|  |  |  |  |
| --- | --- | --- | --- |
|  | 5/24/2012 |  | |
|  | |  | |
| *Denver Regional Council of Governments* | | | |
|  |  | |  |
|  |  | |  |

Focus Activity-Based Travel Model User’s Guide 

Contents

[Focus Model Run Checklist 3](#_Toc280705479)

[Minimum Hardware Requirements 3](#_Toc280705480)

[Software Requirements 4](#_Toc280705481)

[TransCAD Setup and Input Creation 5](#_Toc280705482)

[TransCAD Installation 5](#_Toc280705483)

[TransCAD input files 6](#_Toc280705484)

[1. Combined Peak & Off-Peak Highway Geographic File 7](#_Toc280705485)

[2. Combined Peak & Off-Peak Transit Route System 8](#_Toc280705486)

[3. Zone Attribute Table 9](#_Toc280705487)

[4. Modeling Forecast Year 10](#_Toc280705488)

[Starting Speeds 10](#_Toc280705489)

[SQL Server Database Input Creation Cheat Sheet 11](#_Toc280705490)

[Running the Population Synthesizer 14](#_Toc280705491)

[SQL Server Input File Creation 16](#_Toc280705492)

[Creating New Databases 16](#_Toc280705493)

[Filling IRM\_Data Database 19](#_Toc280705494)

[Input.csv files and fields 19](#_Toc280705495)

[1. Zones 19](#_Toc280705496)

[2. Jobs 22](#_Toc280705497)

[3. HousingUnits 22](#_Toc280705498)

[4. Schools 22](#_Toc280705499)

[5. Households 23](#_Toc280705500)

[6. Persons 23](#_Toc280705501)

[Running the Input Initialization Stored Procedures 23](#_Toc280705502)

[Starting a Model Run 25](#_Toc280705503)

[Current Unfortunate Hard-Codes 26](#_Toc280705504)

[Model Steps 28](#_Toc280705505)

[Tricky Things 28](#_Toc280705506)

[Model Results Summarization 28](#_Toc280705507)

[Database Results 28](#_Toc280705508)

[TransCAD Results 29](#_Toc280705509)

# Focus Model Run Checklist

|  |  |  |
| --- | --- | --- |
| **Done** | **Description of This Step** | **Notes** |
|  | **STAGE 1: GATHER DATA** |  |
|  | * Clear off 50 GB of space to hold model run |  |
|  | * Get the Latest Land Use Run from the LU modelers |  |
|  | * Compile all TransCAD inputs- highway and transit networks, zone.bin |  |
|  | * Run GISDK Parking Cost and Area Type Model if needed |  |
|  | * Create New Points: Household, Employment, School |  |
|  | * Create New Zone Table, intersection density-Aggregate Point Level Data to Zone Level csv file |  |
|  | * Create New Data Database to Hold data:   add irmdbuser as a user. |  |
|  | * Run PopSyn |  |
|  | * Import Data Into Database; Ensure That All the procedures point to your csv files |  |
|  | * Run Procedures to Create All Derived Variables on Households, Zones, and Points Tables |  |
|  | * Put TransCAD inputs into run directory where you’re running |  |
|  | * Back-up Your database |  |
|  | **STAGE 2: MODEL COMPONENT RUNS** |  |
|  | * Make a New Scenario. Set * a) The List of Components To Be Run * b) The Directory Location of all TransCAD input files * c) The Name of the Data Database and Server * d) The names of the Four TransCAD Input Files |  |
|  | * Run Model Components |  |

# 

# Installing the Focus Model

The basic steps to install the Focus model are as follows:

1. Install the necessary hardware and software.

2. Install and configure Git and GitHub code repositories.

3.Clone the C# and GISDK code from GitHub. Ask the DRCOG IT Administrator to add you to the DRCOG organization on GitHub.

4. Compile the GISDK code.

5. Configure the C# code for your system and compile it in Visual Studio.

6. Restore and configure the SQL Server databases: IRM and IRM\_Data\_*YourModelRun.*

# Minimum Hardware Requirements

The following system requirements are recommended at minimum:

* Windows 64-Bit Operating System
* 6 GB RAM
* 2 dual core processors= 4 total processors
* 200 GB Hard Drive Space

The model can be conceivably be run on a smaller system but will require a much longer run time. A full model run requires approximately 50 GB of space for all SQL Server database and TransCAD files.

