15), "#0.0")

Run.TextBox107.Text = Format(Conc8(16), "#0.0")

Run.TextBox106.Text = Format(Conc8(17), "#0.0")

'Prints pass/fail message

If PassFail = False Then

Run.TextBox171.Visible = True

Run.TextBox169.Visible = False

Else

Run.TextBox169.Visible = True

Run.TextBox171.Visible = False

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - 4X4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Build4X4()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'4X4-Specific Parameters

NR = 20

NumInputLines = 71

NL = 20

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To (21)

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'South Bound Approach Queue Link

CALInputLine(23) = 2

CALInputLine(24) = "'SB Queue Link'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & 3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(25) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'South Bound Left Queue Link

CALInputLine(26) = 2

CALInputLine(27) = "'SB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & 0 & " , " & 3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(28) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Approach Queue Link

CALInputLine(29) = 2

CALInputLine(30) = "'WB Queue Link'" & "," & TYP & ", " & 30 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Left Queue Link

CALInputLine(32) = 2

CALInputLine(33) = "'WB Left Queue Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(34) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Approach Queue Link

CALInputLine(35) = 2

CALInputLine(36) = "'NB Queue Link'" & "," & TYP & ", " & 18 & " , " & -30 & " , " & 18 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(37) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Left Queue Link

CALInputLine(38) = 2

CALInputLine(39) = "'NB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 0 & " , " & -3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(40) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Approach Queue Link

CALInputLine(41) = 2

CALInputLine(42) = "'EB Queue Link'" & "," & TYP & ", " & -30 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(43) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Left Queue Link

CALInputLine(44) = 2

CALInputLine(45) = "'EB Left Queue Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(46) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Approach Link

CALInputLine(47) = 1

CALInputLine(48) = "'SB Approach Link'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & 3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'SB Left Link

CALInputLine(49) = 1

CALInputLine(50) = "'SB Left Link'" & "," & TYP & ", " & 0 & " , " & 0 & " , " & 0 & " , " & 3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'SB Departure Link

CALInputLine(51) = 1

CALInputLine(52) = "'SB Departure Link'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & 18 & " , " & 3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all WEST Bound Free Flow Links

'WB Approach Link

CALInputLine(53) = 1

CALInputLine(54) = "'WB Approach Link'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'WB Left Link

CALInputLine(55) = 1

CALInputLine(56) = "'WB Left Link'" & "," & TYP & ", " & 0 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'WB Departure Link

CALInputLine(57) = 1

CALInputLine(58) = "'WB Departure Link'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & 3000 & " , " & -18 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all NORTH Bound Free Flow Links

'NB Approach Link

CALInputLine(59) = 1

CALInputLine(60) = "'NB Approach Link'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & 18 & " , " & -3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'NB Left Link

CALInputLine(61) = 1

CALInputLine(62) = "'NB Left Link'" & "," & TYP & ", " & 0 & " , " & 0 & " , " & 0 & " , " & -3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'NB Departure Link

CALInputLine(63) = 1

CALInputLine(64) = "'NB Departure Link'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & -3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all EAST Bound Free Flow Links

'EB Approach Link

CALInputLine(65) = 1

CALInputLine(66) = "'EB Approach Link'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'EB Left Link

CALInputLine(67) = 1

CALInputLine(68) = "'EB Left Link'" & "," & TYP & ", " & 0 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'EB Departure Link

CALInputLine(69) = 1

CALInputLine(70) = "'EB Departure Link'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & -3000 & " , " & 18 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(71) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - 4X6\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Build4X6()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATSB > ATMax Then ATMax = ATSB

If ATWB > ATMax Then ATMax = ATWB

If ATNB > ATMax Then ATMax = ATNB

If ATEB > ATMax Then ATMax = ATEB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'4X6 -Specific Parameters

NR = 20

NumInputLines = 71

NL = 20

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To (21)

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'West Bound Approach Queue Link

CALInputLine(23) = 2

CALInputLine(24) = "'WB Queue Link'" & "," & TYP & ", " & 48 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(25) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Left Queue Link

CALInputLine(26) = 2

CALInputLine(27) = "'WB Left Queue Link'" & "," & TYP & ", " & 48 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(28) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Approach Queue Link

CALInputLine(29) = 2

CALInputLine(30) = "'NB Queue Link'" & "," & TYP & ", " & 3000 & " , " & -30 & " , " & 0 & " , " & 3 - 0 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(31) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Left Queue Link

CALInputLine(32) = 2

CALInputLine(33) = "'NB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 0 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(34) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Approach Queue Link

CALInputLine(35) = 2

CALInputLine(36) = "'EB Queue Link'" & "," & TYP & ", " & -48 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(37) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Left Queue Link

CALInputLine(38) = 2

CALInputLine(39) = "'EB Left Queue Link'" & "," & TYP & ", " & -48 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(40) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'South Bound Approach Queue Link

CALInputLine(41) = 2

CALInputLine(42) = "'SB Queue Link'" & "," & TYP & ", " & -30 & " , " & 30 & " , " & -30 & " , " & 3000 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(43) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'South Bound Left Queue Link

CALInputLine(44) = 2

CALInputLine(45) = "'SB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & 0 & " , " & 3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(46) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all WEST Bound Free Flow Links

'WB Approach Link

CALInputLine(47) = 1

CALInputLine(48) = "'WB Approach Link'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'WB Left Link

CALInputLine(49) = 1

CALInputLine(50) = "'WB Left Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'WB Departure Link

CALInputLine(51) = 1

CALInputLine(52) = "'WB Departure Link'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & 3000 & " , " & -18 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all NORTH Bound Free Flow Links

'NB Approach Link

CALInputLine(53) = 1

CALInputLine(54) = "'NB Approach Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 30 & " , " & -3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'NB Left Link

CALInputLine(55) = 1

CALInputLine(56) = "'NB Left Link'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & 0 & " , " & -3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'NB Departure Link

CALInputLine(57) = 1

CALInputLine(58) = "'NB Departure Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -30 & " , " & -3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'NEXT, Write all EAST Bound Free Flow Links

'EB Approach Link

CALInputLine(59) = 1

CALInputLine(60) = "'EB Approach Link'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'EB Left Link

CALInputLine(61) = 1

CALInputLine(62) = "'EB Left Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'EB Departure Link

CALInputLine(63) = 1

CALInputLine(64) = "'EB Departure Link'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & -3000 & " , " & 18 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Approach Link

CALInputLine(65) = 1

CALInputLine(66) = "'SB Approach Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -30 & " , " & 3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'SB Left Link

CALInputLine(67) = 1

CALInputLine(68) = "'SB Left Link'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & 0 & " , " & 3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'SB Departure Link

CALInputLine(69) = 1

CALInputLine(70) = "'SB Departure Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 30 & " , " & 3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'Last Line

CALInputLine(71) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - 6X4\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Build6X4()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATSB > ATMax Then ATMax = ATSB

If ATWB > ATMax Then ATMax = ATWB

If ATNB > ATMax Then ATMax = ATNB

If ATEB > ATMax Then ATMax = ATEB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'6X4 -Specific Parameters

NR = 20

NumInputLines = 71

NL = 20

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To (21)

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'South Bound Approach Queue Link

CALInputLine(23) = 2

CALInputLine(24) = "'SB Queue Link'" & "," & TYP & ", " & -18 & " , " & 48 & " , " & -18 & " , " & 3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(25) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'South Bound Left Queue Link

CALInputLine(26) = 2

CALInputLine(27) = "'SB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & 48 & " , " & 0 & " , " & 3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(28) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Approach Queue Link

CALInputLine(29) = 2

CALInputLine(30) = "'WB Queue Link'" & "," & TYP & ", " & 30 & " , " & 30 & " , " & 3000 & " , " & 30 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(31) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Left Queue Link

CALInputLine(32) = 2

CALInputLine(33) = "'WB Left Queue Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(34) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Approach Queue Link

CALInputLine(35) = 2

CALInputLine(36) = "'NB Queue Link'" & "," & TYP & ", " & 18 & " , " & -48 & " , " & 18 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(37) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Left Queue Link

CALInputLine(38) = 2

CALInputLine(39) = "'NB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & -48 & " , " & 0 & " , " & -3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(40) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Approach Queue Link

CALInputLine(41) = 2

CALInputLine(42) = "'EB Queue Link'" & "," & TYP & ", " & -30 & " , " & -30 & " , " & -3000 & " , " & -30 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(43) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Left Queue Link

CALInputLine(44) = 2

CALInputLine(45) = "'EB Left Queue Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(46) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Approach Link

CALInputLine(47) = 1

CALInputLine(48) = "'SB Approach Link'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & 3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'SB Left Link

CALInputLine(49) = 1

CALInputLine(50) = "'SB Left Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 0 & " , " & 3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'SB Departure Link

CALInputLine(51) = 1

CALInputLine(52) = "'SB Departure Link'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & 18 & " , " & 3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all WEST Bound Free Flow Links

'WB Approach Link

CALInputLine(53) = 1

CALInputLine(54) = "'WB Approach Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & 3000 & " , " & 30 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'WB Left Link

CALInputLine(55) = 1

CALInputLine(56) = "'WB Left Link'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'WB Departure Link

CALInputLine(57) = 1

CALInputLine(58) = "'WB Departure Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 3000 & " , " & -30 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'NEXT, Write all NORTH Bound Free Flow Links

'NB Approach Link

CALInputLine(59) = 1

CALInputLine(60) = "'NB Approach Link'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & 18 & " , " & -3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 44

'NB Left Link

CALInputLine(61) = 1

CALInputLine(62) = "'NB Left Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & 0 & " , " & -3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 32

'NB Departure Link

CALInputLine(63) = 1

CALInputLine(64) = "'NB Departure Link'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & -3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all EAST Bound Free Flow Links

'EB Approach Link

CALInputLine(65) = 1

CALInputLine(66) = "'EB Approach Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & -3000 & " , " & -30 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'EB Left Link

CALInputLine(67) = 1

CALInputLine(68) = "'EB Left Link'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'EB Departure Link

CALInputLine(69) = 1

CALInputLine(70) = "'EB Departure Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & -3000 & " , " & 30 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'Last Line

CALInputLine(71) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - 6X6\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Build6X6()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATSB > ATMax Then ATMax = ATSB

If ATWB > ATMax Then ATMax = ATWB

If ATNB > ATMax Then ATMax = ATNB

If ATEB > ATMax Then ATMax = ATEB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'6X6 -Specific Parameters

NR = 20

NumInputLines = 71

NL = 20

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To (21)

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'South Bound Approach Queue Link

CALInputLine(23) = 2

CALInputLine(24) = "'SB Queue Link'" & "," & TYP & ", " & -30 & " , " & 48 & " , " & -30 & " , " & 3000 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(25) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'South Bound Left Queue Link

CALInputLine(26) = 2

CALInputLine(27) = "'SB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & 48 & " , " & 0 & " , " & 3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(28) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Approach Queue Link

CALInputLine(29) = 2

CALInputLine(30) = "'WB Queue Link'" & "," & TYP & ", " & 48 & " , " & 30 & " , " & 3000 & " , " & 30 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(31) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'West Bound Left Queue Link

CALInputLine(32) = 2

CALInputLine(33) = "'WB Left Queue Link'" & "," & TYP & ", " & 48 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(34) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Approach Queue Link

CALInputLine(35) = 2

CALInputLine(36) = "'NB Queue Link'" & "," & TYP & ", " & 30 & " , " & -48 & " , " & 30 & " , " & -3000 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(37) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'North Bound Left Queue Link

CALInputLine(38) = 2

CALInputLine(39) = "'NB Left Queue Link'" & "," & TYP & ", " & 0 & " , " & -48 & " , " & 0 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(40) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Approach Queue Link

CALInputLine(41) = 2

CALInputLine(42) = "'EB Queue Link'" & "," & TYP & ", " & -48 & " , " & -30 & " , " & -3000 & " , " & -30 & " , " & HL & " , " & 36 & " , " & 3

CALInputLine(43) = 120 & " , " & 75 & " , " & YFAC & " , " & ATQ & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'East Bound Left Queue Link

CALInputLine(44) = 2

CALInputLine(45) = "'EB Left Queue Link'" & "," & TYP & ", " & -48 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(46) = 120 & " , " & 105 & " , " & YFAC & " , " & ATLeft & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Approach Link

CALInputLine(47) = 1

CALInputLine(48) = "'SB Approach Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -30 & " , " & 3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'SB Left Link

CALInputLine(49) = 1

CALInputLine(50) = "'SB Left Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 0 & " , " & 3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'SB Departure Link

CALInputLine(51) = 1

CALInputLine(52) = "'SB Departure Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 30 & " , " & 3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'NEXT, Write all WEST Bound Free Flow Links

'WB Approach Link

CALInputLine(53) = 1

CALInputLine(54) = "'WB Approach Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & 3000 & " , " & 30 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'WB Left Link

CALInputLine(55) = 1

CALInputLine(56) = "'WB Left Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'WB Departure Link

CALInputLine(57) = 1

CALInputLine(58) = "'WB Departure Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 3000 & " , " & -30 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'NEXT, Write all NORTH Bound Free Flow Links

'NB Approach Link

CALInputLine(59) = 1

CALInputLine(60) = "'NB Approach Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 30 & " , " & -3000 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'NB Left Link

CALInputLine(61) = 1

CALInputLine(62) = "'NB Left Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 30 & " , " & -3000 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'NB Departure Link

CALInputLine(63) = 1

CALInputLine(64) = "'NB Departure Link'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -30 & " , " & -3000 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'NEXT, Write all EAST Bound Free Flow Links

'EB Approach Link

CALInputLine(65) = 1

CALInputLine(66) = "'EB Approach Link'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & -3000 & " , " & -30 & " , " & ATMax & " , " & EFFree & " , " & HL & " , " & 56

'EB Left Link

CALInputLine(67) = 1

CALInputLine(68) = "'EB Left Link'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & ATLeft & " , " & EFLt & " , " & HL & " , " & 44

'EB Departure Link

CALInputLine(69) = 1

CALInputLine(70) = "'EB Departure Link'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & -3000 & " , " & 30 & " , " & ATMax & " , " & EFFreeAcc & " , " & HL & " , " & 56

'Last Line

CALInputLine(71) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - EAST TEE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildEastTee()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATSB > ATMax Then ATMax = ATSB

If ATWB > ATMax Then ATMax = ATWB

If ATNB > ATMax Then ATMax = ATNB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'\*\*\*Note: For the tee configurations, the traffic queue and flow volumes are calculated within the individual input line statements

'\*\*\*relative to ATMax

'East Tee -Specific Parameters

NR = 17

NumInputLines = 51

NL = 13

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To 18

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(19) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'WB Leg Left Q

CALInputLine(20) = 2

CALInputLine(21) = "'WB Leg Left Q'" & "," & TYP & ", " & 30 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(22) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Right Q

CALInputLine(23) = 2

CALInputLine(24) = "'WB Leg Right Q'" & "," & TYP & ", " & 30 & " , " & 24 & " , " & 3000 & " , " & 24 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(25) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Thru Q

CALInputLine(26) = 2

CALInputLine(27) = "'NB Leg Thru Q'" & "," & TYP & ", " & 18 & " , " & -36 & " , " & 18 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(28) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 1) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Thru Q

CALInputLine(29) = 2

CALInputLine(30) = "'SB Leg Thru Q'" & "," & TYP & ", " & -18 & " , " & 36 & " , " & -18 & " , " & 3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.85) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Left Q

CALInputLine(32) = 2

CALInputLine(33) = "'SB Leg Left Q'" & "," & TYP & ", " & 0 & " , " & 36 & " , " & 0 & " , " & 3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(34) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.15) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all WEST Bound Free Flow Links

'WB Leg Left

CALInputLine(35) = 1

CALInputLine(36) = "'WB Leg Left'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & CInt(ATMax \* 0.5) & " , " & EFLt & " , " & HL & " , " & 44

'WB Leg Right

CALInputLine(37) = 1

CALInputLine(38) = "'WB Leg Right'" & "," & TYP & ", " & 18 & " , " & 24 & " , " & 3000 & " , " & 24 & " , " & CInt(ATMax \* 0.5) & " , " & EFRt & " , " & HL & " , " & 44

'NEXT, Write all NORTH Bound Free Flow Links

'NB Leg Thru

CALInputLine(39) = 1

CALInputLine(40) = "'NB Leg Thru '" & "," & TYP & ", " & 18 & " , " & 24 & " , " & 18 & " , " & -3000 & " , " & CInt(ATMax \* 0.75) & " , " & EFFree & " , " & HL & " , " & 44

'NB Leg Dep

CALInputLine(41) = 1

CALInputLine(42) = "'NB Leg Dep '" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & -3000 & " , " & CInt(ATMax \* 1.35) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all EAST Bound Free Flow Links

'EB Leg Dep

CALInputLine(43) = 1

CALInputLine(44) = "'EB Leg Dep'" & "," & TYP & ", " & 0 & " , " & -24 & " , " & 3000 & " , " & -24 & " , " & CInt(ATMax \* 0.4) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Leg Thru

CALInputLine(45) = 1

CALInputLine(46) = "'SB Leg Thru'" & "," & TYP & ", " & -18 & " , " & 0 & " , " & -18 & " , " & 3000 & " , " & CInt(ATMax \* 0.85) & " , " & EFFree & " , " & HL & " , " & 44

'SB Leg Left

CALInputLine(47) = 1

CALInputLine(48) = "'SB Leg Left'" & "," & TYP & ", " & 0 & " , " & -24 & " , " & 0 & " , " & 3000 & " , " & CInt(ATMax \* 0.15) & " , " & EFLt & " , " & HL & " , " & 32

'SB Leg Dep

CALInputLine(49) = 1

CALInputLine(50) = "'SB Leg Dep'" & "," & TYP & ", " & 18 & " , " & 24 & " , " & 18 & " , " & 3000 & " , " & CInt(ATMax \* 1.25) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(51) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - NORTH TEE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildNorthTee()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATSB > ATMax Then ATMax = ATSB

If ATWB > ATMax Then ATMax = ATWB

If ATEB > ATMax Then ATMax = ATEB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'\*\*\*Note: For the tee configurations, the traffic queue and flow volumes are calculated within the individual input line statements

'\*\*\*relative to ATMax

'NORTH Tee -Specific Parameters

NR = 17

NumInputLines = 51

NL = 13

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To 18

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(19) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'SB Leg Left Q

CALInputLine(20) = 2

CALInputLine(21) = "'SB Leg Left Q'" & "," & TYP & ", " & 0 & " , " & 30 & " , " & 0 & " , " & 3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(22) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Right Q

CALInputLine(23) = 2

CALInputLine(24) = "'SB Leg Right Q'" & "," & TYP & ", " & -24 & " , " & 30 & " , " & -24 & " , " & 3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(25) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Thru Q

CALInputLine(26) = 2

CALInputLine(27) = "'WB Leg Thru Q'" & "," & TYP & ", " & 36 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(28) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 1) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Thru Q

CALInputLine(29) = 2

CALInputLine(30) = "'EB Leg Thru Q'" & "," & TYP & ", " & -36 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.85) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Left Q

CALInputLine(32) = 2

CALInputLine(33) = "'EB Leg Left Q'" & "," & TYP & ", " & -36 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(34) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.15) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Leg Left

CALInputLine(35) = 1

CALInputLine(36) = "'SB Leg Left'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & 0 & " , " & 3000 & " , " & CInt(ATMax \* 0.5) & " , " & EFLt & " , " & HL & " , " & 44

'SB Leg Right

CALInputLine(37) = 1

CALInputLine(38) = "'SB Leg Right'" & "," & TYP & ", " & -24 & " , " & 18 & " , " & -24 & " , " & 3000 & " , " & CInt(ATMax \* 0.5) & " , " & EFRt & " , " & HL & " , " & 44

'NEXT, Write all WEST Bound Free Flow Links

'WB Leg Thru

CALInputLine(39) = 1

CALInputLine(40) = "'WB Leg Thru '" & "," & TYP & ", " & -24 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & CInt(ATMax \* 0.75) & " , " & EFFree & " , " & HL & " , " & 44

'WB Leg Dep

CALInputLine(41) = 1

CALInputLine(42) = "'WB Leg Dep '" & "," & TYP & ", " & 0 & " , " & -18 & " , " & 3000 & " , " & -18 & " , " & CInt(ATMax \* 1.35) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all NORTH Bound Free Flow Links

'NB Leg Dep

CALInputLine(43) = 1

CALInputLine(44) = "'SB Leg Dep'" & "," & TYP & ", " & 24 & " , " & 0 & " , " & 24 & " , " & 3000 & " , " & CInt(ATMax \* 0.4) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all EAST Bound Free Flow Links

'EB Leg Thru

CALInputLine(45) = 1

CALInputLine(46) = "'EB Leg Thru'" & "," & TYP & ", " & 0 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & CInt(ATMax \* 0.85) & " , " & EFFree & " , " & HL & " , " & 44

'EB Leg Left

CALInputLine(47) = 1

CALInputLine(48) = "'EB Leg Left'" & "," & TYP & ", " & 24 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & CInt(ATMax \* 0.15) & " , " & EFLt & " , " & HL & " , " & 32

'EB Leg Dep

CALInputLine(49) = 1

CALInputLine(50) = "'EB Leg Dep'" & "," & TYP & ", " & -24 & " , " & 18 & " , " & -3000 & " , " & 18 & " , " & CInt(ATMax \* 1.25) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(51) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - TOLL BOOTH\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildTollBooth()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'TollBooth - Specific Parameters

NR = 20

NumInputLines = 43

NL = 10

'Note: Tollbooth receptor arrays are filled here,

'all the other roadway configurations have their receptor arrays filled in the intersection type form.

'Fill Receptor Arrays

If EWTollBooth = True Then

'Fills Receptor Array for the E-W Orientation

'First, fills X-coords

XR(1) = -2000

XR(2) = -1250

XR(3) = -500

XR(4) = -150

XR(5) = -50

XR(6) = 50

XR(7) = 150

XR(8) = 500

XR(9) = 1250

XR(10) = 2000

XR(11) = 2000

XR(12) = 1250

XR(13) = 500

XR(14) = 150

XR(15) = 50

XR(16) = -50

XR(17) = -150

XR(18) = -500

XR(19) = -1250

XR(20) = -2000

'Next fills, Y-coords

If (CInt(ETCEB) > 0 Or CInt(ETCWB) > 0) Then

'Fills for the scenario that includes ETC-Only lanes

YR(1) = 68

YR(2) = 68

YR(3) = 116

YR(4) = 116

YR(5) = 116

YR(6) = 116

YR(7) = 116

YR(8) = 116

YR(9) = 68

YR(10) = 68

YR(11) = -68

YR(12) = -68

YR(13) = -116

YR(14) = -116

YR(15) = -116

YR(16) = -116

YR(17) = -116

YR(18) = -116

YR(19) = -68

YR(20) = -68

Else

'Fills for the scenario that does not inclue ETC-only lanes

YR(1) = 68

YR(2) = 68

YR(3) = 80

YR(4) = 80

YR(5) = 80

YR(6) = 80

YR(7) = 80

YR(8) = 80

YR(9) = 68

YR(10) = 68

YR(11) = -68

YR(12) = -68

YR(13) = -80

YR(14) = -80

YR(15) = -80

YR(16) = -80

YR(17) = -80

YR(18) = -80

YR(19) = -68

YR(20) = -68

End If

Else

'Fills Receptor Array for the N-S Orientation

'First, fills Y-coords

YR(1) = 2000

YR(2) = 1250

YR(3) = 500

YR(4) = 150

YR(5) = 50

YR(6) = -50

YR(7) = -150

YR(8) = -500

YR(9) = -1250

YR(10) = -2000

YR(11) = -2000

YR(12) = -1250

YR(13) = -500

YR(14) = -150

YR(15) = -50

YR(16) = 50

YR(17) = 150

YR(18) = 500

YR(19) = -1250

YR(20) = 2000

'Next fills, X-coords

If (CInt(ETCNB) > 0 Or CInt(ETCSB) > 0) Then

'Fills for the scenario that includes ETC-Only lanes

XR(1) = 68

XR(2) = 68

XR(3) = 116

XR(4) = 116

XR(5) = 116

XR(6) = 116

XR(7) = 116

XR(8) = 116

XR(9) = 68

XR(10) = 68

XR(11) = -68

XR(12) = -68

XR(13) = -116

XR(14) = -116

XR(15) = -116

XR(16) = -116

XR(17) = -116

XR(18) = -116

XR(19) = -68

XR(20) = -68

Else

'Fills for the scenario that does not inclue ETC-only lanes

XR(1) = 68

XR(2) = 68

XR(3) = 80

XR(4) = 80

XR(5) = 80

XR(6) = 80

XR(7) = 80

XR(8) = 80

XR(9) = 68

XR(10) = 68

XR(11) = -68

XR(12) = -68

XR(13) = -80

XR(14) = -80

XR(15) = -80

XR(16) = -80

XR(17) = -80

XR(18) = -80

XR(19) = -68

XR(20) = -68

End If

End If

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To (21)

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'Note: No Queue links utilized for the tollbooth scenario

If EWTollBooth = True Then

'Writes Links for E-W Scenario

'First Writes EastBound Links

If (ETCEB > 0 Or ETCWB > 0) Then

'ETC-Only lanes scenario

'EB Free-Flow Link1

CALInputLine(23) = 1

CALInputLine(24) = "'EB Free-Flow Link1'" & "," & TYP & ", " & -2500 & " , " & -30 & " , " & -900 & " , " & -30 & " , " & ATEB & " , " & EFFree & " , " & HL & " , " & 56

'EB Free-Flow Link2

CALInputLine(25) = 1

CALInputLine(26) = "'EB Free-Flow Link2'" & "," & TYP & ", " & -900 & " , " & -30 & " , " & 900 & " , " & -30 & " , " & CInt(ATEB \* (ETCEB / 100)) & " , " & EFFree & " , " & HL & " , " & 44

'EB Free-Flow Link3

CALInputLine(27) = 1

CALInputLine(28) = "'EB Free-Flow Link3'" & "," & TYP & ", " & 900 & " , " & -30 & " , " & 2500 & " , " & -30 & " , " & ATEB & " , " & EFFree & " , " & HL & " , " & 56

'EB Toll Approach Link

CALInputLine(29) = 1

CALInputLine(30) = "'EB Toll Approach Link'" & "," & TYP & ", " & -900 & " , " & -72 & " , " & 0 & " , " & -72 & " , " & CInt(ATEB \* (100 - ETCEB) / 100) & " , " & EF10mph & " , " & HL & " , " & 68

'EB Toll Departure Link

CALInputLine(31) = 1

CALInputLine(32) = "'EB Toll Departure Link'" & "," & TYP & ", " & 0 & " , " & -72 & " , " & 900 & " , " & -72 & " , " & CInt(ATEB \* (100 - ETCEB) / 100) & " , " & EFTollAcc & " , " & HL & " , " & 68

Else

'No ETC-Only lanes scenario

'EB Free-Flow Link1

CALInputLine(23) = 1

CALInputLine(24) = "'EB Free-Flow Link1'" & "," & TYP & ", " & -2500 & " , " & -30 & " , " & -900 & " , " & -30 & " , " & ATEB & " , " & EFFree & " , " & HL & " , " & 56

'EB Free-Flow Link2

CALInputLine(25) = 1

CALInputLine(26) = "'EB Free-Flow Link2'" & "," & TYP & ", " & -900 & " , " & -30 & " , " & 900 & " , " & -30 & " , " & 0 & " , " & EFFree & " , " & HL & " , " & 44

'EB Free-Flow Link3

CALInputLine(27) = 1

CALInputLine(28) = "'EB Free-Flow Link3'" & "," & TYP & ", " & 900 & " , " & -30 & " , " & 2500 & " , " & -30 & " , " & ATEB & " , " & EFFree & " , " & HL & " , " & 56

'EB Toll Approach Link

CALInputLine(29) = 1

CALInputLine(30) = "'EB Toll Approach Link'" & "," & TYP & ", " & -900 & " , " & -36 & " , " & 0 & " , " & -36 & " , " & CInt(ATEB) & " , " & EF10mph & " , " & HL & " , " & 68

'EB Toll Departure Link

CALInputLine(31) = 1

CALInputLine(32) = "'EB Toll Departure Link'" & "," & TYP & ", " & 0 & " , " & -36 & " , " & 900 & " , " & -36 & " , " & CInt(ATEB) & " , " & EFTollAcc & " , " & HL & " , " & 68

End If

'Next Writes WestBound Links

If (ETCEB > 0 Or ETCWB > 0) Then

'ETC-Only lanes scenario

'WB Free-Flow Link1

CALInputLine(33) = 1

CALInputLine(34) = "'WB Free-Flow Link1'" & "," & TYP & ", " & 2500 & " , " & 30 & " , " & 900 & " , " & 30 & " , " & ATWB & " , " & EFFree & " , " & HL & " , " & 56

'WB Free-Flow Link2

CALInputLine(35) = 1

CALInputLine(36) = "'WB Free-Flow Link2'" & "," & TYP & ", " & 900 & " , " & 30 & " , " & -900 & " , " & 30 & " , " & CInt(ATWB \* (ETCWB / 100)) & " , " & EFFree & " , " & HL & " , " & 44

'WB Free-Flow Link3

CALInputLine(37) = 1

CALInputLine(38) = "'WB Free-Flow Link3'" & "," & TYP & ", " & -900 & " , " & 30 & " , " & -2500 & " , " & 30 & " , " & ATWB & " , " & EFFree & " , " & HL & " , " & 56

'WB Toll Approach Link

CALInputLine(39) = 1

CALInputLine(40) = "'WB Toll Approach Link'" & "," & TYP & ", " & 900 & " , " & 72 & " , " & 0 & " , " & 72 & " , " & CInt(ATWB \* (100 - ETCWB) / 100) & " , " & EF10mph & " , " & HL & " , " & 68

'WB Toll Departure Link

CALInputLine(41) = 1

CALInputLine(42) = "'WB Toll Departure Link'" & "," & TYP & ", " & 0 & " , " & 72 & " , " & -900 & " , " & 72 & " , " & CInt(ATWB \* (100 - ETCWB) / 100) & " , " & EFTollAcc & " , " & HL & " , " & 68

Else

'No ETC-Only lanes scenario

'WB Free-Flow Link1

CALInputLine(33) = 1

CALInputLine(34) = "'WB Free-Flow Link1'" & "," & TYP & ", " & 2500 & " , " & 30 & " , " & 900 & " , " & 30 & " , " & ATWB & " , " & EFFree & " , " & HL & " , " & 56

'WB Free-Flow Link2

CALInputLine(35) = 1

CALInputLine(36) = "'WB Free-Flow Link2'" & "," & TYP & ", " & 900 & " , " & 30 & " , " & -900 & " , " & 30 & " , " & 0 & " , " & EFFree & " , " & HL & " , " & 44

'WB Free-Flow Link3

CALInputLine(37) = 1

CALInputLine(38) = "'WB Free-Flow Link3'" & "," & TYP & ", " & -900 & " , " & 30 & " , " & -2500 & " , " & 30 & " , " & ATWB & " , " & EFFree & " , " & HL & " , " & 56

'WB Toll Approach Link

CALInputLine(39) = 1

CALInputLine(40) = "'WB Toll Approach Link'" & "," & TYP & ", " & 900 & " , " & 36 & " , " & 0 & " , " & 36 & " , " & CInt(ATWB) & " , " & EF10mph & " , " & HL & " , " & 68

'WB Toll Departure Link

CALInputLine(41) = 1

CALInputLine(42) = "'WB Toll Departure Link'" & "," & TYP & ", " & 0 & " , " & 36 & " , " & -900 & " , " & 36 & " , " & CInt(ATWB) & " , " & EFTollAcc & " , " & HL & " , " & 68

End If

Else

'Writes Links for N-S Scenario

'First Writes NorthBound Links

If (ETCNB > 0 Or ETCSB > 0) Then

'ETC-Only lanes scenario

'NB Free-Flow Link1

CALInputLine(23) = 1

CALInputLine(24) = "'NB Free-Flow Link1'" & "," & TYP & ", " & 30 & " , " & -2500 & " , " & 30 & " , " & -900 & " , " & ATNB & " , " & EFFree & " , " & HL & " , " & 56

'NB Free-Flow Link2

CALInputLine(25) = 1

CALInputLine(26) = "'NB Free-Flow Link2'" & "," & TYP & ", " & 30 & " , " & -900 & " , " & 30 & " , " & 900 & " , " & CInt(ATNB \* (ETCNB / 100)) & " , " & EFFree & " , " & HL & " , " & 44

'NB Free-Flow Link3

CALInputLine(27) = 1

CALInputLine(28) = "'NB Free-Flow Link3'" & "," & TYP & ", " & 30 & " , " & 900 & " , " & 30 & " , " & 2500 & " , " & ATNB & " , " & EFFree & " , " & HL & " , " & 56

'NB Toll Approach Link

CALInputLine(29) = 1

CALInputLine(30) = "'NB Toll Approach Link'" & "," & TYP & ", " & 30 & " , " & -900 & " , " & 72 & " , " & 0 & " , " & CInt(ATNB \* (100 - ETCNB) / 100) & " , " & EF10mph & " , " & HL & " , " & 68

'NB Toll Departure Link

CALInputLine(31) = 1

CALInputLine(32) = "'NB Toll Departure Link'" & "," & TYP & ", " & 72 & " , " & 0 & " , " & 30 & " , " & 900 & " , " & CInt(ATNB \* (100 - ETCNB) / 100) & " , " & EFTollAcc & " , " & HL & " , " & 68

Else

'No ETC-Only lanes scenario

'NB Free-Flow Link1

CALInputLine(23) = 1

CALInputLine(24) = "'NB Free-Flow Link1'" & "," & TYP & ", " & 30 & " , " & -2500 & " , " & 30 & " , " & -900 & " , " & ATNB & " , " & EFFree & " , " & HL & " , " & 56

'NB Free-Flow Link2

CALInputLine(25) = 1

CALInputLine(26) = "'NB Free-Flow Link2'" & "," & TYP & ", " & 30 & " , " & -900 & " , " & 30 & " , " & 900 & " , " & 0 & " , " & EFFree & " , " & HL & " , " & 44

'NB Free-Flow Link3

CALInputLine(27) = 1

CALInputLine(28) = "'NB Free-Flow Link3'" & "," & TYP & ", " & 30 & " , " & 900 & " , " & 30 & " , " & 2500 & " , " & ATNB & " , " & EFFree & " , " & HL & " , " & 56

'NB Toll Approach Link

CALInputLine(29) = 1

CALInputLine(30) = "'NB Toll Approach Link'" & "," & TYP & ", " & 30 & " , " & -900 & " , " & 36 & " , " & 0 & " , " & CInt(ATNB) & " , " & EF10mph & " , " & HL & " , " & 68

'NB Toll Departure Link

CALInputLine(31) = 1

CALInputLine(32) = "'NB Toll Departure Link'" & "," & TYP & ", " & 36 & " , " & 0 & " , " & 30 & " , " & 900 & " , " & CInt(ATMax) & " , " & EFTollAcc & " , " & HL & " , " & 68

End If

'Next, Writes SouthBound Links

If (ETCNB > 0 Or ETCSB > 0) Then

'ETC-Only lanes scenario

'SB Free-Flow Link1

CALInputLine(33) = 1

CALInputLine(34) = "'SB Free-Flow Link1'" & "," & TYP & ", " & -30 & " , " & 2500 & " , " & -30 & " , " & 900 & " , " & ATSB & " , " & EFFree & " , " & HL & " , " & 56

'SB Free-Flow Link2

CALInputLine(35) = 1

CALInputLine(36) = "'SB Free-Flow Link2'" & "," & TYP & ", " & -30 & " , " & 900 & " , " & -30 & " , " & -900 & " , " & CInt(ATSB \* (ETCSB / 100)) & " , " & EFFree & " , " & HL & " , " & 44

'SB Free-Flow Link3

CALInputLine(37) = 1

CALInputLine(38) = "'SB Free-Flow Link3'" & "," & TYP & ", " & -30 & " , " & -900 & " , " & -30 & " , " & -2500 & " , " & ATSB & " , " & EFFree & " , " & HL & " , " & 56

'SB Toll Approach Link

CALInputLine(39) = 1

CALInputLine(40) = "'SB Toll Approach Link'" & "," & TYP & ", " & -30 & " , " & 900 & " , " & -72 & " , " & 0 & " , " & CInt(ATSB \* (100 - ETCSB) / 100) & " , " & EF10mph & " , " & HL & " , " & 68

'SB Toll Departure Link

CALInputLine(41) = 1

CALInputLine(42) = "'SB Toll Departure Link'" & "," & TYP & ", " & -72 & " , " & 0 & " , " & -30 & " , " & -900 & " , " & CInt(ATSB \* (100 - ETCSB) / 100) & " , " & EFTollAcc & " , " & HL & " , " & 68

Else

'No ETC-Only lanes scenario

'SB Free-Flow Link1

CALInputLine(33) = 1

CALInputLine(34) = "'SB Free-Flow Link1'" & "," & TYP & ", " & -30 & " , " & 2500 & " , " & -30 & " , " & 900 & " , " & ATSB & " , " & EFFree & " , " & HL & " , " & 56

'SB Free-Flow Link2

CALInputLine(35) = 1

CALInputLine(36) = "'SB Free-Flow Link2'" & "," & TYP & ", " & -30 & " , " & 900 & " , " & -30 & " , " & -900 & " , " & 0 & " , " & EFFree & " , " & HL & " , " & 44

'SB Free-Flow Link3

CALInputLine(37) = 1

CALInputLine(38) = "'SB Free-Flow Link3'" & "," & TYP & ", " & -30 & " , " & -900 & " , " & -30 & " , " & -2500 & " , " & ATSB & " , " & EFFree & " , " & HL & " , " & 56

'SB Toll Approach Link

CALInputLine(39) = 1

CALInputLine(40) = "'SB Toll Approach Link'" & "," & TYP & ", " & -30 & " , " & 900 & " , " & -36 & " , " & 0 & " , " & CInt(ATSB) & " , " & EF10mph & " , " & HL & " , " & 68

'SB Toll Departure Link

CALInputLine(41) = 1

CALInputLine(42) = "'SB Toll Departure Link'" & "," & TYP & ", " & -36 & " , " & 0 & " , " & -30 & " , " & -900 & " , " & CInt(ATSB) & " , " & EFTollAcc & " , " & HL & " , " & 68

End If

End If

'Last Line

CALInputLine(43) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - WEST TEE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildWestTee()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATSB > ATMax Then ATMax = ATSB

If ATNB > ATMax Then ATMax = ATNB

If ATEB > ATMax Then ATMax = ATEB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'\*\*\*Note: For the tee configurations, the traffic queue and flow volumes are calculated within the individual input line statements

'\*\*\*relative to ATMax

'WEST Tee -Specific Parameters

NR = 17

NumInputLines = 51

NL = 13

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To 18

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(19) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'EB Leg Left Q

CALInputLine(20) = 2

CALInputLine(21) = "'EB Leg Left Q'" & "," & TYP & ", " & -30 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(22) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Right Q

CALInputLine(23) = 2

CALInputLine(24) = "'EB Leg Right Q'" & "," & TYP & ", " & -30 & " , " & -24 & " , " & -3000 & " , " & -24 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(25) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Thru Q

CALInputLine(26) = 2

CALInputLine(27) = "'SB Leg Thru Q'" & "," & TYP & ", " & -18 & " , " & 36 & " , " & -18 & " , " & 3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(28) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 1) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Thru Q

CALInputLine(29) = 2

CALInputLine(30) = "'NB Leg Thru Q'" & "," & TYP & ", " & 18 & " , " & -36 & " , " & 18 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.85) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Left Q