The software can take advantage of an unlimited number of processors and memory. Extra memory and processors will make the model run faster. DRCOG’s current configuration is:

* Microsoft Windows 2008 R2 Enterprise Server OS
* Dell Poweredge R910
* Windows 64-Bit Operating System
* 8 quad-core processors for 32 total cores
* Hyperthreaded
* 64 GB memory
* RAID 1 OS drive,  RAID 5 Data drive
* 400 GB Hard Drive Space

Note: If you will doing many model runs, you will need approximately 50 GB per run. For example, if you need to archive 10 runs, you will need at least 500 GB of storage space.

# Software Requirements

The Focus travel model requires three pieces of software to be installed:

* Microsoft SQL Server 2008
* Microsoft Visual Studio 2008
* TransCAD 5.0, Build 2045

Changes to the Population Synthesizer require software to modify Java code, such as Eclipse.

# Downloading the code from GitHub

The Focus code is currently stored in two file sets: one for GISDK, and another for C#. Both file sets are stored in the GitHub system. To use GitHub you will need to install Git and obtain a GitHub account.

The following section describes how to install Git and start working with GitHab.

# Generic setup includes the following steps

* Install Git
* Generate SSH keys
* Send the public key to Git Admin (Privately)
* Add your SSH key to GitHub
* Let the Git Admin know your GitHub? username

## Install Git the Windows Way (Summarized)

Download Git: <http://code.google.com/p/msysgit/downloads/list?can=3>

When installing you will be asked a few questions (answer as listed below):

* "How would you like to use Git from the command line?" Select **Use Git Bash only**
* "How should Git treat line endings in text files?" Select **Checkout Windows-style, commit Unix-style line endings**

Install is now finished!

# Generate SSH keys

1. Right click on any folder and select **Git Bash** from the context menu.
2. type cd ~/.ssh
3. type ls  
   If you get a response with data such as id\_rsa and id\_rsa.pub then you already have a public key. Contact the Git Admin

If step 3 did not render a response then continue on

1. Generate a new SSH key

When asked for a passphrase just hit enter leaving it blank. Also accept all the defaults

ssh-keygen -t rsa

When finished you will be left with a new set of files. The key ending in pub is your public key and the file without a .pub but the same first set of characters as the file ending with .pub is your Private key.

**Very Important**: A public key is not exactly clear in saying that I am the key that you can give out but I am a very private key meaning that I only want to give it out to people that I trust.

1. Send the key to the Git Admin so you can be added to Git Internal
2. Log back into your GitHub Account and add your SSH key to GitHub?

### How do I get my public key text for GitHub?

From the Git Bash enter the following in order

$ cd ~/.ssh

$ notepad id\_rsa.pub

The *id\_rsa.pub* file will open in notepad where you can copy the key and then have it to past in GitHub

### Putting your key in GitHub

On the GitHub site *Click "Account Settings" > Click "SSH Public Keys" > Click " Add another public key"*

Paste the copied key info that we did above into "Key" field.

Hit "Add Key".

### Did my key get accepted by GitHub?

Test it out from the Git Bash

ssh git@github.com

The response should look something like this:

The authenticity of host 'github.com (207.97.227.239)' can't be established.

RSA key fingerprint is 16:27:ac:a5:76:28:2d:36:63:1b:56:4d:eb:df:a6:48.

Are you sure you want to continue connecting (yes/no)?

Don’t worry, this is supposed to happen. Type “yes”.

PTY allocation request failed on channel 0

Hi username! You've successfully authenticated, but GitHub does not provide shell access.

Connection to github.com closed.

# Final MUST do step: Set Up Your Info

You need to configure your local personal information.

1. Set your username and email  
     
   Git tracks who makes each commit by checking the user’s name and email. In addition, we use this info to associate your commits with your GitHub account. To set these, enter the code below, replacing the name and email with your own. The name should be your actual name, not your GitHub username.