CALInputLine(32) = 2

CALInputLine(33) = "'NB Leg Left Q'" & "," & TYP & ", " & 0 & " , " & -36 & " , " & 0 & " , " & -3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(34) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.15) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all EAST Bound Free Flow Links

'EB Leg Left

CALInputLine(35) = 1

CALInputLine(36) = "'EB Leg Left'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & -3000 & " , " & 0 & " , " & CInt(ATMax \* 0.5) & " , " & EFLt & " , " & HL & " , " & 44

'EB Leg Right

CALInputLine(37) = 1

CALInputLine(38) = "'EB Leg Right'" & "," & TYP & ", " & -18 & " , " & -24 & " , " & -3000 & " , " & -24 & " , " & CInt(ATMax \* 0.5) & " , " & EFRt & " , " & HL & " , " & 44

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Leg Thru

CALInputLine(39) = 1

CALInputLine(40) = "'SB Leg Thru '" & "," & TYP & ", " & -18 & " , " & -24 & " , " & -18 & " , " & 3000 & " , " & CInt(ATMax \* 0.75) & " , " & EFFree & " , " & HL & " , " & 44

'SB Leg Dep

CALInputLine(41) = 1

CALInputLine(42) = "'SB Leg Dep '" & "," & TYP & ", " & 18 & " , " & 0 & " , " & 18 & " , " & 3000 & " , " & CInt(ATMax \* 1.35) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all WEST Bound Free Flow Links

'WB Leg Dep

CALInputLine(43) = 1

CALInputLine(44) = "'WB Leg Dep'" & "," & TYP & ", " & 0 & " , " & 24 & " , " & -3000 & " , " & 24 & " , " & CInt(ATMax \* 0.4) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all NORTH Bound Free Flow Links

'NB Leg Thru

CALInputLine(45) = 1

CALInputLine(46) = "'NB Leg Thru'" & "," & TYP & ", " & 18 & " , " & 0 & " , " & 18 & " , " & -3000 & " , " & CInt(ATMax \* 0.85) & " , " & EFFree & " , " & HL & " , " & 44

'NB Leg Left

CALInputLine(47) = 1

CALInputLine(48) = "'NB Leg Left'" & "," & TYP & ", " & 0 & " , " & 24 & " , " & 0 & " , " & -3000 & " , " & CInt(ATMax \* 0.15) & " , " & EFLt & " , " & HL & " , " & 32

'NB Leg Dep

CALInputLine(49) = 1

CALInputLine(50) = "'NB Leg Dep'" & "," & TYP & ", " & -18 & " , " & -24 & " , " & -18 & " , " & -3000 & " , " & CInt(ATMax \* 1.25) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(51) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - SOUTH TEE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildSouthTee()

Dim i As Integer

'Calls subroutine GetEF's

GetEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATWB > ATMax Then ATMax = ATWB

If ATNB > ATMax Then ATMax = ATNB

If ATEB > ATMax Then ATMax = ATEB

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

'\*\*\*Note: For the tee configurations, the traffic queue and flow volumes are calculated within the individual input line statements

'\*\*\*relative to ATMax

'SOUTH Tee -Specific Parameters

NR = 17

NumInputLines = 51

NL = 13

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To 18

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(19) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'NB Leg Left Q

CALInputLine(20) = 2

CALInputLine(21) = "'NB Leg Left Q'" & "," & TYP & ", " & 0 & " , " & -30 & " , " & 0 & " , " & -3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(22) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Right Q

CALInputLine(23) = 2

CALInputLine(24) = "'NB Leg Right Q'" & "," & TYP & ", " & 24 & " , " & -30 & " , " & 24 & " , " & -3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(25) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.5) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Thru Q

CALInputLine(26) = 2

CALInputLine(27) = "'EB Leg Thru Q'" & "," & TYP & ", " & -36 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(28) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 1) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Thru Q

CALInputLine(29) = 2

CALInputLine(30) = "'WB Leg Thru Q'" & "," & TYP & ", " & 36 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax \* 0.85) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Left Q

CALInputLine(32) = 2

CALInputLine(33) = "'WB Leg Left Q'" & "," & TYP & ", " & 36 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(34) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax \* 0.15) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all NORTH Bound Free Flow Links

'NB Leg Left

CALInputLine(35) = 1

CALInputLine(36) = "'NB Leg Left'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & 0 & " , " & -3000 & " , " & CInt(ATMax \* 0.5) & " , " & EFLt & " , " & HL & " , " & 44

'NB Leg Right

CALInputLine(37) = 1

CALInputLine(38) = "'NB Leg Right'" & "," & TYP & ", " & 24 & " , " & -18 & " , " & 24 & " , " & -3000 & " , " & CInt(ATMax \* 0.5) & " , " & EFRt & " , " & HL & " , " & 44

'NEXT, Write all EAST Bound Free Flow Links

'EB Leg Thru

CALInputLine(39) = 1

CALInputLine(40) = "'EB Leg Thru '" & "," & TYP & ", " & 24 & " , " & -18 & " , " & -3000 & " , " & -18 & " , " & CInt(ATMax \* 0.75) & " , " & EFFree & " , " & HL & " , " & 44

'EB Leg Dep

CALInputLine(41) = 1

CALInputLine(42) = "'EB Leg Dep '" & "," & TYP & ", " & 0 & " , " & 18 & " , " & -3000 & " , " & 18 & " , " & CInt(ATMax \* 1.35) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Leg Dep

CALInputLine(43) = 1

CALInputLine(44) = "'SB Leg Dep'" & "," & TYP & ", " & -24 & " , " & 0 & " , " & -24 & " , " & -3000 & " , " & CInt(ATMax \* 0.4) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all WEST Bound Free Flow Links

'WB Leg Thru

CALInputLine(45) = 1

CALInputLine(46) = "'WB Leg Thru'" & "," & TYP & ", " & 0 & " , " & 18 & " , " & 3000 & " , " & 18 & " , " & CInt(ATMax \* 0.85) & " , " & EFFree & " , " & HL & " , " & 44

'WB Leg Left

CALInputLine(47) = 1

CALInputLine(48) = "'WB Leg Left'" & "," & TYP & ", " & -24 & " , " & 0 & " , " & 3000 & " , " & 0 & " , " & CInt(ATMax \* 0.15) & " , " & EFLt & " , " & HL & " , " & 32

'WB Leg Dep

CALInputLine(49) = 1

CALInputLine(50) = "'WB Leg Dep'" & "," & TYP & ", " & 24 & " , " & -18 & " , " & 3000 & " , " & -18 & " , " & CInt(ATMax \* 1.25) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(51) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - N-S DIAMOND\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildNSDiamond()

Dim i As Integer

'Calls subroutine GetDiamondEF's

GetDiamondEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = ATEB

If ATWB > ATMax Then ATMax = ATWB

ATLeft = CSng(0.5 \* OREB)

If CSng(0.5 \* ORWB) > ATLeft Then ATLeft = CSng(0.5 \* ORWB)

ATRight = ATLeft

FTMax = ATSB

If ATNB > FTMax Then FTMax = ATNB

FTRight = CSng(0.5 \* ORSB)

If CSng(0.5 \* ORNB) > FTRight Then FTRight = CSng(0.5 \* ORNB)

FTLeft = FTRight

FTDep = CInt(FTMax) + (CInt(ATRight) + CInt(ATLeft)) - (CInt(FTRight) + CInt(FTLeft))

ATDep = CInt(ATMax) - (CInt(ATRight) + CInt(ATLeft)) + (CInt(FTRight) + CInt(FTLeft))

'N-S Diamond particular parameters

NR = 20

NumInputLines = 111

NL = 38

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To 21

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'SB OffRamp RightQ

CALInputLine(23) = 2

CALInputLine(24) = "'SB OffRamp RightQ'" & "," & TYP & ", " & -208 & " , " & 36 & " , " & -48 & " , " & 1019 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(25) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(FTRight) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'SB OffRamp LeftQ

CALInputLine(26) = 2

CALInputLine(27) = "'SB OffRamp LeftQ'" & "," & TYP & ", " & -193 & " , " & 36 & " , " & -48 & " , " & 944 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(28) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(FTLeft) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'WB Leg Thru1Q

CALInputLine(29) = 2

CALInputLine(30) = "'WB Leg Thru1Q'" & "," & TYP & ", " & 215 & " , " & 24 & " , " & 3000 & " , " & 24 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Thru2Q

CALInputLine(32) = 2

CALInputLine(33) = "'WB Leg Thru2Q'" & "," & TYP & ", " & -190 & " , " & 24 & " , " & 193 & " , " & 24 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(34) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Left1Q

CALInputLine(35) = 2

CALInputLine(36) = "'WB Leg Left1Q'" & "," & TYP & ", " & 215 & " , " & 6 & " , " & 3000 & " , " & 6 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(37) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'WB Leg Left2Q

CALInputLine(38) = 2

CALInputLine(39) = "'WB Leg Left2Q'" & "," & TYP & ", " & -190 & " , " & 6 & " , " & 215 & " , " & 6 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(40) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB OffRamp RightQ

CALInputLine(41) = 2

CALInputLine(42) = "'NB OffRamp RightQ'" & "," & TYP & ", " & 208 & " , " & -36 & " , " & 48 & " , " & -1019 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(43) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(FTRight) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'NB OffRamp LeftQ

CALInputLine(44) = 2

CALInputLine(45) = "'NB OffRamp LeftQ'" & "," & TYP & ", " & 193 & " , " & -36 & " , " & 48 & " , " & -944 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(46) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(FTLeft) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'EB Leg Thru1Q

CALInputLine(47) = 2

CALInputLine(48) = "'EB Leg Thru1Q'" & "," & TYP & ", " & -215 & " , " & -24 & " , " & -3000 & " , " & -24 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(49) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Thru2Q

CALInputLine(50) = 2

CALInputLine(51) = "'EB Leg Thru2Q'" & "," & TYP & ", " & 190 & " , " & -24 & " , " & -193 & " , " & -24 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(52) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Left1Q

CALInputLine(53) = 2

CALInputLine(54) = "'EB Leg Left1Q'" & "," & TYP & ", " & -215 & " , " & -6 & " , " & -3000 & " , " & -6 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(55) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB Leg Left2Q

CALInputLine(56) = 2

CALInputLine(57) = "'EB Leg Left2Q'" & "," & TYP & ", " & 190 & " , " & -6 & " , " & -215 & " , " & -6 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(58) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Fwy Thru1

CALInputLine(59) = 1

CALInputLine(60) = "'SB Fwy Thru1'" & "," & TYP & ", " & -30 & " , " & 981 & " , " & -30 & " , " & 4000 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'SB Fwy Thru2

CALInputLine(61) = 1

CALInputLine(62) = "'SB Fwy Thru2'" & "," & TYP & ", " & -30 & " , " & 981 & " , " & -30 & " , " & -981 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'SB OffRamp Right

CALInputLine(63) = 1

CALInputLine(64) = "'SB OffRamp Right'" & "," & TYP & ", " & -208 & " , " & 36 & " , " & -48 & " , " & 1019 & " , " & CInt(FTRight) & " , " & EFHwy & " , " & HL & " , " & 32

'SB OffRamp Left1

CALInputLine(65) = 1

CALInputLine(66) = "'SB OffRamp Left1'" & "," & TYP & ", " & -193 & " , " & 36 & " , " & -48 & " , " & 944 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'SB OffRamp Left2

CALInputLine(67) = 1

CALInputLine(68) = "'SB OffRamp Left2'" & "," & TYP & ", " & -193 & " , " & 36 & " , " & -193 & " , " & -24 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'SB OnRamp

CALInputLine(69) = 1

CALInputLine(70) = "'SB OnRamp'" & "," & TYP & ", " & -200 & " , " & -36 & " , " & -48 & " , " & -981 & " , " & (CInt(ATLeft) + CInt(ATRight)) & " , " & EFHwyOn & " , " & HL & " , " & 44

'SB Leg Dep

CALInputLine(71) = 1

CALInputLine(72) = "'SB Leg Dep'" & "," & TYP & ", " & 30 & " , " & 981 & " , " & 30 & " , " & 4000 & " , " & CInt(FTDep) & " , " & EFHwy & " , " & HL & " , " & 56

'NEXT, Write all WEST Bound Free Flow Links

'WB Leg Thru1

CALInputLine(73) = 1

CALInputLine(74) = "'WB Leg Thru1'" & "," & TYP & ", " & 193 & " , " & 24 & " , " & 3000 & " , " & 24 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'WB Leg Thru2

CALInputLine(75) = 1

CALInputLine(76) = "'WB Leg Thru2'" & "," & TYP & ", " & -190 & " , " & 24 & " , " & 193 & " , " & 24 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'WB Leg Left1

CALInputLine(77) = 1

CALInputLine(78) = "'WB Leg Left1'" & "," & TYP & ", " & 215 & " , " & 6 & " , " & 3000 & " , " & 6 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'WB Leg Left2

CALInputLine(79) = 1

CALInputLine(80) = "'WB Leg Left2'" & "," & TYP & ", " & -190 & " , " & 6 & " , " & 215 & " , " & 6 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'WB Leg Left3

CALInputLine(81) = 1

CALInputLine(82) = "'WB Leg Left3'" & "," & TYP & ", " & -190 & " , " & 6 & " , " & -200 & " , " & -36 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'WB Leg Dep

CALInputLine(83) = 1

CALInputLine(84) = "'WB Leg Dep'" & "," & TYP & ", " & 190 & " , " & -24 & " , " & 3000 & " , " & -24 & " , " & CInt(ATDep) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all NORTH Bound Free Flow Links

'NB Fwy Thru 1

CALInputLine(85) = 1

CALInputLine(86) = "'NB Fwy Thru 1'" & "," & TYP & ", " & 30 & " , " & -981 & " , " & 30 & " , " & -4000 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'NBFwy Thru 2

CALInputLine(87) = 1

CALInputLine(88) = "'NBFwy Thru 2'" & "," & TYP & ", " & 30 & " , " & -981 & " , " & 30 & " , " & 981 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'NB OffRamp Right

CALInputLine(89) = 1

CALInputLine(90) = "'NB OffRamp Right'" & "," & TYP & ", " & 208 & " , " & -36 & " , " & 48 & " , " & -1019 & " , " & CInt(FTRight) & " , " & EFHwy & " , " & HL & " , " & 32

'NB OffRamp Left1

CALInputLine(91) = 1

CALInputLine(92) = "'NB OffRamp Left'" & "," & TYP & ", " & 193 & " , " & -36 & " , " & 48 & " , " & -944 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'NB OffRamp Left2

CALInputLine(93) = 1

CALInputLine(94) = "'NB OffRamp Left'" & "," & TYP & ", " & 193 & " , " & -36 & " , " & 193 & " , " & 24 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'NB OnRamp

CALInputLine(95) = 1

CALInputLine(96) = "'NB OnRamp'" & "," & TYP & ", " & 200 & " , " & 36 & " , " & 48 & " , " & 981 & " , " & (CInt(ATLeft) + CInt(ATRight)) & " , " & EFHwyOn & " , " & HL & " , " & 44

'NB Leg Dep

CALInputLine(97) = 1

CALInputLine(98) = "'NB Leg Dep'" & "," & TYP & ", " & -30 & " , " & -981 & " , " & -30 & " , " & -4000 & " , " & CInt(FTDep) & " , " & EFHwy & " , " & HL & " , " & 56

'NEXT, Write all EAST Bound Free Flow Links

'EB Leg Thru1

CALInputLine(99) = 1

CALInputLine(100) = "'EB Leg Thru1'" & "," & TYP & ", " & -193 & " , " & -24 & " , " & -3000 & " , " & -24 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'EB Leg Thru2

CALInputLine(101) = 1

CALInputLine(102) = "'EB Leg Thru2'" & "," & TYP & ", " & 190 & " , " & -24 & " , " & -193 & " , " & -24 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'EB Leg Left1

CALInputLine(103) = 1

CALInputLine(104) = "'EB Leg Left1'" & "," & TYP & ", " & -215 & " , " & -6 & " , " & -3000 & " , " & -6 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'EB Leg Left2

CALInputLine(105) = 1

CALInputLine(106) = "'EB Leg Left2'" & "," & TYP & ", " & 190 & " , " & -6 & " , " & -215 & " , " & -6 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'EB Leg Left3

CALInputLine(107) = 1

CALInputLine(108) = "'EB Leg Left3'" & "," & TYP & ", " & 190 & " , " & -6 & " , " & 200 & " , " & 36 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'EB Leg Dep

CALInputLine(109) = 1

CALInputLine(110) = "'EB Leg Dep'" & "," & TYP & ", " & -190 & " , " & 24 & " , " & -3000 & " , " & 24 & " , " & CInt(ATDep) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(111) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Builds input string array for CAL3QHC input files - E-W Diamond\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub BuildEWDiamond()

Dim i As Integer

'Calls subroutine GetDiamondEF's

GetDiamondEFs()

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = ATNB

If ATSB > ATMax Then ATMax = ATSB

ATLeft = CSng(0.5 \* ORNB)

If CSng(0.5 \* ORSB) > ATLeft Then ATLeft = CSng(0.5 \* ORSB)

ATRight = ATLeft

FTMax = ATEB

If ATWB > FTMax Then FTMax = ATWB

FTRight = CSng(0.5 \* OREB)

If CSng(0.5 \* ORWB) > FTRight Then FTRight = CSng(0.5 \* ORWB)

FTLeft = FTRight

FTDep = CInt(FTMax) + (CInt(ATRight) + CInt(ATLeft)) - (CInt(FTRight) + CInt(FTLeft))

ATDep = CInt(ATMax) - (CInt(ATRight) + CInt(ATLeft)) + (CInt(FTRight) + CInt(FTLeft))

'E-W Diamond particular parameters

NR = 20

NumInputLines = 111

NL = 38

'First Line

CALInputLine(1) = "'" & Job & "', " & ATIM & " , " & ZO & " , " & VS & " , " & VD & " , " & NR & " , " & SCAL & " , " & IOPT & " , " & IDEBUG

'Receptor Lines

For i = 2 To 21

CALInputLine(i) = "'Receptor " & (i - 1) & "', " & XR(i - 1) & " , " & YR(i - 1) & " , " & ZR

Next

'RunName Title Line

CALInputLine(22) = "'" & RunName & "', " & NL & " , " & NM & " , " & PRINT2 & " ," & MODE

'NEXT, Write all Queue Links

'WB OffRamp RightQ

CALInputLine(23) = 2

CALInputLine(24) = "'WB OffRamp RightQ'" & "," & TYP & ", " & 36 & " , " & 208 & " , " & 1019 & " , " & 48 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(25) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(FTRight) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'WB OffRamp LeftQ

CALInputLine(26) = 2

CALInputLine(27) = "'WB OffRamp LeftQ'" & "," & TYP & ", " & 36 & " , " & 193 & " , " & 944 & " , " & 48 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(28) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(FTLeft) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'NB Leg Thru1Q

CALInputLine(29) = 2

CALInputLine(30) = "'NB Leg Thru1Q'" & "," & TYP & ", " & 24 & " , " & -215 & " , " & 24 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(31) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Thru2Q

CALInputLine(32) = 2

CALInputLine(33) = "'NB Leg Thru2Q'" & "," & TYP & ", " & 24 & " , " & 190 & " , " & 24 & " , " & -193 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(34) = 120 & " , " & 105 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Left1Q

CALInputLine(35) = 2

CALInputLine(36) = "'NB Leg Left1Q'" & "," & TYP & ", " & 6 & " , " & -215 & " , " & 6 & " , " & -3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(37) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NB Leg Left2Q

CALInputLine(38) = 2

CALInputLine(39) = "'NB Leg Left2Q'" & "," & TYP & ", " & 6 & " , " & 190 & " , " & 6 & " , " & -215 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(40) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'EB OffRamp RightQ

CALInputLine(41) = 2

CALInputLine(42) = "'EB OffRamp RightQ'" & "," & TYP & ", " & -36 & " , " & -208 & " , " & -1019 & " , " & -48 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(43) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(FTRight) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'EB OffRamp LeftQ