Your email address needs to be the one used on GitHub

$ git config --global user.name "Firstname Lastname"

$ git config --global user.email "your\_email@youremail.com"

$ git config --global github.user username

# Obtaining and using the Focus code:

# Using Git and GitHub for the Focus model

Some additional information available at <http://utilsrvdev/trac/tools>

Once you have installed and configured Git and GitHub, you will need to get the latest code from GitHub. First let the DRCOG IT administrator (currently Mike Tafel ) know that you would like to be added to the DRCOG organization on GitHub and the TravelModel.Focus (C#) and TravelModel.GISDK (GISDK) repositories. They will need your GitHub user name to enter you into the system. You will need to pull down both the C# and GISDK code to run the model.

The following describes how to download and make changes to the code.

1. Go to the GitHub website, https://github.com and log in.

2. Switch your account context to DRCOG .

3. Click on the branch you want (on the right side)

                It will now show some sort of directory string in a little box in the upper middle (next to HTTP)

4. Now go to windows explorer, and make a directory where you want to download the code.Right click on that directory. Select GitBash here. In the DOS-window type *Git Init* to initialize your directory.

5. Next in the DOS-window type *Git clone*, followed by the long string that is associated with the branch you are trying to download (looks like this - [git@github.com:DRCOG/TravelModel.GISDK.git](mailto:git@github.com:DRCOG/TravelModel.GISDK.git))

                This should download the whole branch.

Note that you should only clone once, then for subsequent downloads, you “pull”. Now you have a copy of the code.

-----------------------------------------------------------------------------------------------------------------------------

If you want to make a change to the code, follow these directions.

NOW TO MAKE A CHANGE AND PUT BACK INTO THE SYSTEM, FIRST ADD TO YOUR LOCAL REPOSITORY AS FOLLOWS

1. Make some change to one of the files you have downloaded, and save.

2. Now right click on the directory again, and select GIT BASH HERE

3. IF YOU HAVE NEW FILE IN YOUR LOCAL DIRECTORY (ONE THAT DOESN’T EXIST ON GIT), YOU NEED TO add the file by typing: Git add FILENAME

4. Now type git commit -a -m “comment” (where comment is whatever you want to type) in the DOS window

5. THEN TO PUT ON THE WEB REPOSITORY. Type Git push in the DOS window.

6. Then go back on GIT website, click on the branch you’re working on (if you haven’t already), and refresh the web page.   You should see the new file

AND YOU’RE DONE!

Some more git stuff:

Full list: <http://utilsrvdev/trac/tools>

New to Git: <http://utilsrvdev/trac/tools/wiki/NewToGit>

Cheat sheets: <http://utilsrvdev/trac/tools/wiki/GitCheatSheets>

# TransCAD Setup and Input Creation

## TransCAD Installation

First install TransCAD 5.0, Build 2040 to the directory C:\Program Files (x86)\TransCAD\

After installing TransCAD, take the following steps to install Compass:

Open TransCAD, and select: Tools🡪Add-Ins Select “GIS Developers Kit” and click OK.

A dialog box will appear with five icons. The middle icon is “Compile to UI Database”. Select this middle icon.

From the files included on the CD, select the source code file and copy it to the directory C:\GISDKCode

DRCOG\_xxxxxxxx\_PartOne.rsc

DRCOG\_xxxxxxxx\_PartTwo.rsc

DRCOG\_xxxxxxxx\_CompileThis.rsc

The resource (rsc) code has three components for Focus. There are two code sections called PartOne and PartTwo and another file which ties the two together that has a suffix of CompileThis. This file includes the location of where PartOne and PartTwo are located.

Select the file to store the compiled program. The file to compile to is:

C:\Program Files (x86)\TransCAD\drcog\_ui.dbd

If the file does not exist, create it.

In TransCAD, select: Tools🡪Add-Ins. Click “Setup”. Click Add. Fill in the blanks as follows:

Type: Dialog Box

Description: DRCOG TransCAD Model

Name: DRCOG TransCAD Model

UI Database: C:\Program Files (x86)\TransCAD\drcog\_ui.dbd

Click “OK”, and the model is installed.

## TransCAD input files

The TransCAD part of the model scenario is defined by the following four elements:

Highway Geographic file (.dbd file),

Transit Route System (.rts file),

Socioeconomic Dataset (\*zone.bin file), and

Year (parameter).

These inputs are discussed in more detail below. It is possible to “mix and match” these inputs depending on the desired scenario. For example, the user might want to compare two highway geographic files with exactly the same transit route system, socioeconomic dataset and year. Alternatively, one might want to change both the highway geographic file and transit route system as a combined package of improvements. In addition to these variable inputs, the model uses a number of files that contain parameters, such as trip generation rates, that do not vary between model runs.

To set up a working directory follow these steps:

Create the directory from which you plan to work. You should only use one model run in each directory. Keep the path names short, because some TransCAD functions have a 64-character limit on file names.

Copy the contents of the “Inputs” directory from the CD or DVD into your working directory, or just copy an entire model run.

Place all the files associated with your highway geographic file (the combined highway geographic file) into your working directory.

Place all the files associated with your transit route system (the combined transit route system and combined transit base geographic file) into your working directory.

Copy the socioeconomic dataset (zonexxx.bin) into your working directory.

Having copied all of the model files from the CD to an appropriate directory, the next step is to add a scenario telling the model where to find those files.

Click the “Dir” tab, and select the appropriate directory.

When you are done adjusting any file names or parameters, click “OK” to exit the scenario dialog box, and you will be returned to the main model dialog box. You will notice that the name of the scenario selected will appear in the Scenarios Window.

Most of the model files and parameters are standard, and their names do not change from one scenario to another. There are some exceptions to this rule, and it is important to change these file names in the scenario manager if they are not set by default. The files and parameters you need to pay close attention to include:

**Highway and Transit Network Files:**

Found at: Initialization 🡪 Input Files

combined highway geographic file

combined transit route system

**Socioeconomic Zonal Data Set**

Found at: Trip Generation 🡪 Input Files

zone attribute table

**Model Year**

Found at: Trip Generation 🡪 Parameters

**Modeling forecast year (4 digits)**

These model inputs are described in more detail below. In addition to these inputs, it is important to ensure that the highway network includes reasonable speeds from which to start. These speeds serve as the starting point for skimming. The model will converge in fewer iterations of speed balancing if the starting speeds are reasonable. These are discussed below as well.

### Combined Peak & Off-Peak Highway Geographic File

This combined set of files is what actually gets edited when coding highway networks. It contains only the basic link and node attributes provided by the user, and any auxiliary attributes such as traffic counts that are not used in the modeling process.

When running the model, the “General Preprocess” step of “Initialization” copies this geographic file into separate peak and off-peak highway geographic files. All of the data in this combined file are copied. The “Peak Highway Network Processing” and “Off-Peak Highway Network Processing” steps process these files to add link attribute fields that include peak and off-peak travel times, capacities, free-flow speeds, and other items necessary for running the remainder of the model. These steps also compile the geographic files into peak and off-peak TransCAD network (.net) files. The .net files are not directly viewable by the user, but are used by the shortest path and highway assignment algorithms. Having this compiled .net file stripped of extraneous information greatly improves the processing time associated with these steps.

Three important points should be taken away from this discussion:

The combined highway geographic file should be edited when coding highway networks.

The peak and off-peak highway geographic files will be overwritten when the “General Preprocess” step of “Initialization” is run.

Changes made to the highway geographic file will not be reflected in the shortest path or highway assignment calculations unless “Peak Highway Network Processing” and “Off-Peak Highway Network Processing” are run to compile to the .net files.

### Combined Peak & Off-Peak Transit Route System

Like the combined highway file, the combined transit route system is the file that actually gets edited when coding transit networks. The “General Preprocess” selects the routes from the combined file that operate in the peak and saves them to the peak route system, and selects the routes that operate in the off-peak and saves them in the off peak route system. The “Peak Transit Network Processing” and “Off-Peak Transit Network Processing” steps compile these route systems into transit network (.tnw) files. These transit networks are used for calculating transit shortest paths and performing transit assignment.

With respect to transit route systems, remember:

The combined transit route system should be edited when coding transit networks.

The peak and off-peak transit route systems will be overwritten when the “General Preprocess” step of “Initialization” is run.

Changes made to the transit route system file will not be reflected in the shortest path or transit assignment calculations unless “Peak DB Preprocess” and “Off-Peak DB Preprocess” are run to compile to the .tnw files.