CALInputLine(44) = 2

CALInputLine(45) = "'EB OffRamp LeftQ'" & "," & TYP & ", " & -36 & " , " & -193 & " , " & -944 & " , " & -48 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(46) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(FTLeft) & " , " & EFId & " , " & SFRRamps & " , " & ST & " , " & AT

'SB Leg Thru1Q

CALInputLine(47) = 2

CALInputLine(48) = "'SB Leg Thru1Q'" & "," & TYP & ", " & -24 & " , " & 215 & " , " & -24 & " , " & -3000 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(49) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Thru2Q

CALInputLine(50) = 2

CALInputLine(51) = "'SB Leg Thru2Q'" & "," & TYP & ", " & -24 & " , " & -190 & " , " & -24 & " , " & 193 & " , " & HL & " , " & 24 & " , " & 2

CALInputLine(52) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATMax) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Left1Q

CALInputLine(53) = 2

CALInputLine(54) = "'SB Leg Left1Q'" & "," & TYP & ", " & -6 & " , " & 215 & " , " & -6 & " , " & 3000 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(55) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'SB Leg Left2Q

CALInputLine(56) = 2

CALInputLine(57) = "'SB Leg Left2Q'" & "," & TYP & ", " & -6 & " , " & -190 & " , " & -6 & " , " & 215 & " , " & HL & " , " & 12 & " , " & 1

CALInputLine(58) = 120 & " , " & 75 & " , " & YFAC & " , " & CInt(ATLeft) & " , " & EFId & " , " & SFR & " , " & ST & " , " & AT

'NEXT, Write all WEST Bound Free Flow Links

'WB Fwy Thru1

CALInputLine(59) = 1

CALInputLine(60) = "'WB Fwy Thru1'" & "," & TYP & ", " & 981 & " , " & 30 & " , " & 4000 & " , " & 30 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'WB Fwy Thru2

CALInputLine(61) = 1

CALInputLine(62) = "'WB Fwy Thru2'" & "," & TYP & ", " & 981 & " , " & 30 & " , " & -981 & " , " & 30 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'WB OffRamp Right

CALInputLine(63) = 1

CALInputLine(64) = "'WB OffRamp Right'" & "," & TYP & ", " & 36 & " , " & 208 & " , " & 1019 & " , " & 48 & " , " & CInt(FTRight) & " , " & EFHwy & " , " & HL & " , " & 32

'WB OffRamp Left1

CALInputLine(65) = 1

CALInputLine(66) = "'WB OffRamp Left1'" & "," & TYP & ", " & 36 & " , " & 193 & " , " & 944 & " , " & 48 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'WB OffRamp Left2

CALInputLine(67) = 1

CALInputLine(68) = "'WB OffRamp Left2'" & "," & TYP & ", " & 36 & " , " & 193 & " , " & -24 & " , " & 193 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'WB OnRamp

CALInputLine(69) = 1

CALInputLine(70) = "'WB OnRamp'" & "," & TYP & ", " & -36 & " , " & 200 & " , " & -981 & " , " & 48 & " , " & (CInt(ATLeft) + CInt(ATRight)) & " , " & EFHwyOn & " , " & HL & " , " & 44

'WB Leg Dep

CALInputLine(71) = 1

CALInputLine(72) = "'WB Leg Dep'" & "," & TYP & ", " & 981 & " , " & -30 & " , " & 4000 & " , " & -30 & " , " & CInt(FTDep) & " , " & EFHwy & " , " & HL & " , " & 56

'NEXT, Write all MORTH Bound Free Flow Links

'NB Leg Thru1

CALInputLine(73) = 1

CALInputLine(74) = "'NB Leg Thru1'" & "," & TYP & ", " & 24 & " , " & -193 & " , " & 24 & " , " & -3000 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'NB Leg Thru2

CALInputLine(75) = 1

CALInputLine(76) = "'NB Leg Thru2'" & "," & TYP & ", " & 24 & " , " & 190 & " , " & 24 & " , " & -193 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'NB Leg Left1

CALInputLine(77) = 1

CALInputLine(78) = "'NB Leg Left1'" & "," & TYP & ", " & 6 & " , " & -215 & " , " & 6 & " , " & -3000 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'NB Leg Left2

CALInputLine(79) = 1

CALInputLine(80) = "'NB Leg Left2'" & "," & TYP & ", " & 6 & " , " & 190 & " , " & 6 & " , " & -215 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'NB Leg Left3

CALInputLine(81) = 1

CALInputLine(82) = "'NB Leg Left3'" & "," & TYP & ", " & 6 & " , " & 190 & " , " & -36 & " , " & 200 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'NB Leg Dep

CALInputLine(83) = 1

CALInputLine(84) = "'NB Leg Dep'" & "," & TYP & ", " & -24 & " , " & -190 & " , " & -24 & " , " & -3000 & " , " & CInt(ATDep) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'NEXT, Write all EAST Bound Free Flow Links

'EB Fwy Thru 1

CALInputLine(85) = 1

CALInputLine(86) = "'EB Fwy Thru 1'" & "," & TYP & ", " & -981 & " , " & -30 & " , " & -4000 & " , " & -30 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'EBFwy Thru 2

CALInputLine(87) = 1

CALInputLine(88) = "'EBFwy Thru 2'" & "," & TYP & ", " & -981 & " , " & -30 & " , " & 981 & " , " & -30 & " , " & CInt(FTMax) & " , " & EFHwy & " , " & HL & " , " & 56

'EB OffRamp Right

CALInputLine(89) = 1

CALInputLine(90) = "'EB OffRamp Right'" & "," & TYP & ", " & -36 & " , " & -208 & " , " & -1019 & " , " & -48 & " , " & CInt(FTRight) & " , " & EFHwy & " , " & HL & " , " & 32

'EB OffRamp Left1

CALInputLine(91) = 1

CALInputLine(92) = "'EB OffRamp Left1'" & "," & TYP & ", " & -36 & " , " & -193 & " , " & -944 & " , " & -48 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'EB OffRamp Left2

CALInputLine(93) = 1

CALInputLine(94) = "'EB OffRamp Left2'" & "," & TYP & ", " & -36 & " , " & -193 & " , " & 24 & " , " & -193 & " , " & CInt(FTLeft) & " , " & EFHwy & " , " & HL & " , " & 32

'EB OnRamp

CALInputLine(95) = 1

CALInputLine(96) = "'EB OnRamp'" & "," & TYP & ", " & 36 & " , " & -200 & " , " & 981 & " , " & -48 & " , " & (CInt(ATLeft) + CInt(ATRight)) & " , " & EFHwyOn & " , " & HL & " , " & 44

'EB Leg Dep

CALInputLine(97) = 1

CALInputLine(98) = "'EB Leg Dep'" & "," & TYP & ", " & -981 & " , " & 30 & " , " & -4000 & " , " & 30 & " , " & CInt(FTDep) & " , " & EFHwy & " , " & HL & " , " & 56

'NEXT, Write all SOUTH Bound Free Flow Links

'SB Leg Thru1

CALInputLine(99) = 1

CALInputLine(100) = "'SB Leg Thru1'" & "," & TYP & ", " & -24 & " , " & 193 & " , " & -24 & " , " & 3000 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'SB Leg Thru2

CALInputLine(101) = 1

CALInputLine(102) = "'SB Leg Thru2'" & "," & TYP & ", " & -24 & " , " & -190 & " , " & -24 & " , " & 193 & " , " & CInt(ATMax) & " , " & EFFree & " , " & HL & " , " & 44

'SB Leg Left1

CALInputLine(103) = 1

CALInputLine(104) = "'SB Leg Left1'" & "," & TYP & ", " & -6 & " , " & 215 & " , " & -6 & " , " & 3000 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'SB Leg Left2

CALInputLine(105) = 1

CALInputLine(106) = "'SB Leg Left2'" & "," & TYP & ", " & -6 & " , " & -190 & " , " & -6 & " , " & 215 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'SB Leg Left3

CALInputLine(107) = 1

CALInputLine(108) = "'SB Leg Left3'" & "," & TYP & ", " & -6 & " , " & -190 & " , " & 36 & " , " & -200 & " , " & CInt(ATLeft) & " , " & EFLt & " , " & HL & " , " & 32

'SB Leg Dep

CALInputLine(109) = 1

CALInputLine(110) = "'SB Leg Dep'" & "," & TYP & ", " & 24 & " , " & 190 & " , " & 24 & " , " & 3000 & " , " & CInt(ATDep) & " , " & EFFreeAcc & " , " & HL & " , " & 44

'Last Line

CALInputLine(111) = U & " , " & BRG & " , " & CLAS & " , " & MIXH & " , " & AMB & " ," & VAR & ", " & DEGR & " , " & VAL1 & " , " & VAL2

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Open and Save Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub OpenFile()

Dim i As Integer

StreamToDisplay = New StreamReader(FDOTin)

i = 1

Do Until StreamToDisplay.EndOfStream

IFR(i) = StreamToDisplay.ReadLine()

i = i + 1

Loop

If IFR(1) <> "CO Florida 2012 - User Input Summary File" Then

MsgBox("The file that you've selected is not a valid COFL input file.", 0, "CO Florida 2012")

Exit Sub

Else

ClearAll()

End If

Job = IFR(5)

FacilityName = IFR(8)

UsersName = IFR(11)

RunName = IFR(14)

ProjectYear = IFR(17)

Temperature = IFR(20)

LandUse = IFR(23)

ZO = IFR(26)

CLAS = IFR(29)

CLASLetter = IFR(30)

COBack1Hr = IFR(33)

COBack8Hr = IFR(36)

DistrictNum = IFR(39)

IntType = IFR(42)

SSB = IFR(45)

SWB = IFR(48)

SNB = IFR(51)

SEB = IFR(54)

ATSB = IFR(57)

ATWB = IFR(60)

ATNB = IFR(63)

ATEB = IFR(66)

ORSB = IFR(69)

ORWB = IFR(72)

ORNB = IFR(75)

OREB = IFR(78)

ETCSB = IFR(81)

ETCWB = IFR(84)

ETCNB = IFR(87)

ETCEB = IFR(90)

InputsCorrect(1) = IFR(93)

InputsCorrect(2) = IFR(96)

InputsCorrect(3) = IFR(99)

InputsCorrect(4) = IFR(102)

For i = 105 To 124

XR(i - 104) = IFR(i)

Next

For i = 127 To 146

YR(i - 126) = IFR(i)

Next

EWTollBooth = IFR(149)

StreamToDisplay.Close()

FillEFArray()

Select Case IntType

Case "4 X 4"

Int4X4.Init4X4()

Case "4 X 6"

Int4X6.Init4X6()

Case ("6 X 4")

Int6X4.Init6X4()

Case ("6 X 6")

Int6X6.Init6X6()

Case "East Tee"

EastTee.InitEastTee()

Case ("North Tee")

NorthTee.InitNorthTee()

Case ("Toll Booth")

TollBooth.InitTollBooth()

Case ("West Tee")

WestTee.InitWestTee()

Case ("South Tee")

SouthTee.InitSouthTee()

Case ("N-S Diamond")

NSDiamond.InitNSDiamond()

Case ("E-W Diamond")

I11.InitEWDiamond()

End Select

End Sub

Sub SaveFile()

Dim i As Integer

'Writes user inputted data to the user input file

Dim sWriter As IO.StreamWriter

sWriter = New IO.StreamWriter(FDOTin)

sWriter.WriteLine("CO Florida 2012 - User Input Summary File")

sWriter.WriteLine("")

sWriter.WriteLine("")

sWriter.WriteLine("Project Title")

sWriter.WriteLine(Job)

sWriter.WriteLine("")

sWriter.WriteLine("Facility Name")

sWriter.WriteLine(FacilityName)

sWriter.WriteLine("")

sWriter.WriteLine("User's Name")

sWriter.WriteLine(UsersName)

sWriter.WriteLine("")

sWriter.WriteLine("Run Name")

sWriter.WriteLine(RunName)

sWriter.WriteLine("")

sWriter.WriteLine("Project Year")

sWriter.WriteLine(ProjectYear)

sWriter.WriteLine("")

sWriter.WriteLine("Temperature")

sWriter.WriteLine(Temperature)

sWriter.WriteLine("")

sWriter.WriteLine("Land Use")

sWriter.WriteLine(LandUse)

sWriter.WriteLine("")

sWriter.WriteLine("Zo")

sWriter.WriteLine(ZO)

sWriter.WriteLine("")

sWriter.WriteLine("Stability Class")

sWriter.WriteLine(CLAS)

sWriter.WriteLine(CLASLetter)

sWriter.WriteLine("")

sWriter.WriteLine("1-hr CO Background Concentration")

sWriter.WriteLine(COBack1Hr)

sWriter.WriteLine("")

sWriter.WriteLine("8-hr CO Background Concentration")

sWriter.WriteLine(COBack8Hr)

sWriter.WriteLine("")

sWriter.WriteLine("FDOT District Number")

sWriter.WriteLine(DistrictNum)

sWriter.WriteLine("")

sWriter.WriteLine("Intersection Type")

sWriter.WriteLine(IntType)

sWriter.WriteLine("")

sWriter.WriteLine("Speed South Bound")

sWriter.WriteLine(SSB)

sWriter.WriteLine("")

sWriter.WriteLine("Speed West Bound")

sWriter.WriteLine(SWB)

sWriter.WriteLine("")

sWriter.WriteLine("Speed North Bound")

sWriter.WriteLine(SNB)

sWriter.WriteLine("")

sWriter.WriteLine("Speed East Bound")

sWriter.WriteLine(SEB)

sWriter.WriteLine("")

sWriter.WriteLine("Approach Traffic South Bound")

sWriter.WriteLine(ATSB)

sWriter.WriteLine("")

sWriter.WriteLine("Approach Traffic West Bound")

sWriter.WriteLine(ATWB)

sWriter.WriteLine("")

sWriter.WriteLine("Approach Traffic North Bound")

sWriter.WriteLine(ATNB)

sWriter.WriteLine("")

sWriter.WriteLine("Approach Traffic East Bound")

sWriter.WriteLine(ATEB)

sWriter.WriteLine("")

sWriter.WriteLine("On/Off Ramp Traffic South Bound")

sWriter.WriteLine(ORSB)

sWriter.WriteLine("")

sWriter.WriteLine("On/Off Ramp Traffic West Bound")

sWriter.WriteLine(ORWB)

sWriter.WriteLine("")

sWriter.WriteLine("On/Off Ramp Traffic North Bound")

sWriter.WriteLine(ORNB)

sWriter.WriteLine("")

sWriter.WriteLine("On/Off Ramp Traffic East Bound")

sWriter.WriteLine(OREB)

sWriter.WriteLine("")

sWriter.WriteLine("ETC-Only Percentage South Bound")

sWriter.WriteLine(ETCSB)

sWriter.WriteLine("")

sWriter.WriteLine("ETC-Only Percentage West Bound")

sWriter.WriteLine(ETCWB)

sWriter.WriteLine("")

sWriter.WriteLine("ETC-Only Percentage North Bound")

sWriter.WriteLine(ETCNB)

sWriter.WriteLine("")

sWriter.WriteLine("ETC-Only Percentage East Bound")

sWriter.WriteLine(ETCEB)

sWriter.WriteLine("")

sWriter.WriteLine("InputsCorrect(1)")

sWriter.WriteLine(InputsCorrect(1))

sWriter.WriteLine("")

sWriter.WriteLine("InputsCorrect(2)")

sWriter.WriteLine(InputsCorrect(2))

sWriter.WriteLine("")

sWriter.WriteLine("InputsCorrect(3)")

sWriter.WriteLine(InputsCorrect(3))

sWriter.WriteLine("")

sWriter.WriteLine("InputsCorrect(4)")

sWriter.WriteLine(InputsCorrect(4))

sWriter.WriteLine("")

sWriter.WriteLine("Receptor 'X' Coordinates")

For i = 1 To 20

sWriter.WriteLine(XR(i))

Next i

sWriter.WriteLine("")

sWriter.WriteLine("Receptor 'Y' Coordinates")

For i = 1 To 20

sWriter.WriteLine(YR(i))

Next i

sWriter.WriteLine("")

sWriter.WriteLine("EW Tollbooth")

sWriter.WriteLine(EWTollBooth)

sWriter.Flush()

sWriter.Close()

End Sub

End Module

## 

## Run

Imports System.IO

Public Class Run

Dim FinalClose As Boolean = False

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

If FinalClose = False Then

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to exit without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

End If

FinalClose = True

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TOP NAVIGATION BUTTONS\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to open a new project without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub OpenToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

Dim TestMsg As Integer

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to open an existing project without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub TitleToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to return to the Intersection Data screen without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

My.Forms.Title.Show()

Me.Visible = False

End Sub

Private Sub SaveToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub ToolStripButton1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to return to the Intersection Data screen without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

My.Forms.District.Show()

Me.Visible = False

End Sub

Private Sub IntersectionTypeToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to return to the Intersection Data screen without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

My.Forms.IntersectionType.Show()

Me.Visible = False

End Sub

Private Sub IntersectionDataToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionDataToolStripButton.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to return to the Intersection Data screen without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sends Output to Printer\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub PrintDocument1\_PrintPage\_1(ByVal sender As System.Object, ByVal e As System.Drawing.Printing.PrintPageEventArgs) Handles PrintDocument1.PrintPage

Dim i As Integer

Dim prFont As New Font("Calibri", 11, GraphicsUnit.Point)

Dim StringToPrint As String = ""

Dim y As Integer

PassFail = True

e.Graphics.DrawString("CO Florida 2012 - Results", prFont, Brushes.Black, 340, 75)

e.Graphics.DrawString(Format$(Now, "Long Date"), prFont, Brushes.Black, 335, 92)

e.Graphics.DrawString("Project Description", prFont, Brushes.Black, 365, 126)

e.Graphics.DrawString("Project Title", prFont, Brushes.Black, 168, 160)

e.Graphics.DrawString(Job, prFont, Brushes.Black, 390, 160)

e.Graphics.DrawString("Facility Name", prFont, Brushes.Black, 168, 177)

e.Graphics.DrawString(FacilityName, prFont, Brushes.Black, 390, 177)

e.Graphics.DrawString("User's Name", prFont, Brushes.Black, 168, 194)

e.Graphics.DrawString(UsersName, prFont, Brushes.Black, 390, 194)

e.Graphics.DrawString("Run Name", prFont, Brushes.Black, 168, 211)

e.Graphics.DrawString(RunName, prFont, Brushes.Black, 390, 211)

e.Graphics.DrawString("FDOT District", prFont, Brushes.Black, 168, 228)

e.Graphics.DrawString(DistrictNum, prFont, Brushes.Black, 390, 228)

e.Graphics.DrawString("Year", prFont, Brushes.Black, 168, 245)

e.Graphics.DrawString(ProjectYear, prFont, Brushes.Black, 390, 245)

e.Graphics.DrawString("Intersection Type", prFont, Brushes.Black, 168, 262)

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

e.Graphics.DrawString("E-W Freeway " & IntType, prFont, Brushes.Black, 390, 262)

Else

e.Graphics.DrawString("N-S Freeway " & IntType, prFont, Brushes.Black, 390, 262)

End If

Else

e.Graphics.DrawString(IntType, prFont, Brushes.Black, 390, 262)

End If

If (IntType = "N-S Diamond" Or IntType = "E-W Diamond") Then

e.Graphics.DrawString("Speed", prFont, Brushes.Black, 168, 279)

e.Graphics.DrawString("Arterial", prFont, Brushes.Black, 390, 279)

e.Graphics.DrawString(Speed & " mph", prFont, Brushes.Black, 450, 279)

e.Graphics.DrawString("Freeway", prFont, Brushes.Black, 535, 279)

e.Graphics.DrawString(HwySpeed & " mph", prFont, Brushes.Black, 605, 279)

e.Graphics.DrawString("Approach Traffic", prFont, Brushes.Black, 168, 296)

e.Graphics.DrawString("Arterial", prFont, Brushes.Black, 390, 296)

e.Graphics.DrawString(ATMax & " vph", prFont, Brushes.Black, 450, 296)

e.Graphics.DrawString("Freeway", prFont, Brushes.Black, 535, 296)

e.Graphics.DrawString(FTMax & " vph", prFont, Brushes.Black, 605, 296)

Else

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

e.Graphics.DrawString("Speed", prFont, Brushes.Black, 168, 279)

e.Graphics.DrawString("East Bound", prFont, Brushes.Black, 390, 279)

e.Graphics.DrawString(SEB & " mph", prFont, Brushes.Black, 475, 279)

e.Graphics.DrawString("West Bound", prFont, Brushes.Black, 560, 279)

e.Graphics.DrawString(SWB & " mph", prFont, Brushes.Black, 655, 279)

e.Graphics.DrawString("Approach Traffic", prFont, Brushes.Black, 168, 296)

e.Graphics.DrawString("EB Stopping", prFont, Brushes.Black, 390, 296)

e.Graphics.DrawString(CInt(ATEB \* ((100 - ETCEB) / 100)) & " vph", prFont, Brushes.Black, 475, 296)

e.Graphics.DrawString("WB Stopping", prFont, Brushes.Black, 560, 296)

e.Graphics.DrawString(CInt(ATWB \* ((100 - ETCWB) / 100)) & " vph", prFont, Brushes.Black, 655, 296)

e.Graphics.DrawString("EB ETC-only", prFont, Brushes.Black, 390, 313)

e.Graphics.DrawString(CInt(ATEB \* ETCEB / 100) & " vph", prFont, Brushes.Black, 475, 313)

e.Graphics.DrawString("WB ETC-only", prFont, Brushes.Black, 560, 313)

e.Graphics.DrawString(CInt(ATWB \* ETCWB / 100) & " vph", prFont, Brushes.Black, 655, 313)

Else

e.Graphics.DrawString("Speed", prFont, Brushes.Black, 168, 279)

e.Graphics.DrawString("North Bound", prFont, Brushes.Black, 390, 279)

e.Graphics.DrawString(SNB & " mph", prFont, Brushes.Black, 475, 279)

e.Graphics.DrawString("South Bound", prFont, Brushes.Black, 560, 279)

e.Graphics.DrawString(SSB & " mph", prFont, Brushes.Black, 655, 279)

e.Graphics.DrawString("Approach Traffic", prFont, Brushes.Black, 168, 296)

e.Graphics.DrawString("NB Stopping", prFont, Brushes.Black, 390, 296)

e.Graphics.DrawString(CInt(ATNB \* ((100 - ETCNB) / 100)) & " vph", prFont, Brushes.Black, 475, 296)

e.Graphics.DrawString("SB Stopping", prFont, Brushes.Black, 560, 296)

e.Graphics.DrawString(CInt(ATSB \* ((100 - ETCSB) / 100)) & " vph", prFont, Brushes.Black, 655, 296)

e.Graphics.DrawString("NB ETC-only", prFont, Brushes.Black, 390, 313)

e.Graphics.DrawString(CInt(ATNB \* ETCNB / 100) & " vph", prFont, Brushes.Black, 475, 313)

e.Graphics.DrawString("SB ETC-only", prFont, Brushes.Black, 560, 313)

e.Graphics.DrawString(CInt(ATSB \* ETCSB / 100) & " vph", prFont, Brushes.Black, 655, 313)

End If

Else

e.Graphics.DrawString("Speed", prFont, Brushes.Black, 168, 279)

e.Graphics.DrawString("Arterial", prFont, Brushes.Black, 390, 279)

e.Graphics.DrawString(Speed & " mph", prFont, Brushes.Black, 450, 279)

e.Graphics.DrawString("Approach Traffic", prFont, Brushes.Black, 168, 296)

e.Graphics.DrawString("Arterial", prFont, Brushes.Black, 390, 296)

e.Graphics.DrawString(ATMax & " vph", prFont, Brushes.Black, 450, 296)

End If

End If

e.Graphics.DrawString("Environmental Data", prFont, Brushes.Black, 365, 347)

e.Graphics.DrawString("Temperature", prFont, Brushes.Black, 168, 381)

e.Graphics.DrawString(Temperature & " °F", prFont, Brushes.Black, 390, 381)

e.Graphics.DrawString("Reid Vapor Pressure", prFont, Brushes.Black, 168, 398)

e.Graphics.DrawString(RVP & " psi", prFont, Brushes.Black, 390, 398)

e.Graphics.DrawString("Land Use", prFont, Brushes.Black, 168, 415)

e.Graphics.DrawString(LandUse, prFont, Brushes.Black, 390, 415)

e.Graphics.DrawString("Stability Class", prFont, Brushes.Black, 168, 432)

e.Graphics.DrawString(CLASLetter, prFont, Brushes.Black, 390, 432)

e.Graphics.DrawString("Surface Roughness", prFont, Brushes.Black, 168, 449)

e.Graphics.DrawString(ZO & " cm", prFont, Brushes.Black, 390, 449)

e.Graphics.DrawString("1 Hr. Background Concentration", prFont, Brushes.Black, 168, 466)

e.Graphics.DrawString(COBack1Hr & " ppm", prFont, Brushes.Black, 390, 466)

e.Graphics.DrawString("8 Hr. Background Concentration", prFont, Brushes.Black, 168, 483)

e.Graphics.DrawString(COBack8Hr & " ppm", prFont, Brushes.Black, 390, 483)

e.Graphics.DrawString("Results", prFont, Brushes.Black, 388, 517)

e.Graphics.DrawString("(ppm, including background CO)", prFont, Brushes.Black, 313, 534)

e.Graphics.DrawString("Receptor Max 1-Hr Max 8-Hr", prFont, Brushes.Black, 310, 551)

e.Graphics.DrawString("----------- ------------ --------------", prFont, Brushes.Black, 315, 568)

y = 568

For i = 1 To 9

If (Conc1(i) > 35 Or Conc8(i) > 9) Then PassFail = False

y = y + 17

e.Graphics.DrawString(" " & i, prFont, Brushes.Black, 330, y)

If Conc1(i) < 100 Then

StringToPrint = " " & Format(Conc1(i), "#0.0")

If Conc1(i) < 10 Then

StringToPrint = " " & Format(Conc1(i), "#0.0")

End If

End If

e.Graphics.DrawString(StringToPrint, prFont, Brushes.Black, 400, y)

If Conc8(i) < 100 Then

StringToPrint = " " & Format(Conc8(i), "#0.0")

If Conc8(i) < 10 Then

StringToPrint = " " & Format(Conc8(i), "#0.0")

End If

End If

e.Graphics.DrawString(StringToPrint, prFont, Brushes.Black, 485, y)

Next i

For i = 10 To 17

If (Conc1(i) > 35 Or Conc8(i) > 9) Then PassFail = False

y = y + 17

e.Graphics.DrawString(i, prFont, Brushes.Black, 330, y)

If Conc1(i) < 100 Then

StringToPrint = " " & Format(Conc1(i), "#0.0")

If Conc1(i) < 10 Then

StringToPrint = " " & Format(Conc1(i), "#0.0")

End If

End If

e.Graphics.DrawString(StringToPrint, prFont, Brushes.Black, 400, y)

If Conc8(i) < 100 Then

StringToPrint = " " & Format(Conc8(i), "#0.0")

If Conc8(i) < 10 Then

StringToPrint = " " & Format(Conc8(i), "#0.0")

End If

End If

e.Graphics.DrawString(StringToPrint, prFont, Brushes.Black, 485, y)

Next i

If (IntType <> "East Tee" And IntType <> "North Tee" And IntType <> "West Tee" And IntType <> "South Tee") Then

For i = 18 To 20

If (Conc1(i) > 35 Or Conc8(i) > 9) Then PassFail = False

y = y + 17

e.Graphics.DrawString(i, prFont, Brushes.Black, 330, y)

If Conc1(i) < 100 Then

StringToPrint = " " & Format(Conc1(i), "#0.0")

If Conc1(i) < 10 Then

StringToPrint = " " & Format(Conc1(i), "#0.0")

End If

End If

e.Graphics.DrawString(StringToPrint, prFont, Brushes.Black, 400, y)

If Conc8(i) < 100 Then

StringToPrint = " " & Format(Conc8(i), "#0.0")

If Conc8(i) < 10 Then

StringToPrint = " " & Format(Conc8(i), "#0.0")

End If

End If

e.Graphics.DrawString(StringToPrint, prFont, Brushes.Black, 485, y)

Next i

End If

'Checks for pass/fail

For i = 1 To 17

'Checks for 1 hr exceedances

If Conc1(i) >= 35 Then PassFail = False

'Checks for 8 hr exceedances

If Conc8(i) >= 9 Then PassFail = False

Next i

If (IntType = "East Tee" Or IntType = "South Tee" Or IntType = "West Tee" Or IntType = "North Tee") Then

'do nothing

Else

'Calculations on receptors 18-20

For i = 18 To 20

'Checks for 1 hr exceedances

If Conc1(i) >= 35 Then PassFail = False

'Checsk for 8 hr exceedances

If Conc8(i) >= 9 Then PassFail = False

Next i

End If

y = y + 34

'Prints pass/fail message

If PassFail = False Then

e.Graphics.DrawString("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", prFont, Brushes.Black, 290, y)

y = y + 17

e.Graphics.DrawString("\*PROJECT FAILS SCREENING MODEL\*", prFont, Brushes.Black, 290, y)

y = y + 17

e.Graphics.DrawString("\*DETAILED MODELING IS REQUIRED\*", prFont, Brushes.Black, 290, y)

y = y + 17

e.Graphics.DrawString("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", prFont, Brushes.Black, 290, y)

Else

e.Graphics.DrawString("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", prFont, Brushes.Black, 230, y)

y = y + 17

e.Graphics.DrawString("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PROJECT PASSES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", prFont, Brushes.Black, 230, y)

y = y + 17

e.Graphics.DrawString("\*NO EXCEEDANCES OF NAAQ STANDARDS ARE PREDICTED\*", prFont, Brushes.Black, 230, y)

y = y + 17

e.Graphics.DrawString("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*", prFont, Brushes.Black, 230, y)

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Saves Output to .out File\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub SaveOutputFile()

'Writes user program run Output to .out or .doc File

Dim i As Integer

Dim Tab1 As Integer

Dim Tab2 As Integer

Dim Tab3 As Integer

Dim sWriter As IO.StreamWriter

PassFail = True

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Output files (\*.out)|\*.out|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTout = SaveFileDialog1.FileName

Try

sWriter = New IO.StreamWriter(FDOTout)

Catch ex As Exception

MsgBox("The selected output file is currently open. Please close the file and try again.", 0, "CO Florida 2012")

Exit Sub

End Try

sWriter.WriteLine(Space(7) & "CO Florida 2012 - Results")

sWriter.WriteLine(Space(7) & Format$(Now, "Long Date"))

sWriter.WriteLine("")

sWriter.WriteLine(Space(10) & "Project Description")

sWriter.WriteLine(Space(0) & "Project Title" & Space(19) & Job)

sWriter.WriteLine(Space(0) & "Facility Name" & Space(19) & FacilityName)

sWriter.WriteLine(Space(0) & "User's Name" & Space(21) & UsersName)

sWriter.WriteLine(Space(0) & "Run Name" & Space(24) & RunName)

sWriter.WriteLine(Space(0) & "FDOT District" & Space(19) & DistrictNum)

sWriter.WriteLine(Space(0) & "Year" & Space(28) & ProjectYear)

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

sWriter.WriteLine(Space(0) & "Intersection Type" & Space(15) & "E-W Freeway " & IntType)

Else

sWriter.WriteLine(Space(0) & "Intersection Type" & Space(15) & "N-S Freeway " & IntType)

End If

Else

sWriter.WriteLine(Space(0) & "Intersection Type" & Space(15) & "E-W Freeway " & IntType)

End If

If (IntType = "N-S Diamond" Or IntType = "E-W Diamond") Then

'Freeway Diamond Configuration

Tab1 = 6 - Microsoft.VisualBasic.Len(ATMax)

Tab2 = 5 - Microsoft.VisualBasic.Len(FTMax)

sWriter.WriteLine("Speed" & Space(27) & "Arterial" & Space(4) & Speed & " mph" & Space(3) & "Freeway" & Space(5) & HwySpeed & " mph")

sWriter.WriteLine("Approach Traffic" & Space(16) & "Arterial" & Space(Tab1) & ATMax & " vph" & Space(3) & "Freeway " & Space(Tab2) & FTMax & " vph")

sWriter.WriteLine("")

Else

'Toll Booth Configuration

If IntType = "Toll Booth" Then

If EWTollBooth = True Then

Tab1 = 7 - Microsoft.VisualBasic.Len(CStr(CInt(ATEB \* ((100 - ETCEB) / 100))))

Tab2 = 6 - Microsoft.VisualBasic.Len(CStr(CInt(ATWB \* ((100 - ETCWB) / 100))))

sWriter.WriteLine("Speed" & Space(27) & "East Bound" & Space(6) & SEB & " mph" & Space(2) & "West Bound" & Space(5) & SWB & " mph")

sWriter.WriteLine("Approach Traffic" & Space(16) & "EB Stopping" & Space(Tab1) & CInt(ATEB \* ((100 - ETCEB) / 100)) \_

& " vph WB Stopping" & Space(Tab2) & CInt(ATWB \* ((100 - ETCWB) / 100)) & " vph")

sWriter.WriteLine(Space(32) & "EB ETC-only" & Space(Tab1) & CInt(ATEB \* (ETCEB / 100)) \_

& " vph WB ETC-only" & Space(Tab2) & CInt(ATWB \* ETCWB / 100) & " vph")

sWriter.WriteLine("")

Else

Tab1 = 7 - Microsoft.VisualBasic.Len(CStr(CInt(ATNB \* ((100 - ETCNB) / 100))))

Tab2 = 6 - Microsoft.VisualBasic.Len(CStr(CInt(ATSB \* ((100 - ETCSB) / 100))))

sWriter.WriteLine("Speed" & Space(27) & "North Bound" & Space(5) & SNB & " mph" & Space(2) & "South Bound" & Space(4) & SSB & " mph")

sWriter.WriteLine("Approach Traffic" & Space(16) & "NB Stopping" & Space(Tab1) & CInt(ATNB \* ((100 - ETCNB) / 100)) \_

& " vph SB Stopping" & Space(Tab2) & CInt(ATSB \* ((100 - ETCSB) / 100)) & " vph")

sWriter.WriteLine(Space(32) & "NB ETC-only" & Space(Tab1) & CInt(ATNB \* (ETCNB / 100)) \_

& " vph SB ETC-only" & Space(Tab2) & CInt(ATSB \* ETCSB / 100) & " vph")

sWriter.WriteLine("")

End If

Else

'Other configuratons

Tab1 = 6 - Microsoft.VisualBasic.Len(ATMax)

sWriter.WriteLine("Arterial Speed" & Space(18) & Speed & Space(4) & "mph")

sWriter.WriteLine("Max Approach Traffic" & Space(12) & ATMax & Space(Tab1) & "vph")

sWriter.WriteLine("")

End If

End If

sWriter.WriteLine(Space(10) & "Environmental Data")

sWriter.WriteLine(Space(0) & "Temperature" & Space(21) & Temperature & " F")

sWriter.WriteLine(Space(0) & "Reid Vapor Pressure" & Space(13) & RVP & " psi")

sWriter.WriteLine(Space(0) & "Land Use" & Space(24) & LandUse)

sWriter.WriteLine(Space(0) & "Stability Class" & Space(17) & CLASLetter)

sWriter.WriteLine(Space(0) & "Surface Roughness" & Space(15) & ZO & " cm")

sWriter.WriteLine(Space(0) & "1 Hr. Background Concentration" & Space(2) & COBack1Hr & " ppm")

sWriter.WriteLine(Space(0) & "8 Hr. Background Concentration" & Space(2) & COBack8Hr & " ppm")

sWriter.WriteLine("")

sWriter.WriteLine(Space(16) & "Results")

sWriter.WriteLine(Space(4) & "(ppm, including background CO)")

sWriter.WriteLine(Space(4) & "Receptor Max 1-Hr Max 8-Hr")

sWriter.WriteLine(Space(4) & "-------- -------- --------")

For i = 1 To 17

If (Conc1(i) > 35 Or Conc8(i) > 9) Then PassFail = False

Tab1 = 9 - Microsoft.VisualBasic.Len(CStr(i))

If Fix(Conc1(i)) = Conc1(i) Then

Tab2 = 10 - Microsoft.VisualBasic.Len(CStr(Conc1(i)))

Else

Tab2 = 12 - Microsoft.VisualBasic.Len(CStr(Conc1(i)))

End If

If Fix(Conc8(i)) = Conc8(i) Then

Tab3 = 9 - Microsoft.VisualBasic.Len(CStr(Conc8(i)))

Else

Tab3 = 11 - Microsoft.VisualBasic.Len(CStr(Conc8(i)))

End If

sWriter.WriteLine(Space(Tab1) & i & Space(Tab2) & Format(Conc1(i), "#0.0") & Space(Tab3) & Format(Conc8(i), "#0.0"))

Next i

If (IntType <> "East Tee" And IntType <> "North Tee" And IntType <> "West Tee" And IntType <> "South Tee") Then

For i = 18 To 20

If (Conc1(i) > 35 Or Conc8(i) > 9) Then PassFail = False

Tab1 = 9 - Microsoft.VisualBasic.Len(CStr(i))

If Fix(Conc1(i)) = Conc1(i) Then

Tab2 = 10 - Microsoft.VisualBasic.Len(CStr(Conc1(i)))

Else

Tab2 = 12 - Microsoft.VisualBasic.Len(CStr(Conc1(i)))

End If

If Fix(Conc8(i)) = Conc8(i) Then

Tab3 = 9 - Microsoft.VisualBasic.Len(CStr(Conc8(i)))

Else

Tab3 = 11 - Microsoft.VisualBasic.Len(CStr(Conc8(i)))

End If

sWriter.WriteLine(Space(Tab1) & i & Space(Tab2) & Format(Conc1(i), "#0.0") & Space(Tab3) & Format(Conc8(i), "#0.0"))

Next i

End If

'Checks for pass/fail

For i = 1 To 17

'Checks for 1 hr exceedances

If Conc1(i) >= 35 Then PassFail = False

'Checks for 8 hr exceedances

If Conc8(i) >= 9 Then PassFail = False

Next i

If (IntType = "East Tee" Or IntType = "South Tee" Or IntType = "West Tee" Or IntType = "North Tee") Then

'do nothing

Else

'Calculations on receptors 18-20

For i = 18 To 20

'Checks for 1 hr exceedances

If Conc1(i) >= 35 Then PassFail = False

'Checsk for 8 hr exceedances

If Conc8(i) >= 9 Then PassFail = False

Next i

End If

sWriter.WriteLine("")

'Prints pass/fail message

If PassFail = False Then

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*PROJECT FAILS SCREENING MODEL\*\*\*\*\*\*\*\*\*")

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*DETAILED MODELING IS REQUIRED\*\*\*\*\*\*\*\*\*")

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Else

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PROJECT PASSES\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

sWriter.WriteLine(Space(0) & "\*NO EXCEEDANCES OF NAAQ STANDARDS ARE PREDICTED\*")

sWriter.WriteLine(Space(0) & "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

End If

sWriter.Flush()

sWriter.Close()

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom Navigation Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub Button2\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

PrintDialog1.PrinterSettings = PrintDocument1.PrinterSettings

If PrintDialog1.ShowDialog() = Windows.Forms.DialogResult.OK Then

PrintDocument1.PrinterSettings = PrintDialog1.PrinterSettings

PrintDocument1.Print()

End If

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to return to the Intersection Data screen without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

SaveOutputFile()

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

TestMsg = MsgBox("Click 'Yes' to save inputs or 'No' to return to the Title screen without saving.", vbYesNo, "Save data?")