Note that the transit route system is linked to a highway geographic file; in this case the combined transit route system is linked to a file called TransitBase.dbd. The route system can’t be opened if the .dbd file isn’t in the same file folder.

### Zone Attribute Table

The zone attribute table contains the socioeconomic data by TAZ. In any cases where different development scenarios are evaluated, the population, households, and employment in this file should be changed. Socioeconomic data sets will generally be provided by DRCOG for a specific year and development scenario. They should only be changed when evaluating specific development alternatives.

The following 13 fields are inputs in the zonal attribute table, and need to be complete. The remaining fields in the table are generated within the model. The first six fields below are standard, and will not be changed between different scenarios.

ZONE ID: TAZ ID

DISTRICT: district names used in k-factors.

DIATime: pseudo travel time used for distribution trips to DIA.

UrbanCenter: flag indicating if the zone is part of an Urban Center

TAZ\_ID: TAZ\_ID by RSA\_ID (The 5 digit-ID used in economic modeling).

ACREAGE: zonal acreage (units of acres).

HH POP: total household population, not including population in group quarters.

LOW INC. HH: number of households low-income (bottom 11%)

MED. INC. HH: number of households medium-income (middle 64%)

HIGH INC. HH: number of households high-income (top 25%)

PROD./DIST. EMP: total employment in production and distribution

RETAIL EMP: total employment in retail.

SERVICE EMP: total employment in service.

### Modeling Forecast Year

The modeling forecast year is the year for which the model run is performed. The model uses this in two ways:

* External station volumes are factored up from a base year by applying a growth factor for the appropriate number of years. These volumes are used to estimate external-external and internal-external trips.
* The model forecast year is used with a lookup table to determine the number of enplanements at DIA, which is used to forecast the traffic at DIA.

### Starting Speeds

As will be discussed in the speed balancing section of this chapter, the model converges most quickly if it has good speeds from which to start. The starting speeds are stored on each link in the following fields of the combined highway geographic file:

* AM\_AB\_SPEED – AM peak period speed in AB direction,
* AM\_BA\_SPEED – AM peak period speed in BA direction,
* MD\_AB\_SPEED – Mid-Day period speed in AB direction, and
* MD\_BA\_SPEED – Mid-Day period speed in BA direction.

When the combined network is copied into the peak and off-peak networks, these speeds are copied as well. In the peak and off-peak networks, default speeds are filled-in if any are missing and travel times are calculated from those speeds. During speed balancing, revised speeds get fed back into the peak and off-peak networks, but the values in the combined network keep their original values.

The model will run with any set of starting speeds. However, using equivalent starting speeds can be important for comparing two scenarios or for reproducing a model run exactly. Because the speeds from the final iteration of a model run are only stored in the peak and off-peak highway geographic files, the only way to get exactly the same results is to copy those speeds back into the combined network and start from the beginning.

# SQL Server Database Input Creation Cheat Sheet

Here is a quick cheat sheet of how to set up database input files (once you get the hang of it).