If TestMsg = vbYes Then

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End If

My.Forms.Title.Show()

Me.Visible = False

End Sub

End Class

Welcome

Imports System.IO

Imports System.Drawing

Public Class Welcome

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sub Welcome\_Load\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub Welcome\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Dim i As Integer

'Gets the primary directory

PrimaryDirectory = Directory.GetCurrentDirectory()

'States that each form has not been completed

For i = 1 To 4

InputsCorrect(i) = False

Next i

'Clears all user input variables

ClearAll()

'Diplays the picture in PictureBox1 and hides all the rest

PictureBox1.Visible = True

PictureBox2.Visible = False

PictureBox3.Visible = False

PictureBox4.Visible = False

PictureBox5.Visible = False

PictureBox6.Visible = False

PictureBox7.Visible = False

PictureBox8.Visible = False

PictureBox9.Visible = False

PictureBox10.Visible = False

PictureBox11.Visible = False

PictureBox12.Visible = False

PictureBox13.Visible = False

PictureBox14.Visible = False

PictureBox15.Visible = False

PictureBox16.Visible = False

PictureBox17.Visible = False

PictureBox18.Visible = False

PictureBox19.Visible = False

PictureBox20.Visible = False

'Inovkes the random number generator

Randomize()

RandNum = Int(Rnd() \* 20)

'Displays the picture associated with the random number generated

Select Case RandNum

Case 1

'Do nothing

Case 2

PictureBox1.Visible = False

PictureBox2.Visible = True

Case 3

PictureBox1.Visible = False

PictureBox3.Visible = True

Case 4

PictureBox1.Visible = False

PictureBox4.Visible = True

Case 5

PictureBox1.Visible = False

PictureBox5.Visible = True

Case 6

PictureBox1.Visible = False

PictureBox6.Visible = True

Case 7

PictureBox1.Visible = False

PictureBox7.Visible = True

Case 8

PictureBox1.Visible = False

PictureBox8.Visible = True

Case 9

PictureBox1.Visible = False

PictureBox9.Visible = True

Case 10

PictureBox1.Visible = False

PictureBox10.Visible = True

Case 11

PictureBox1.Visible = False

PictureBox11.Visible = True

Case 12

PictureBox1.Visible = False

PictureBox12.Visible = True

Case 13

PictureBox1.Visible = False

PictureBox13.Visible = True

Case 14

PictureBox1.Visible = False

PictureBox14.Visible = True

Case 15

PictureBox1.Visible = False

PictureBox15.Visible = True

Case 16

PictureBox1.Visible = False

PictureBox16.Visible = True

Case 17

PictureBox1.Visible = False

PictureBox17.Visible = True

Case 18

PictureBox1.Visible = False

PictureBox18.Visible = True

Case 19

PictureBox1.Visible = False

PictureBox19.Visible = True

Case 20

PictureBox1.Visible = False

PictureBox20.Visible = True

End Select

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom Navigation Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnContinue\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnContinue.Click

My.Forms.Title.Show()

Me.Visible = False

End Sub

End Class

## About

Public Class About

Private Sub btnCloseAbout\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

Me.visible = False

End Sub

Private Sub About\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

'Makes PictureBox1 visible and hides the rest

PictureBox1.Visible = True

PictureBox2.Visible = False

PictureBox3.Visible = False

PictureBox4.Visible = False

PictureBox5.Visible = False

PictureBox6.Visible = False

PictureBox7.Visible = False

PictureBox8.Visible = False

PictureBox9.Visible = False

PictureBox10.Visible = False

PictureBox11.Visible = False

PictureBox12.Visible = False

PictureBox13.Visible = False

PictureBox14.Visible = False

PictureBox15.Visible = False

PictureBox16.Visible = False

PictureBox17.Visible = False

PictureBox18.Visible = False

PictureBox19.Visible = True

'Generates a random number to be used to select image to display

Randomize()

RandNum = Int(Rnd() \* 18)

'Displays the image associated with the random number and hides the rest

Select Case RandNum

Case 1

'Do nothing

Case 2

PictureBox1.Visible = False

PictureBox2.Visible = True

Case 3

PictureBox1.Visible = False

PictureBox3.Visible = True

Case 4

PictureBox1.Visible = False

PictureBox4.Visible = True

Case 5

PictureBox1.Visible = False

PictureBox5.Visible = True

Case 6

PictureBox1.Visible = False

PictureBox6.Visible = True

Case 7

PictureBox1.Visible = False

PictureBox7.Visible = True

Case 8

PictureBox1.Visible = False

PictureBox8.Visible = True

Case 9

PictureBox1.Visible = False

PictureBox9.Visible = True

Case 10

PictureBox1.Visible = False

PictureBox10.Visible = True

Case 11

PictureBox1.Visible = False

PictureBox11.Visible = True

Case 12

PictureBox1.Visible = False

PictureBox12.Visible = True

Case 13

PictureBox1.Visible = False

PictureBox13.Visible = True

Case 14

PictureBox1.Visible = False

PictureBox14.Visible = True

Case 15

PictureBox1.Visible = False

PictureBox15.Visible = True

Case 16

PictureBox1.Visible = False

PictureBox16.Visible = True

Case 17

PictureBox1.Visible = False

PictureBox17.Visible = True

Case 18

PictureBox1.Visible = False

PictureBox18.Visible = True

End Select

End Sub

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Me.visible = False

End Sub

End Class

## Title

Imports System.IO

Public Class Title

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub Title\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

'Calls the subroutine to initialize the Title Form

InitTitle()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

InitTitle()

End Sub

Private Sub SaveToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub OpenToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub DistrictToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

'Call the routine to check form inputs

VerifyTitleForm()

'Verifies that all fields have been completed, then advances to selected form

If InputsCorrect(1) = True Then

My.Forms.District.Show()

Me.Visible = False

Else

MsgBox("All fields must be completed to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub IntersectionTypeToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

'Call the routine to check form inputs

VerifyTitleForm()

'Verifies that all fields have been completed, then advances to selected form

If InputsCorrect(1) = True Then

If InputsCorrect(2) = True Then

My.Forms.IntersectionType.Show()

Me.Visible = False

Else

MsgBox("All intermediary screens must be completed to proceed.", 0, "CO Florida 2012")

End If

Else

MsgBox("All fields must be completed to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub IntersectionDataToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionDataToolStripButton.Click

'Call the routine to check form inputs

VerifyTitleForm()

If InputsCorrect(1) = True Then

If (InputsCorrect(2) = True And InputsCorrect(3) = True) Then

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

Else

MsgBox("All intermediary screens must be completed to proceed.", 0, "CO Florida 2012")

End If

Else

MsgBox("All fields must be completed to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub RunToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyTitleForm()

If InputsCorrect(1) = True Then

If (InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4)) Then

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

Else

MsgBox("All intermediary screens must be completed to proceed.", 0, "CO Florida 2012")

End If

Else

MsgBox("All fields must be completed to proceed.", 0, "CO Florida 2012")

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sets all Buttons to Unselected\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub AllRegButtons()

Button1.Visible = True

Button2.Visible = True

Button3.Visible = True

Button4.Visible = False

Button5.Visible = False

Button6.Visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes the Title Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitTitle()

AllRegButtons()

'Loads all stored variables

TextBox1.Text = Job

TextBox2.Text = FacilityName

TextBox3.Text = UsersName

TextBox4.Text = RunName

TextBox5.Text = ProjectYear

Select Case LandUse

Case "Urban"

Button4.Visible = True

Button1.Visible = False

Case "Suburban"

Button5.Visible = True

Button2.Visible = False

Case "Rural"

Button6.Visible = True

Button3.Visible = False

End Select

'Sets picturebox1 to visible and hides the rest

PictureBox1.Visible = True

PictureBox2.Visible = False

PictureBox3.Visible = False

PictureBox4.Visible = False

PictureBox5.Visible = False

PictureBox6.Visible = False

PictureBox7.Visible = False

PictureBox8.Visible = False

PictureBox9.Visible = False

PictureBox10.Visible = False

PictureBox11.Visible = False

PictureBox12.Visible = False

PictureBox13.Visible = False

PictureBox14.Visible = False

PictureBox15.Visible = False

PictureBox16.Visible = False

PictureBox17.Visible = False

PictureBox18.Visible = False

PictureBox19.Visible = False

PictureBox20.Visible = False

'Creates a random number to choose the picturebox to display

Randomize()

RandNum = Int(Rnd() \* 20)

Select Case RandNum

Case 1

'Do nothing

Case 2

PictureBox1.Visible = False

PictureBox2.Visible = True

Case 3

PictureBox1.Visible = False

PictureBox3.Visible = True

Case 4

PictureBox1.Visible = False

PictureBox4.Visible = True

Case 5

PictureBox1.Visible = False

PictureBox5.Visible = True

Case 6

PictureBox1.Visible = False

PictureBox6.Visible = True

Case 7

PictureBox1.Visible = False

PictureBox7.Visible = True

Case 8

PictureBox1.Visible = False

PictureBox8.Visible = True

Case 9

PictureBox1.Visible = False

PictureBox9.Visible = True

Case 10

PictureBox1.Visible = False

PictureBox10.Visible = True

Case 11

PictureBox1.Visible = False

PictureBox11.Visible = True

Case 12

PictureBox1.Visible = False

PictureBox12.Visible = True

Case 13

PictureBox1.Visible = False

PictureBox13.Visible = True

Case 14

PictureBox1.Visible = False

PictureBox14.Visible = True

Case 15

PictureBox1.Visible = False

PictureBox15.Visible = True

Case 16

PictureBox1.Visible = False

PictureBox16.Visible = True

Case 17

PictureBox1.Visible = False

PictureBox17.Visible = True

Case 18

PictureBox1.Visible = False

PictureBox18.Visible = True

Case 19

PictureBox1.Visible = False

PictureBox19.Visible = True

Case 20

PictureBox1.Visible = False

PictureBox20.Visible = True

End Select

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Input Controls\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

Job = TextBox1.Text

'Verifies that Job length does not exceed 40 characters

If Len(Job) > 40 Then

MsgBox("Please note: The Project Title may not exceed 40 characters in length.", 0, "CO Florida 2012")

End If

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

FacilityName = TextBox2.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

UsersName = TextBox3.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

RunName = TextBox4.Text

'Verifies that RunName length does not exceed 40 characters

If Len(RunName) > 40 Then

MsgBox("Please note: The Run Name may not exceed 40 characters in length.", 0, "CO Florida 2012")

End If

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ProjectYear = TextBox5.Text

End Sub

'Land Use Selection Buttons

Private Sub Button1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

AllRegButtons()

'Highlights the selected button and assigns global variable values

Button4.Visible = True

Button1.Visible = False

LandUse = "Urban"

ZO = 175

CLAS = 4

CLASLetter = "D"

COBack1Hr = "5.0"

COBack8Hr = "3.0"

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

AllRegButtons()

'Highlights the selected button and assigns global variable values

Button5.Visible = True

Button2.Visible = False

LandUse = "Suburban"

ZO = 108

CLAS = 4

CLASLetter = "D"

COBack1Hr = "3.3"

COBack8Hr = "2.0"

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

AllRegButtons()

'Highlights the selected button and assigns global variable values

Button6.Visible = True

Button3.Visible = False

LandUse = "Rural"

ZO = 10

CLAS = 5

CLASLetter = "E"

COBack1Hr = "1.7"

COBack8Hr = "1.0"

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

'Call the routine to check form inputs

VerifyTitleForm()

If InputsCorrect(1) = True Then

My.Forms.District.Show()

Me.Visible = False

End If

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

'Call the routine to check form inputs

VerifyTitleForm()

If InputsCorrect(1) = True Then

My.Forms.District.Show()

Me.Visible = False

End If

End Sub

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

'Call the routine to check form inputs

VerifyTitleForm()

If InputsCorrect(1) = True Then

My.Forms.District.Show()

Me.Visible = False

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that Form Inputs are Appropriate\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyTitleForm()

InputsCorrect(1) = False

'Verifies that ProjectYear has a numeric value

Try

ProjectYear = CInt(TextBox5.Text)

Catch ex As Exception

MsgBox("The Project Year must be between 2010 and 2050. Please re-enter.", 0, "CO Florida 2012")

TextBox5.Text = ""

Exit Sub

End Try

'Verifies that all fields have been completed and that ProjectYear is between 2010 and 2050

If Job = "" Or FacilityName = "" Or UsersName = "" Or RunName = "" Or ProjectYear = "" Or LandUse = "" Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(1) = False

Else

If ProjectYear < 2010 Or ProjectYear > 2050 Then

MsgBox("The Project Year must be an integer between 2010 and 2050. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(1) = False

Else

InputsCorrect(1) = True

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom Navigation Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

VerifyTitleForm()

'Verifies that all fields have been completed, then advances to selected form

If InputsCorrect(1) = True Then

My.Forms.District.Show()

Me.Visible = False

Else

MsgBox("All fields must be completed to proceed.", 0, "CO Florida 2012")

End If

End Sub

End Class

## District

Imports System.IO

Public Class District

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub District\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InputsCorrect(2) = False

'Calls Sub to Initialize District Form

InitDistrict()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub OpenToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub SaveToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub TitleToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

My.Forms.Title.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub IntersectionTypeToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

'Verifies that a district has been selected, then advances to selected form

If InputsCorrect(2) = True Then

My.Forms.IntersectionType.Show()

FillEFArray()

Me.Visible = False

Else

MsgBox("Please select a district to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub IntersectionDataToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionDataToolStripButton.Click

If InputsCorrect(2) = True Then

If InputsCorrect(3) = True Then

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

FillEFArray()

Me.Visible = False

Else

MsgBox("All intermediary screens must be completed to proceed.", 0, "CO Florida 2012")

End If

Else

MsgBox("Please select a district to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub RunToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

If InputsCorrect(2) = True Then

If (InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4)) Then

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

FillEFArray()

Me.Visible = False

Else

MsgBox("All intermediary screens must be completed to proceed.", 0, "CO Florida 2012")

End If

Else

MsgBox("Please select a district to proceed.", 0, "CO Florida 2012")

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sets all Buttons to Unselected\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub AllRegButtons()

Button8.Visible = False

Button9.Visible = False

Button10.Visible = False

Button11.Visible = False

Button12.Visible = False

Button13.Visible = False

Button14.Visible = False

'Makes main district map visible and hides all others

PictureBox12.Visible = True

PictureBox13.Visible = False

PictureBox14.Visible = False

PictureBox15.Visible = False

PictureBox16.Visible = False

PictureBox17.Visible = False

PictureBox18.Visible = False

PictureBox19.Visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes District Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitDistrict()

AllRegButtons()

'Loads Stored District Number

Select Case DistrictNum

Case 1

Button8.Visible = True

PictureBox13.Visible = True

InputsCorrect(2) = True

Case 2

Button9.Visible = True

PictureBox14.Visible = True

InputsCorrect(2) = True

Case 3

Button10.Visible = True

PictureBox15.Visible = True

InputsCorrect(2) = True

Case 4

Button11.Visible = True

PictureBox16.Visible = True

InputsCorrect(2) = True

Case 5

Button12.Visible = True

PictureBox17.Visible = True

InputsCorrect(2) = True

Case 6

Button13.Visible = True

PictureBox18.Visible = True

InputsCorrect(2) = True

Case 7

Button14.Visible = True

PictureBox19.Visible = True

InputsCorrect(2) = True

End Select

'Makes PictureBox11 visible and hides all the others

PictureBox11.Visible = True

PictureBox2.Visible = False

PictureBox3.Visible = False

PictureBox4.Visible = False

PictureBox5.Visible = False

PictureBox6.Visible = False

PictureBox7.Visible = False

PictureBox8.Visible = False

PictureBox9.Visible = False

PictureBox10.Visible = False

'Produces a random number to use for selecting a picturebox

Randomize()

RandNum = Int(Rnd() \* 10)

'Makes the picturebox associated with the random number visible

Select Case RandNum

Case 1

'Do nothing

Case 2

PictureBox11.Visible = False

PictureBox2.Visible = True

Case 3

PictureBox11.Visible = False

PictureBox3.Visible = True

Case 4

PictureBox11.Visible = False

PictureBox4.Visible = True

Case 5

PictureBox11.Visible = False

PictureBox5.Visible = True

Case 6

PictureBox11.Visible = False

PictureBox6.Visible = True

Case 7

PictureBox11.Visible = False

PictureBox7.Visible = True

Case 8

PictureBox11.Visible = False

PictureBox8.Visible = True

Case 9

PictureBox11.Visible = False

PictureBox9.Visible = True

Case 10

PictureBox11.Visible = False

PictureBox10.Visible = True

End Select

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Input Controls\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub Button1\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button8.Visible = True

'Shows the district-specific map picture

PictureBox13.Visible = True

DistrictNum = 1

Temperature = "48.3"

InputsCorrect(2) = True

End Sub

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button9.Visible = True

'Shows the district-specific map picture

PictureBox14.Visible = True

DistrictNum = 2

Temperature = "41.1"

InputsCorrect(2) = True

End Sub

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button10.Visible = True

'Shows the district-specific map picture

PictureBox15.Visible = True

DistrictNum = 3

Temperature = "39.3"

InputsCorrect(2) = True

End Sub

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button11.Visible = True

'Shows the district-specific map picture

PictureBox16.Visible = True

DistrictNum = 4

Temperature = "53.9"

InputsCorrect(2) = True

End Sub

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button12.Visible = True

'Shows the district-specific map picture

PictureBox17.Visible = True

DistrictNum = 5

Temperature = "47.8"

InputsCorrect(2) = True

End Sub

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button13.Visible = True

'Shows the district-specific map picture

PictureBox18.Visible = True

DistrictNum = 6

Temperature = "53.9"

InputsCorrect(2) = True

End Sub

Private Sub Button7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button7.Click

'Clears previously selected district

AllRegButtons()

'Highlights the selected button

Button14.Visible = True

'Shows the district-specific map picture

PictureBox19.Visible = True

DistrictNum = 7

Temperature = "48.8"

InputsCorrect(2) = True

End Sub

Private Sub Button8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button8.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub Button9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button9.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub Button10\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button10.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub Button11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button11.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub Button12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button12.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub Button13\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button13.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

Private Sub Button14\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button14.Click

My.Forms.IntersectionType.Show()

FillEFArray()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom Navigation Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnPrev\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

'Verifies that a district has been selected, then advances to selected form

If InputsCorrect(2) = True Then

My.Forms.Title.Show()

Me.Visible = False

Else

MsgBox("Please select a district to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Verifies that a district has been selected, then advances to selected form

If InputsCorrect(2) = True Then

FillEFArray()

My.Forms.IntersectionType.Show()

Me.Visible = False

Else

MsgBox("Please select a district to proceed.", 0, "CO Florida 2012")

End If

End Sub

End Class

## Intersection Type

Imports System.IO

Public Class IntersectionType

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub IntersectionType\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitIntType()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub OpenToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

InitIntType()

End If

End Sub

Private Sub SaveToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub TitleToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub DistrictToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub IntersectionDataToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionDataToolStripButton.Click

If InputsCorrect(3) = True Then

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

Else

MsgBox("Please select an intersection type to proceed.", 0, "CO Florida 2012")

End If

End Sub

Private Sub RunToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

If InputsCorrect(3) = True Then

Select Case IntType

Case ""

My.Forms.Int4X4.Show()

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

Else

MsgBox("Please select an intersection type to proceed.", 0, "CO Florida 2012")

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Input Control Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub Select4X4()

InputsCorrect(3) = True

IntType = "4 X 4"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox12.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 40