|  |  |  |  |
| --- | --- | --- | --- |
| **Population Synthesizer** | | **Input to PopSyn** |  |
|  |  |  |  |
| **Regional Controls** |  |  |  |
| **Total Households in the Region in the** | Control\_ID | type | age |
| **following Categories** | 10 | OneAdultNoKids | 18-44 |
|  | 11 | OneAdultNoKids | 45-64 |
|  | 12 | OneAdultNoKids | 65 & over |
|  | 13 | OneAdultwithKids | 18-44 |
|  | 14 | OneAdultwithKids | 45-64 |
|  | 15 | OneAdultwithKids | 65 & over |
|  | 16 | TwoorMoreAdultsNoKids | 18-44 |
|  | 17 | TwoorMoreAdultsNoKids | 45-64 |
|  | 18 | TwoorMoreAdultsNoKids | 65 & over |
| **Zonal Controls** |  |  |  |
| **Total Households by Zone in the following Categories** | | HH= Households |  |
|  |  | i1= Number of Households in Zone 0-30K | |
|  |  | i2 = Number of Households in Zone 30-60K | |
|  |  | i3= Number of Households in Zone 60-100K | |
|  |  | i4= Number of Households in Zone 100K+ | |
|  |  | s1 =Number of Households of Size 1 | |
|  |  | s2= Number of Households of Size 2 | |
|  |  | s3= Number of Households of Size 3 | |
|  |  | s4= Number of Households of Size 4 | |
| **Total Households By Zone Overall** |  |  |  |
|  |  |  |  |
| **Job Points** |  |  |  |
|  |  |  |  |
| X |  |  |  |
| Y |  |  |  |
| TAZ05\_ID |  |  |  |
| Job TypeName- one of the six types | Education, Entertainment,Production,Retail,Restaurant,Service | | |
| **Housing Unit Points** |  |  |  |
|  |  |  |  |
|  |  |  |  |
| X INT, |  |  |  |
| Y INT, |  |  |  |
| TAZ05\_ID INT, |  |  |  |
| DistTransit REAL, |  |  |  |
|  |  |  |  |
| **School Points** |  |  |  |
| SchoolName | these are the ids for the school types | |  |
| X | 1 | PublicPreK-8 |  |
| Y | 2 | PrivatePreK-8 |  |
| TAZ05\_ID | 3 | Public9-12 |  |
| School Enrollment | 4 | PrivatePreK-8 |  |
| School Type (PK-8, HS, University) | 5 | University |  |
| **Zonal Data File** |  | Import To SQL Server |  |
| ***ZonalDataTemp*** |  |  |  |
| Zonal Data From GISDK Zones File |  |  |  |
| HHVirtHHBuffer INT, |  |  |  |
| HHVirEmpBuffer INT, |  |  |  |
| NumTransStops INT, |  |  |  |
| AverageDailyParkingCost REAL, |  |  |  |
| TotalEmployment INT, |  |  |  |
| EducationEmployment INT, |  |  |  |
| EntertainmentEmployment INT, |  |  |  |
| ProductionEmployment INT, |  |  |  |
| RestaurantEmployment INT, |  |  |  |
| RetailEmployment INT, |  |  |  |
| ServiceEmployment INT, |  |  |  |
| UniversityEmployment INT, |  |  |  |
| IntrDensEmpBuffer REAL, |  |  |  |
| ServDensHHCentroid INT, |  |  |  |
| EmpDensEmpCentroid INT, |  |  |  |
| EdDensEmpCentroid REAL, |  |  |  |
| EntDensEmpCentroid REAL, |  |  |  |
| ProdDensEmpCentroid REAL, |  |  |  |
| ServDensEmpCentroid REAL, |  |  |  |
| RetailDensHHCentroid REAL, |  |  |  |
| RetailDensEmpCentroid REAL, |  |  |  |
| RestaurantDensHHCentroid REAL, |  |  |  |
| RestaurantDensEmpCentroid REAL, |  |  |  |
| Private\_PK8Enrollment INT, |  |  |  |
| Public\_PK8Enrollment INT, |  |  |  |
| Private\_912Enrollment INT, |  |  |  |
| Public\_912Enrollment INT, |  |  |  |
| UniversityEnrollment INT, |  |  |  |
| MixedUseDensityHHBuffer REAL, |  |  |  |
| MixedUseDensityEmpBuffer REAL, |  |  |  |
| RuralAreaType INT, |  |  |  |
| CBDAreaType INT, |  | d |  |
| LNServDens REAL, |  |  |  |
| SchoolDistrictZone INT, |  |  |  |
| ParkingCostLowInc |  |  |  |
| ParkingCostMedInc |  |  |  |
| ParkingCostHighInc |  |  |  |
| NewDistrictName varchar(25), |  |  |  |
| NewDistrictID smallint |  |  |  |
|  |  |  |  |
| **PopSynOutput** |  |  |  |
| Import Households |  |  |  |
| Import Persons |  |  |  |

# Running the Population Synthesizer

If the land use model results have changed or you are running a new model year, you will need to run the population synthesizer.

The Population Synthesizer has three input files that must be created and saved at the beginning of the run:

1. Households by TAZ
2. Regional Control Totals on Households by Type
3. Households by Income Group and Size

The following three step process explains how to create the input files:

1. Create a .csv file for Households by TAZ for the forecast year. This is simply a land use model result by zone. Save the file to the directory PopSyn\inputs\DataTables.

2. Create another .csv file for Regional Controls for the forecast year for the following categories. The total of the regional controls must match the total of households by TAZ; scale the regional controls to match the total of households by TAZ if they do not. Save the file to the directory PopSyn\inputs\DataTables.

|  |  |  |
| --- | --- | --- |
| Control\_ID | Type | age |
| 10 | OneAdultNoKids | 18-44 |
| 11 | OneAdultNoKids | 45-64 |
| 12 | OneAdultNoKids | 65 & over |
| 13 | OneAdultwithKids | 18-44 |
| 14 | OneAdultwithKids | 45-64 |
| 15 | OneAdultwithKids | 65 & over |
| 16 | TwoorMoreAdultsNoKids | 18-44 |
| 17 | TwoorMoreAdultsNoKids | 45-64 |
| 18 | TwoorMoreAdultsNoKids | 65 & over |
| 19 | TwoorMoreAdultswithKids | 18-44 |
| 20 | TwoorMoreAdultswithKids | 45-64 |
| 21 | TwoorMoreAdultswithKids | 65 & over |

3. Create a third .csv file for households by zone with the following columns:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| HH | i1 | i2 | i3 | i4 | s1 | s2 | s3 | s4 |

HH= Households

i1= Number of Households in Zone 0-30K

i2 = Number of Households in Zone 30-60K

i3= Number of Households in Zone 60-100K

i4= Number of Households in Zone 100K+

s1 =Number of Households of Size 1

s2= Number of Households of Size 2

s3= Number of Households of Size 3

s4= Number of Households of Size 4

Save the file to the directory PopSyn\inputs\DataTables.

After your three input files have been created, you will need to set the properties of the year and file names.In the directory RunControl, create a new properties file by copying the file PopSyn2005.properties to a new file PopSynForecastYear.properties (where ForecastYear is the year for which you are forecasting).

Switch the input file names:

In line 220:

converted.tables =sf1\_1.csv,sf3\_1.csv,ctpp\_1.csv,Households2005.csv,zone\_05.csv,region\_05.csv

replace the last three files with the names of the input files you have just created.

In line 293:

Forecast.HouseholdsByTAZFile =Households2005.csv

Replace with the file for the Households by TAZ.

Switch the output file names:

Line 154:

design.ForecastTablePrefix =Y2005

Line 242:

BalancedSeedDistribution =2005Distribution.csv

Line 250:

HHFile =2005SynHH.csv

Line 256:

PersonFile =2005SynPerson.csv

Line 303:

Forecast.OutputSeedDistributionFile =2005SeedDistribution.csv

6. Create a new .bat file that references the .properties file you made:

java -Xms500m -Xmx1000m -Dlog4j.configuration=log4j.xml -cp "/PopSyn.DRCOG/Jar/PopSyn3.jar;/PopSyn.DRCOG/Jar/common-base.jar;/PopSyn.DRCOG/Jar/common-daf-v1.jar;/PopSyn.DRCOG/Jar/log4j-1.2.9.jar;/PopSyn.DRCOG/RunControl" net.JBowman.StdOpen.PopSyn.PopSyn PopSyn2005

By changing the bold portion here to the name of the new properties file. Save the .bat file to an appropriate name and run the .bat file by clicking on it or opening a new dos window and calling this file.

# SQL Server Input File Creation

To create a new scenario, you will need two sets of completely created files:

1. TransCAD inputs (as mentioned on the previous pages), including the four files that change in each scenario, and other unchanging inputs
2. Two SQL Server Databases:
3. One Model Structural Database that Does not Change called: IRM
4. One Data Database that Does Change that the user chosing the name for: IRM\_Data\_XYZ

You will also need two code bases compiled:

1. As mentioned before, the GISDK code will need to be compiled to the drcog\_ui.dbd
2. The C# Code will need to be compiled in Visual Studio, if any changes have been made.

## Creating New Databases

#### Setting the databases for the first time

Once you have population synthesizer results, you are ready to start creating your SQL Server databases. The databases run off SQL Server 2008 R2, which will need to be installed on the machine to which the code will connect.