XR(2) = 40

XR(3) = 40

XR(4) = 80

XR(5) = 180

XR(6) = 180

XR(7) = 80

XR(8) = 40

XR(9) = 40

XR(10) = 40

XR(11) = -40

XR(12) = -40

XR(13) = -40

XR(14) = -80

XR(15) = -180

XR(16) = -180

XR(17) = -80

XR(18) = -40

XR(19) = -40

XR(20) = -40

YR(1) = 180

YR(2) = 80

YR(3) = 40

YR(4) = 40

YR(5) = 40

YR(6) = -40

YR(7) = -40

YR(8) = -40

YR(9) = -80

YR(10) = -180

YR(11) = -180

YR(12) = -80

YR(13) = -40

YR(14) = -40

YR(15) = -40

YR(16) = 40

YR(17) = 40

YR(18) = 40

YR(19) = 80

YR(20) = 180

End Sub

Private Sub Select4X6()

InputsCorrect(3) = True

IntType = "4 X 6"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox13.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 58

XR(2) = 58

XR(3) = 58

XR(4) = 98

XR(5) = 198

XR(6) = 198

XR(7) = 98

XR(8) = 58

XR(9) = 58

XR(10) = 58

XR(11) = -58

XR(12) = -58

XR(13) = -58

XR(14) = -98

XR(15) = -198

XR(16) = -198

XR(17) = -98

XR(18) = -58

XR(19) = -58

XR(20) = -58

YR(1) = 180

YR(2) = 80

YR(3) = 40

YR(4) = 40

YR(5) = 40

YR(6) = -40

YR(7) = -40

YR(8) = -40

YR(9) = -80

YR(10) = -180

YR(11) = -180

YR(12) = -80

YR(13) = -40

YR(14) = -40

YR(15) = -40

YR(16) = 40

YR(17) = 40

YR(18) = 40

YR(19) = 80

YR(20) = 180

End Sub

Private Sub Select6X4()

InputsCorrect(3) = True

IntType = "6 X 4"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox14.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 40

XR(2) = 40

XR(3) = 40

XR(4) = 80

XR(5) = 180

XR(6) = 180

XR(7) = 80

XR(8) = 40

XR(9) = 40

XR(10) = 40

XR(11) = -40

XR(12) = -40

XR(13) = -40

XR(14) = -80

XR(15) = -180

XR(16) = -180

XR(17) = -80

XR(18) = -40

XR(19) = -40

XR(20) = -40

YR(1) = 198

YR(2) = 98

YR(3) = 58

YR(4) = 58

YR(5) = 58

YR(6) = -58

YR(7) = -58

YR(8) = -58

YR(9) = -98

YR(10) = -198

YR(11) = -198

YR(12) = -98

YR(13) = -58

YR(14) = -58

YR(15) = -58

YR(16) = 58

YR(17) = 58

YR(18) = 58

YR(19) = 98

YR(20) = 198

End Sub

Private Sub Select6X6()

InputsCorrect(3) = True

IntType = "6 X 6"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox15.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 58

XR(2) = 58

XR(3) = 58

XR(4) = 98

XR(5) = 198

XR(6) = 198

XR(7) = 98

XR(8) = 58

XR(9) = 58

XR(10) = 58

XR(11) = -58

XR(12) = -58

XR(13) = -58

XR(14) = -98

XR(15) = -198

XR(16) = -198

XR(17) = -98

XR(18) = -58

XR(19) = -58

XR(20) = -58

YR(1) = 198

YR(2) = 98

YR(3) = 58

YR(4) = 58

YR(5) = 58

YR(6) = -58

YR(7) = -58

YR(8) = -58

YR(9) = -98

YR(10) = -198

YR(11) = -198

YR(12) = -98

YR(13) = -58

YR(14) = -58

YR(15) = -58

YR(16) = 58

YR(17) = 58

YR(18) = 58

YR(19) = 98

YR(20) = 198

End Sub

Private Sub SelectEastTee()

InputsCorrect(3) = True

IntType = "East Tee"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox16.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 40

XR(2) = 40

XR(3) = 40

XR(4) = 80

XR(5) = 180

XR(6) = 180

XR(7) = 80

XR(8) = 40

XR(9) = 40

XR(10) = 40

XR(11) = -40

XR(12) = -40

XR(13) = -40

XR(14) = -40

XR(15) = -40

XR(16) = -40

XR(17) = -40

XR(18) = 999

XR(19) = 999

XR(20) = 999

YR(1) = 186

YR(2) = 86

YR(3) = 46

YR(4) = 46

YR(5) = 46

YR(6) = -46

YR(7) = -46

YR(8) = -46

YR(9) = -86

YR(10) = -186

YR(11) = -186

YR(12) = -86

YR(13) = -24

YR(14) = 0

YR(15) = 24

YR(16) = 86

YR(17) = 186

YR(18) = 999

YR(19) = 999

YR(20) = 999

End Sub

Private Sub SelectNorthTee()

InputsCorrect(3) = True

IntType = "North Tee"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox17.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 46

XR(2) = 46

XR(3) = 46

XR(4) = 86

XR(5) = 186

XR(6) = 186

XR(7) = 86

XR(8) = 24

XR(9) = 0

XR(10) = -24

XR(11) = -86

XR(12) = -186

XR(13) = -186

XR(14) = -86

XR(15) = -40

XR(16) = -40

XR(17) = -40

XR(18) = 999

XR(19) = 999

XR(20) = 999

YR(1) = 180

YR(2) = 80

YR(3) = 40

YR(4) = 40

YR(5) = 40

YR(6) = -40

YR(7) = -40

YR(8) = -40

YR(9) = -40

YR(10) = -40

YR(11) = -40

YR(12) = -40

YR(13) = 40

YR(14) = 40

YR(15) = 40

YR(16) = 80

YR(17) = 180

YR(18) = 999

YR(19) = 999

YR(20) = 999

End Sub

Private Sub SelectTollBooth()

InputsCorrect(3) = True

IntType = "Toll Booth"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox18.Visible = True

End Sub

Private Sub SelectWestTee()

InputsCorrect(3) = True

IntType = "West Tee"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox19.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 40

XR(2) = 40

XR(3) = 40

XR(4) = 40

XR(5) = 40

XR(6) = 40

XR(7) = 40

XR(8) = -40

XR(9) = -40

XR(10) = -40

XR(11) = -80

XR(12) = -180

XR(13) = -180

XR(14) = -80

XR(15) = -40

XR(16) = -40

XR(17) = -40

XR(18) = 999

XR(19) = 999

XR(20) = 999

YR(1) = 186

YR(2) = 86

YR(3) = 24

YR(4) = 0

YR(5) = -24

YR(6) = -86

YR(7) = -186

YR(8) = -186

YR(9) = -86

YR(10) = -46

YR(11) = -46

YR(12) = -46

YR(13) = 46

YR(14) = 46

YR(15) = 46

YR(16) = 86

YR(17) = 186

YR(18) = 999

YR(19) = 999

YR(20) = 999

End Sub

Private Sub SelectSouthTee()

InputsCorrect(3) = True

IntType = "South Tee"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox20.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = -186

XR(2) = -86

XR(3) = -24

XR(4) = 0

XR(5) = 24

XR(6) = 86

XR(7) = 186

XR(8) = 186

XR(9) = 86

XR(10) = 46

XR(11) = 46

XR(12) = 46

XR(13) = -46

XR(14) = -46

XR(15) = -46

XR(16) = -86

XR(17) = -186

XR(18) = 999

XR(19) = 999

XR(20) = 999

YR(1) = 40

YR(2) = 40

YR(3) = 40

YR(4) = 40

YR(5) = 40

YR(6) = 40

YR(7) = 40

YR(8) = -40

YR(9) = -40

YR(10) = -40

YR(11) = -80

YR(12) = -180

YR(13) = -180

YR(14) = -80

YR(15) = -40

YR(16) = -40

YR(17) = -40

YR(18) = 999

YR(19) = 999

YR(20) = 999

End Sub

Private Sub SelectNSDiamond()

InputsCorrect(3) = True

IntType = "N-S Diamond"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox21.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 46

XR(2) = 116

XR(3) = 166

XR(4) = 261

XR(5) = 361

XR(6) = 361

XR(7) = 261

XR(8) = 166

XR(9) = 116

XR(10) = 46

XR(11) = -46

XR(12) = -116

XR(13) = -166

XR(14) = -261

XR(15) = -361

XR(16) = -361

XR(17) = -261

XR(18) = -166

XR(19) = -116

XR(20) = -46

YR(1) = 1036

YR(2) = 336

YR(3) = 46

YR(4) = 46

YR(5) = 46

YR(6) = -46

YR(7) = -46

YR(8) = -46

YR(9) = -336

YR(10) = -1036

YR(11) = -1036

YR(12) = -336

YR(13) = -46

YR(14) = -46

YR(15) = -46

YR(16) = 46

YR(17) = 46

YR(18) = 46

YR(19) = 336

YR(20) = 1036

End Sub

Private Sub SelectEWDiamond()

InputsCorrect(3) = True

IntType = "E-W Diamond"

'First, clears out any big pictures

AllSmallPics()

'Next, shows the big picture for this intersection

PictureBox22.Visible = True

'Fills the receptor coordinate arrays for selected intersection type

XR(1) = 46

XR(2) = 46

XR(3) = 46

XR(4) = 336

XR(5) = 1036

XR(6) = 1036

XR(7) = 336

XR(8) = 46

XR(9) = 46

XR(10) = 46

XR(11) = -46

XR(12) = -46

XR(13) = -46

XR(14) = -336

XR(15) = -1036

XR(16) = -1036

XR(17) = -336

XR(18) = -46

XR(19) = -46

XR(20) = -46

YR(1) = 361

YR(2) = 261

YR(3) = 166

YR(4) = 116

YR(5) = 46

YR(6) = -46

YR(7) = -116

YR(8) = -166

YR(9) = -261

YR(10) = -361

YR(11) = -361

YR(12) = -261

YR(13) = -166

YR(14) = -116

YR(15) = -46

YR(16) = 46

YR(17) = 116

YR(18) = 166

YR(19) = 261

YR(20) = 361

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Sets all Picture Buttons to Unselected\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub AllSmallPics()

PictureBox12.Visible = False

PictureBox13.Visible = False

PictureBox14.Visible = False

PictureBox15.Visible = False

PictureBox16.Visible = False

PictureBox17.Visible = False

PictureBox18.Visible = False

PictureBox19.Visible = False

PictureBox20.Visible = False

PictureBox21.Visible = False

PictureBox22.Visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes Intersection Type Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitIntType()

AllSmallPics()

'Loads Stored Intersection Type

Select Case IntType

Case "4 X 4"

PictureBox12.Visible = True

Case "4 X 6"

PictureBox13.Visible = True

Case "6 X 4"

PictureBox14.Visible = True

Case "6 X 6"

PictureBox15.Visible = True

Case "East Tee"

PictureBox16.Visible = True

Case "North Tee"

PictureBox17.Visible = True

Case "Toll Booth"

PictureBox18.Visible = True

Case "West Tee"

PictureBox19.Visible = True

Case "South Tee"

PictureBox20.Visible = True

Case "N-S Diamond"

PictureBox21.Visible = True

Case "E-W Diamond"

PictureBox22.Visible = True

End Select

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom Navigation Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub btnNext\_Click\_1(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Verifies that an intersection type has been selected, then advanced to the appropriate intersection data page

If InputsCorrect(3) = True Then

Select Case IntType

Case "4 X 4"

My.Forms.Int4X4.Show()

Case "4 X 6"

My.Forms.Int4X6.Show()

Case "6 X 4"

My.Forms.Int6X4.Show()

Case "6 X 6"

My.Forms.Int6X6.Show()

Case "East Tee"

My.Forms.EastTee.Show()

Case "North Tee"

My.Forms.NorthTee.Show()

Case "Toll Booth"

My.Forms.TollBooth.Show()

Case "West Tee"

My.Forms.WestTee.Show()

Case "South Tee"

My.Forms.SouthTee.Show()

Case "N-S Diamond"

My.Forms.NSDiamond.Show()

Case "E-W Diamond"

My.Forms.I11.Show()

End Select

Me.Visible = False

Else

MsgBox("Please select a district to proceed.", 0, "CO Florida 2012")

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*User Clicks that Fire Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub PictureBox1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox1.Click

Select4X4()

End Sub

Private Sub PictureBox3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox3.Click

Select4X6()

End Sub

Private Sub PictureBox4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox4.Click

Select6X4()

End Sub

Private Sub PictureBox2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox2.Click

Select6X6()

End Sub

Private Sub PictureBox5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox5.Click

SelectEastTee()

End Sub

Private Sub PictureBox8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox8.Click

SelectNorthTee()

End Sub

Private Sub PictureBox9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox9.Click

SelectTollBooth()

End Sub

Private Sub PictureBox7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox7.Click

SelectWestTee()

End Sub

Private Sub PictureBox6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox6.Click

SelectSouthTee()

End Sub

Private Sub PictureBox10\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox10.Click

SelectNSDiamond()

End Sub

Private Sub PictureBox11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox11.Click

SelectEWDiamond()

End Sub

Private Sub Label2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label2.Click

Select4X4()

End Sub

Private Sub Label3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label3.Click

Select4X6()

End Sub

Private Sub Label4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label4.Click

Select6X4()

End Sub

Private Sub Label5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label5.Click

Select6X6()

End Sub

Private Sub Label6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label6.Click

SelectEastTee()

End Sub

Private Sub Label7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label7.Click

SelectNorthTee()

End Sub

Private Sub Label8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label8.Click

SelectTollBooth()

End Sub

Private Sub Label9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label9.Click

SelectWestTee()

End Sub

Private Sub Label10\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label10.Click

SelectSouthTee()

End Sub

Private Sub Label11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label11.Click

SelectNSDiamond()

End Sub

Private Sub Label12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Label12.Click

SelectEWDiamond()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Handle Click-Clicks on Images\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub PictureBox12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox12.Click

Select4X4()

My.Forms.Int4X4.Show()

Me.visible = False

End Sub

Private Sub PictureBox13\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox13.Click

Select4X6()

My.Forms.Int4X6.Show()

Me.visible = False

End Sub

Private Sub PictureBox14\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox14.Click

Select6X4()

My.Forms.Int6X4.Show()

Me.visible = False

End Sub

Private Sub PictureBox15\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox15.Click

Select6X6()

My.Forms.Int6X6.Show()

Me.visible = False

End Sub

Private Sub PictureBox16\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox16.Click

SelectEastTee()

My.Forms.EastTee.Show()

Me.visible = False

End Sub

Private Sub PictureBox17\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox17.Click

SelectNorthTee()

My.Forms.NorthTee.Show()

Me.visible = False

End Sub

Private Sub PictureBox18\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox18.Click

SelectTollBooth()

My.Forms.TollBooth.Show()

Me.visible = False

End Sub

Private Sub PictureBox19\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox19.Click

SelectWestTee()

My.Forms.WestTee.Show()

Me.visible = False

End Sub

Private Sub PictureBox20\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox20.Click

SelectSouthTee()

My.Forms.SouthTee.Show()

Me.visible = False

End Sub

Private Sub PictureBox21\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox21.Click

SelectNSDiamond()

My.Forms.NSDiamond.Show()

Me.visible = False

End Sub

Private Sub PictureBox22\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles PictureBox22.Click

SelectEWDiamond()

My.Forms.I11.Show()

Me.visible = False

End Sub

End Class

## Int4X4

Imports System.IO

Public Class Int4X4

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I01\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Init4X4()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub ToolStripButton1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton3.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton12.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton7.Click

QuickVerify = True

Verify4X4Form()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton8.Click

QuickVerify = True

Verify4X4Form()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton9.Click

QuickVerify = True

Verify4X4Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton11.Click

'Call the routine to check form inputs

Verify4X4Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build4X4()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes 4X4 Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Init4X4()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox5.Text = ATEB

TextBox6.Text = SEB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Verify4X4Form()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SWB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ATWB <> "" And ATNB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATWB < 0 Or ATNB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

Verify4X4Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build4X4()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

Verify4X4Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub

End Class

## Int4X6

Imports System.IO

Public Class Int4X6

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I02\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Init4X6()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub ToolStripButton1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton3.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton12\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton12.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton7.Click

QuickVerify = True

Verify4X6Form()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton8.Click

QuickVerify = True

Verify4X6Form()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton9.Click

QuickVerify = True

Verify4X6Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton11.Click

'Call the routine to check form inputs

Verify4X6Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build4X6()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes 4X6 Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Init4X6()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox5.Text = ATEB

TextBox6.Text = SEB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Verify4X6Form()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Checks other input conditions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SWB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ATWB <> "" And ATNB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATWB < 0 Or ATNB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

Verify4X6Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build4X6()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

Verify4X6Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub

End Class

## Int6X4

Imports System.IO

Public Class Int6X4

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I03\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Init6X4()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub ToolStripButton1\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton1.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton2.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton3.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton5.Click

QuickVerify = True

Verify6X4Form()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton8.Click

QuickVerify = True

Verify6X4Form()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton9.Click

QuickVerify = True

Verify6X4Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles ToolStripButton11.Click

'Call the routine to check form inputs

Verify6X4Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build6X4()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes 6X4 Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Init6X4()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox5.Text = ATEB

TextBox6.Text = SEB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Verify6X4Form()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SWB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ATWB <> "" And ATNB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATWB < 0 Or ATNB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

Verify6X4Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build6X4()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

Verify6X4Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub

End Class

## Int6X6

Imports System.IO

Public Class Int6X6

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I04\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

Init6X6()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

Verify6X6Form()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

Verify6X6Form()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

Verify6X6Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

Verify6X6Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build6X6()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initalizes 6X6 Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Init6X6()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox5.Text = ATEB

TextBox6.Text = SEB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub Verify6X6Form()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SWB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ATWB <> "" And ATNB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATWB < 0 Or ATNB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

Verify6X6Form()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

Build6X6()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

Verify6X6Form()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub

End Class

## East Tee

Imports System.IO

Public Class EastTee

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I05\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitEastTee()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifyEastTeeForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifyEastTeeForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifyEastTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyEastTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildEastTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes East Tee Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitEastTee()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyEastTeeForm()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SWB <> "" And SNB <> "" And ATSB <> "" And ATWB <> "" And ATNB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATWB < 0 Or ATNB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

VerifyEastTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildEastTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifyEastTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

End Class

## North Tee

Imports System.IO

Public Class NorthTee

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I06\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitNorthTee()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifyNorthTeeForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifyNorthTeeForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifyNorthTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyNorthTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildNorthTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes North Tee Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitNorthTee()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox5.Text = ATEB

TextBox6.Text = SEB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyNorthTeeForm()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SWB <> "" And SEB <> "" And ATSB <> "" And ATWB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATWB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

VerifyNorthTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildNorthTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifyNorthTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub  
End Class

South Tee

Imports System.IO

Public Class SouthTee

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I09\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitSouthTee()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifySouthTeeForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifySouthTeeForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifySouthTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifySouthTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildSouthTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes South Tee Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitSouthTee()

TextBox3.Text = ATWB

TextBox4.Text = SWB

TextBox5.Text = ATEB

TextBox6.Text = SEB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifySouthTeeForm()

'Extracts variables from form

ATWB = TextBox3.Text

SWB = TextBox4.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SWB <> "" And SNB <> "" And SEB <> "" And ATWB <> "" And ATNB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATWB < 0 Or ATNB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SWB < 15 Or SNB < 15 Or SEB < 15) Or (SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATWB) Then ATMax = CInt(ATWB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

VerifySouthTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildSouthTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifySouthTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to Input Boxes\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

SWB = TextBox4.Text

End Sub

Private Sub TextBox3\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox3.TextChanged

ATWB = TextBox3.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub

End Class

West Tee

Imports System.IO

Public Class WestTee

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I08\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitWestTee()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifyWestTeeForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifyWestTeeForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifyWestTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyWestTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildWestTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes West Tee Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitWestTee()

'Load Stored Variables

TextBox1.Text = SSB

TextBox2.Text = ATSB

TextBox5.Text = ATEB

TextBox6.Text = SEB

TextBox7.Text = ATNB

TextBox8.Text = SNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyWestTeeForm()

'Extracts variables from form

SSB = TextBox1.Text

ATSB = TextBox2.Text

ATEB = TextBox5.Text

SEB = TextBox6.Text

ATNB = TextBox7.Text

SNB = TextBox8.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SSB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ATNB <> "" And ATEB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ATNB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If ATMax < CInt(ATEB) Then ATMax = CInt(ATEB)

ATLeft = CInt(ATMax \* 0.15)

ATQ = CInt(ATMax \* 0.85)

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

VerifyWestTeeForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildWestTee()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifyWestTeeForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Saves changes to input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox1\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox1.TextChanged

SSB = TextBox1.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ATSB = TextBox2.Text

End Sub

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SNB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATNB = TextBox7.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

SEB = TextBox6.Text

End Sub

Private Sub TextBox5\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox5.TextChanged

ATEB = TextBox5.Text

End Sub

End Class

## E-W Diamond

Imports System.IO

Public Class I11

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I11\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitEWDiamond()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifyEWDiamondForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifyEWDiamondForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifyEWDiamondForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyEWDiamondForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildEWDiamond()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes E-W Diamond Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitEWDiamond()

'Load stored variables

TextBox8.Text = SSB

TextBox7.Text = ATSB

TextBox9.Text = ORSB

TextBox15.Text = ATWB

TextBox16.Text = SWB

TextBox4.Text = ORWB

TextBox13.Text = ATEB

TextBox14.Text = SEB

TextBox6.Text = OREB

TextBox11.Text = ATNB

TextBox12.Text = SNB

TextBox2.Text = ORNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyEWDiamondForm()

'Extracts variables from form

SSB = TextBox8.Text

ATSB = TextBox7.Text

ORSB = TextBox9.Text

ATWB = TextBox15.Text

SWB = TextBox16.Text

ORWB = TextBox4.Text

ATEB = TextBox13.Text

SEB = TextBox14.Text

OREB = TextBox6.Text

ATNB = TextBox11.Text

SNB = TextBox12.Text

ORNB = TextBox2.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ORSB = CInt(ORSB)

Catch ex As Exception

If ORSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox9.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox15.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox16.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ORWB = CInt(ORWB)

Catch ex As Exception

If ORWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox13.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox14.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

OREB = CInt(OREB)

Catch ex As Exception

If OREB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox11.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox12.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ORNB = CInt(ORNB)

Catch ex As Exception

If ORNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Checks other input conditions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Verify that all data fields have been completed

If Not (SSB <> "" And SWB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ORSB <> "" \_

And ATWB <> "" And ORWB <> "" And ATNB <> "" And ORWB <> "" \_

And ATEB <> "" And OREB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ORSB < 0 Or ORWB < 0 Or ORWB < 0 Or \_

ATNB < 0 Or ORNB < 0 Or ATEB < 0 Or OREB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that all AT's are > OR's

If (CInt(ORSB) > CInt(ATSB) Or CInt(ORWB) > CInt(ATWB) Or CInt(ORNB) > CInt(ATNB) Or CInt(OREB) > CInt(ATEB)) Then

If QuickVerify = False Then

MsgBox("Ramp volumes may not exceed approach volumes. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

FTMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If FTMax < CInt(ATEB) Then ATMax = CInt(ATEB)

If FTMax < CInt(ATWB) Then ATMax = CInt(ATWB)

End If

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifyEWDiamondForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub btnRun\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun.Click

'Call the routine to check form inputs

VerifyEWDiamondForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildEWDiamond()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SSB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATSB = TextBox7.Text

End Sub

Private Sub TextBox9\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox9.TextChanged

ORSB = TextBox9.Text

End Sub

Private Sub TextBox16\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox16.TextChanged

SWB = TextBox16.Text

End Sub

Private Sub TextBox15\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox15.TextChanged

ATWB = TextBox15.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

ORWB = TextBox4.Text

End Sub

Private Sub TextBox12\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox12.TextChanged

SNB = TextBox12.Text

End Sub

Private Sub TextBox11\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox11.TextChanged

ATNB = TextBox11.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ORNB = TextBox2.Text

End Sub

Private Sub TextBox14\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox14.TextChanged

SEB = TextBox14.Text

End Sub

Private Sub TextBox13\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox13.TextChanged

ATEB = TextBox13.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

OREB = TextBox6.Text

End Sub

End Class

## N-S Diamond

Imports System.IO

Public Class NSDiamond

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub I10\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitNSDiamond()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifyNSDiamondForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifyNSDiamondForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifyNSDiamondForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyNSDiamondForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildNSDiamond()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes N-S Diamond\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitNSDiamond()

'Load stored variables

TextBox8.Text = SSB

TextBox7.Text = ATSB

TextBox9.Text = ORSB

TextBox15.Text = ATWB

TextBox16.Text = SWB

TextBox4.Text = ORWB

TextBox13.Text = ATEB

TextBox14.Text = SEB

TextBox6.Text = OREB

TextBox11.Text = ATNB

TextBox12.Text = SNB

TextBox2.Text = ORNB

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyNSDiamondForm()

'Extracts variables from form

SSB = TextBox8.Text

ATSB = TextBox7.Text

ORSB = TextBox9.Text

ATWB = TextBox15.Text

SWB = TextBox16.Text

ORWB = TextBox4.Text

ATEB = TextBox13.Text

SEB = TextBox14.Text

OREB = TextBox6.Text

ATNB = TextBox11.Text

SNB = TextBox12.Text

ORNB = TextBox2.Text

'Verifies that all fields have numeric inputs

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ORSB = CInt(ORSB)

Catch ex As Exception

If ORSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox9.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox15.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox16.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ORWB = CInt(ORWB)

Catch ex As Exception

If ORWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox13.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox14.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

OREB = CInt(OREB)

Catch ex As Exception

If OREB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox6.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox11.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox12.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ORNB = CInt(ORNB)

Catch ex As Exception

If ORNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Verify that all data fields have been completed

If Not (SSB <> "" And SWB <> "" And SNB <> "" And SEB <> "" And ATSB <> "" And ORSB <> "" \_

And ATWB <> "" And ORWB <> "" And ATNB <> "" And ORWB <> "" \_

And ATEB <> "" And OREB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATSB < 0 Or ORSB < 0 Or ORWB < 0 Or ORWB < 0 Or \_

ATNB < 0 Or ORNB < 0 Or ATEB < 0 Or OREB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SSB < 15 Or SWB < 15 Or SNB < 15 Or SEB < 15) Or (SSB > 65 Or SWB > 65 Or SNB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATSB > 100000 Or ATWB > 100000 Or ATNB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that all AT's are > OR's

If (CInt(ORSB) > CInt(ATSB) Or CInt(ORWB) > CInt(ATWB) Or CInt(ORNB) > CInt(ATNB) Or CInt(OREB) > CInt(ATEB)) Then

If QuickVerify = False Then

MsgBox("Ramp volumes may not exceed approach volumes. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Determines traffic volume to be used for all directions in the CAL3QHC input file

ATMax = 0

FTMax = 0

If ATMax < CInt(ATSB) Then ATMax = CInt(ATSB)

If ATMax < CInt(ATNB) Then ATMax = CInt(ATNB)

If FTMax < CInt(ATEB) Then ATMax = CInt(ATEB)

If FTMax < CInt(ATWB) Then ATMax = CInt(ATWB)

End If

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifyNSDiamondForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub btnRun\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnRun.Click

'Call the routine to check form inputs

VerifyNSDiamondForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildNSDiamond()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Changes to variables in input fields\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TextBox8\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox8.TextChanged

SSB = TextBox8.Text

End Sub

Private Sub TextBox7\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox7.TextChanged

ATSB = TextBox7.Text

End Sub

Private Sub TextBox9\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox9.TextChanged

ORSB = TextBox9.Text

End Sub

Private Sub TextBox16\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox16.TextChanged

SWB = TextBox16.Text

End Sub

Private Sub TextBox15\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox15.TextChanged

ATWB = TextBox15.Text

End Sub

Private Sub TextBox4\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox4.TextChanged

ORWB = TextBox4.Text

End Sub

Private Sub TextBox12\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox12.TextChanged

SNB = TextBox12.Text

End Sub

Private Sub TextBox11\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox11.TextChanged

ATNB = TextBox11.Text

End Sub

Private Sub TextBox2\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox2.TextChanged

ORNB = TextBox2.Text

End Sub

Private Sub TextBox14\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox14.TextChanged

SEB = TextBox14.Text

End Sub

Private Sub TextBox13\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox13.TextChanged

ATEB = TextBox13.Text

End Sub

Private Sub TextBox6\_TextChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TextBox6.TextChanged

OREB = TextBox6.Text

End Sub

End Class

## Tollbooth

Imports System.IO

Public Class TollBooth

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Form Load and Close Subroutines\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub TollBooth\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

InitTollBooth()

End Sub

Private Sub Form\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

Application.Exit()

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Top Navigation Menus and Buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub NewToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles NewToolStripButton.Click

ClearAll()

District.InitDistrict()

IntersectionType.InitIntType()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

End Sub

Private Sub ToolStripButton2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles OpenToolStripButton.Click

OpenFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

OpenFileDialog1.Filter = "Input files (\*in)|\*.in|All files (\*.\*)|\*.\*"

OpenFileDialog1.FileName = ""

If OpenFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = OpenFileDialog1.FileName

OpenFile()

My.Forms.Title.Show()

Me.Visible = False

Title.InitTitle()

District.InitDistrict()

IntersectionType.InitIntType()

End If

End Sub

Private Sub ToolStripButton3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles SaveToolStripButton.Click

'Writes user program run inputs to text file

SaveFileDialog1.InitialDirectory = PrimaryDirectory & "\UsrFiles\"

SaveFileDialog1.Filter = "Input files (\*.in)|\*.in|All files (\*.\*)|\*.\*"

SaveFileDialog1.FileName = ""

If SaveFileDialog1.ShowDialog() = DialogResult.OK Then

FDOTin = SaveFileDialog1.FileName

SaveFile()

End If

End Sub

Private Sub AboutToolStripButton\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles AboutToolStripButton.Click

My.Forms.About.Show()

End Sub

Private Sub ToolStripButton7\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles TitleToolStripButton.Click

QuickVerify = True

VerifyTollboothForm()

QuickVerify = False

My.Forms.Title.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton8\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles DistrictToolStripButton.Click

QuickVerify = True

VerifyTollboothForm()

QuickVerify = False

My.Forms.District.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton9\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles IntersectionTypeToolStripButton.Click

QuickVerify = True

VerifyTollboothForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub ToolStripButton11\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles RunToolStripButton.Click

'Call the routine to check form inputs

VerifyTollboothForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildTollBooth()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Initializes Toll Booth Form\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub InitTollBooth()

'Load Stored Variables

TextBox6.Text = SSB

TextBox7.Text = ATSB

TextBox8.Text = ETCSB

TextBox5.Text = SWB

TextBox4.Text = ATWB

TextBox3.Text = ETCWB

TextBox9.Text = SNB

TextBox10.Text = ATNB

TextBox11.Text = ETCNB

TextBox14.Text = SEB

TextBox1.Text = ATEB

TextBox2.Text = ETCEB

If EWTollBooth = False Then

PictureBox2.Visible = False

PictureBox3.Visible = False

PictureBox4.Visible = False

Label2.Visible = False

Label3.Visible = False

Label4.Visible = False

Label5.Visible = False

Label6.Visible = False

Label10.Visible = False

Label11.Visible = False

Label19.Visible = False

Label20.Visible = False

TextBox1.Visible = False

TextBox2.Visible = False

TextBox3.Visible = False

TextBox4.Visible = False

TextBox5.Visible = False

TextBox14.Visible = False

PictureBox1.Visible = True

PictureBox5.Visible = True

PictureBox6.Visible = True

Label7.Visible = True

Label8.Visible = True

Label9.Visible = True

Label12.Visible = True

Label13.Visible = True

Label14.Visible = True

Label15.Visible = True

Label16.Visible = True

Label17.Visible = True

TextBox6.Visible = True

TextBox7.Visible = True

TextBox8.Visible = True

TextBox9.Visible = True

TextBox10.Visible = True

TextBox11.Visible = True

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Verifies that proper data has been inputted\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Sub VerifyTollboothForm()

'Extracts variables from form

SSB = TextBox6.Text

ATSB = TextBox7.Text

ETCSB = TextBox8.Text

SWB = TextBox5.Text

ATWB = TextBox4.Text

ETCWB = TextBox3.Text

SNB = TextBox9.Text

ATNB = TextBox10.Text

ETCNB = TextBox11.Text

SEB = TextBox14.Text

ATEB = TextBox1.Text

ETCEB = TextBox2.Text

'Verifies that all fields have numeric inputs

'Checks for EW Orientation:

If PictureBox2.Visible = True Then

Try

SWB = CInt(SWB)

Catch ex As Exception

If SWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox5.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATWB = CInt(ATWB)

Catch ex As Exception

If ATWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox4.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ETCWB = CInt(ETCWB)

Catch ex As Exception

If ETCWB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox3.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SEB = CInt(SEB)

Catch ex As Exception

If SEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox14.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATEB = CInt(ATEB)

Catch ex As Exception

If ATEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox1.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ETCEB = CInt(ETCEB)

Catch ex As Exception

If ETCEB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox2.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions

'Check that at least one speed and one approach traffic volume have been inputted

If Not (SWB <> "" And SEB <> "" And ATWB <> "" And ATEB <> "" And ETCEB <> "" And ETCWB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATWB < 0 Or ATEB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SWB < 15 Or SEB < 15) Or (SWB > 65 Or SEB > 65) Then

If QuickVerify = False Then

MsgBox("Freeway cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATWB > 100000 Or ATEB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that ETC's are between 0 and 100

If (CInt(ETCWB) < 0 Or CInt(ETCWB) > 100 Or CInt(ETCEB) < 0 Or CInt(ETCEB > 100)) Then

If QuickVerify = False Then

MsgBox("The percentage of vehicles utilizing the ETC-Only lanes must be between 0 and 100. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Sets speeds on unused directions = 55

SSB = 55

SNB = 55

End If

End If

End If

End If

End If

Else

'Checks for NS Orientation

Try

SSB = CInt(SSB)

Catch ex As Exception

If SSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATSB = CInt(ATSB)

Catch ex As Exception

If ATSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox7.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ETCSB = CInt(ETCSB)

Catch ex As Exception

If ETCSB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox8.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

SNB = CInt(SNB)

Catch ex As Exception

If SNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox9.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ATNB = CInt(ATNB)

Catch ex As Exception

If ATNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox10.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

Try

ETCNB = CInt(ETCNB)

Catch ex As Exception

If ETCNB <> "" Then

If QuickVerify = False Then

MsgBox("All input fields must be positive integers. Please correct inputs", 0, "CO Florida 2012")

InputsCorrect(4) = False

TextBox11.Text = ""

Exit Sub

Else

InputsCorrect(4) = False

End If

End If

End Try

'Checks other input conditions\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'Check that at least one speed and one approach traffic volume have been inputted

If Not (ETCNB <> "" And ETCSB <> "" And SNB <> "" And SSB <> "" And ATNB <> "" And ATSB <> "") Then

If QuickVerify = False Then

MsgBox("All fields must be completed in order to proceed.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'verifies that traffic volumes are positive numbers

If ATNB < 0 Or ATSB < 0 Then

If QuickVerify = False Then

MsgBox("Traffic volumes must be positive numbers. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that cruise speeds are between 15 and 65 mph

If (SNB < 15 Or SSB < 15) Or (SNB > 65 Or SSB > 65) Then

If QuickVerify = False Then

MsgBox("Freeway cruise speeds must be between 15 and 65 mph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies tyhat All AT's are <= 100,000

If ATNB > 100000 Or ATSB > 100000 Then

If QuickVerify = False Then

MsgBox("Traffic volumes may not exceed 100,000 vph. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Verifies that ETC's are between 0 and 100

If (CInt(ETCNB) < 0 Or CInt(ETCNB) > 100 Or CInt(ETCSB) < 0 Or CInt(ETCSB > 100)) Then

If QuickVerify = False Then

MsgBox("The percentage of vehicles utilizing the ETC-Only lanes must be between 0 and 100. Please re-enter.", 0, "CO Florida 2012")

InputsCorrect(4) = False

Else

InputsCorrect(4) = False

End If

Else

'Assigns "true" to InputsCorrect, if all tests were passed

InputsCorrect(4) = True

'Sets speeds on unused directions = 55

SEB = 55

SWB = 55

End If

End If

End If

End If

End If

End If

End Sub

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bottom navigation buttons\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Private Sub btnNext\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnNext.Click

'Call the routine to check form inputs

VerifyTollboothForm()

'Verifies that all input forms have been completed before making a run

If (InputsCorrect(1) = True And InputsCorrect(2) = True And InputsCorrect(3) = True And InputsCorrect(4) = True) Then

BuildTollBooth()

MakeARun()

Me.visible = False

Else

If InputsCorrect(4) = True Then

MsgBox("A run cannot be made until all input forms have been completed.", 0, "CO Florida 2012")

End If

End If

End Sub

Private Sub btnPrevious\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnPrevious.Click

QuickVerify = True

VerifyTollboothForm()

QuickVerify = False

My.Forms.IntersectionType.Show()

Me.visible = False

End Sub

Private Sub PictureBox2\_Click(sender As System.Object, e As System.EventArgs) Handles PictureBox2.Click

'Switches to N-S Orientation

EWTollBooth = False

PictureBox2.Visible = False

PictureBox3.Visible = False

PictureBox4.Visible = False

Label2.Visible = False

Label3.Visible = False

Label4.Visible = False

Label5.Visible = False

Label6.Visible = False

Label10.Visible = False

Label11.Visible = False

Label19.Visible = False

Label20.Visible = False

TextBox1.Visible = False

TextBox2.Visible = False

TextBox3.Visible = False

TextBox4.Visible = False

TextBox5.Visible = False

TextBox14.Visible = False

PictureBox1.Visible = True

PictureBox5.Visible = True

PictureBox6.Visible = True

Label7.Visible = True

Label8.Visible = True

Label9.Visible = True

Label12.Visible = True

Label13.Visible = True

Label14.Visible = True

Label15.Visible = True

Label16.Visible = True

Label17.Visible = True

TextBox6.Visible = True

TextBox7.Visible = True

TextBox8.Visible = True

TextBox9.Visible = True

TextBox10.Visible = True

TextBox11.Visible = True

End Sub

Private Sub PictureBox1\_Click(sender As System.Object, e As System.EventArgs) Handles PictureBox1.Click

'Switches to E-W Orientation

EWTollBooth = True

PictureBox2.Visible = True

PictureBox3.Visible = True

PictureBox4.Visible = True

Label2.Visible = True

Label3.Visible = True

Label4.Visible = True

Label5.Visible = True

Label6.Visible = True

Label10.Visible = True

Label11.Visible = True

Label19.Visible = True

Label20.Visible = True

TextBox1.Visible = True

TextBox2.Visible = True

TextBox3.Visible = True

TextBox4.Visible = True

TextBox5.Visible = True

TextBox14.Visible = True

PictureBox1.Visible = False

PictureBox5.Visible = False

PictureBox6.Visible = False

Label7.Visible = False

Label8.Visible = False

Label9.Visible = False

Label12.Visible = False

Label13.Visible = False

Label14.Visible = False

Label15.Visible = False

Label16.Visible = False

Label17.Visible = False

TextBox6.Visible = False

TextBox7.Visible = False

TextBox8.Visible = False

TextBox9.Visible = False

TextBox10.Visible = False

TextBox11.Visible = False

End Sub

End Class