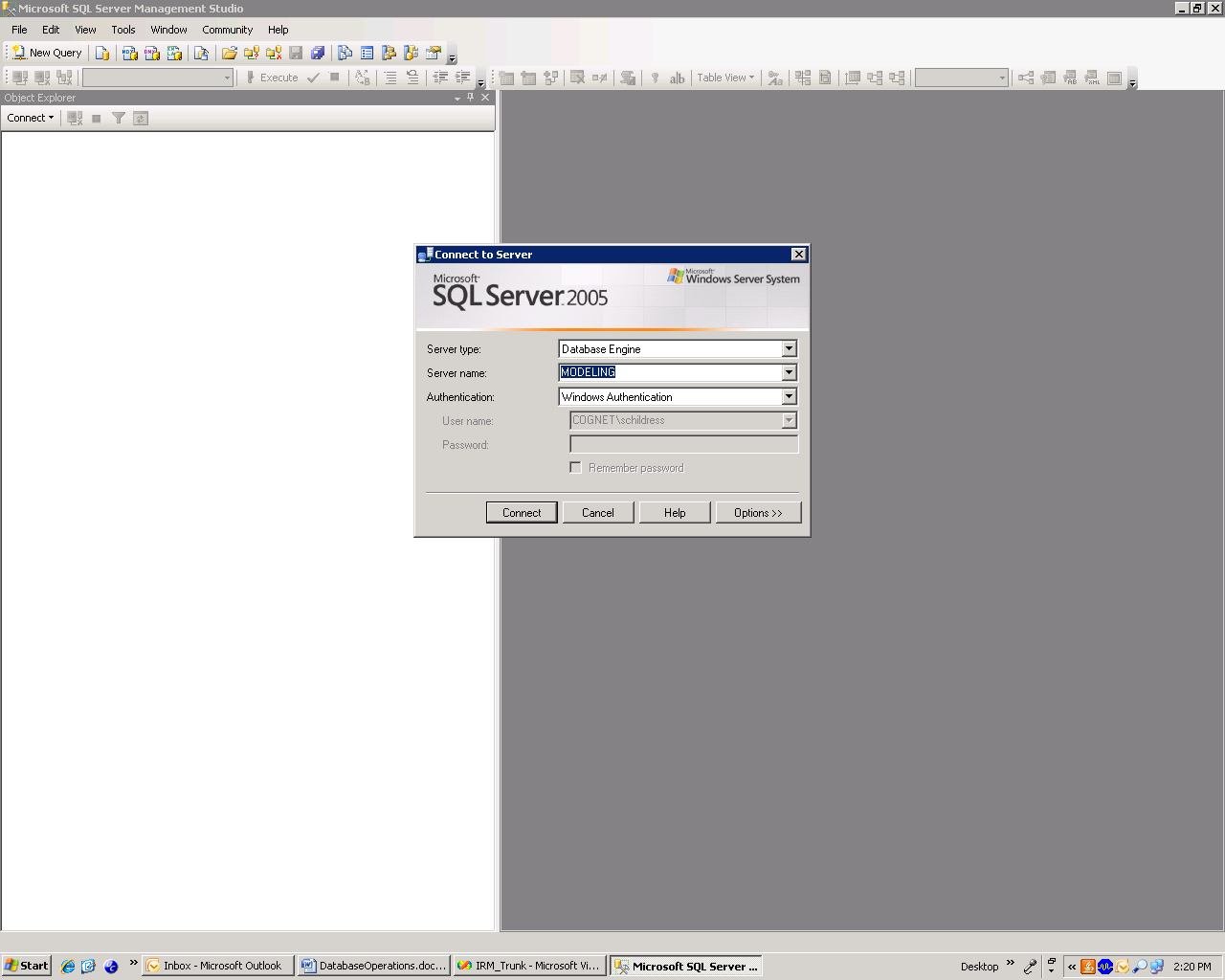
The IRM has two databases associated with it:

* a model structure database called “IRM”
* a scenario database that holds particular data for a scenario, that changes name depending on the scenario name

The IRM database holds the utility functions, coefficients, and model components, which are fairly static run to run. This database should just be able to be attached and not modified. The data database holds information specific to each scenario like all tables for all households in the region and their demographic data and point-level data for the location of employment.

#### Logging into SQLServer

Your system administrator should have given you a SQL account with Windows authentication. When you open SQL server it should provide a box like this. Use your regular Windows password to log in.



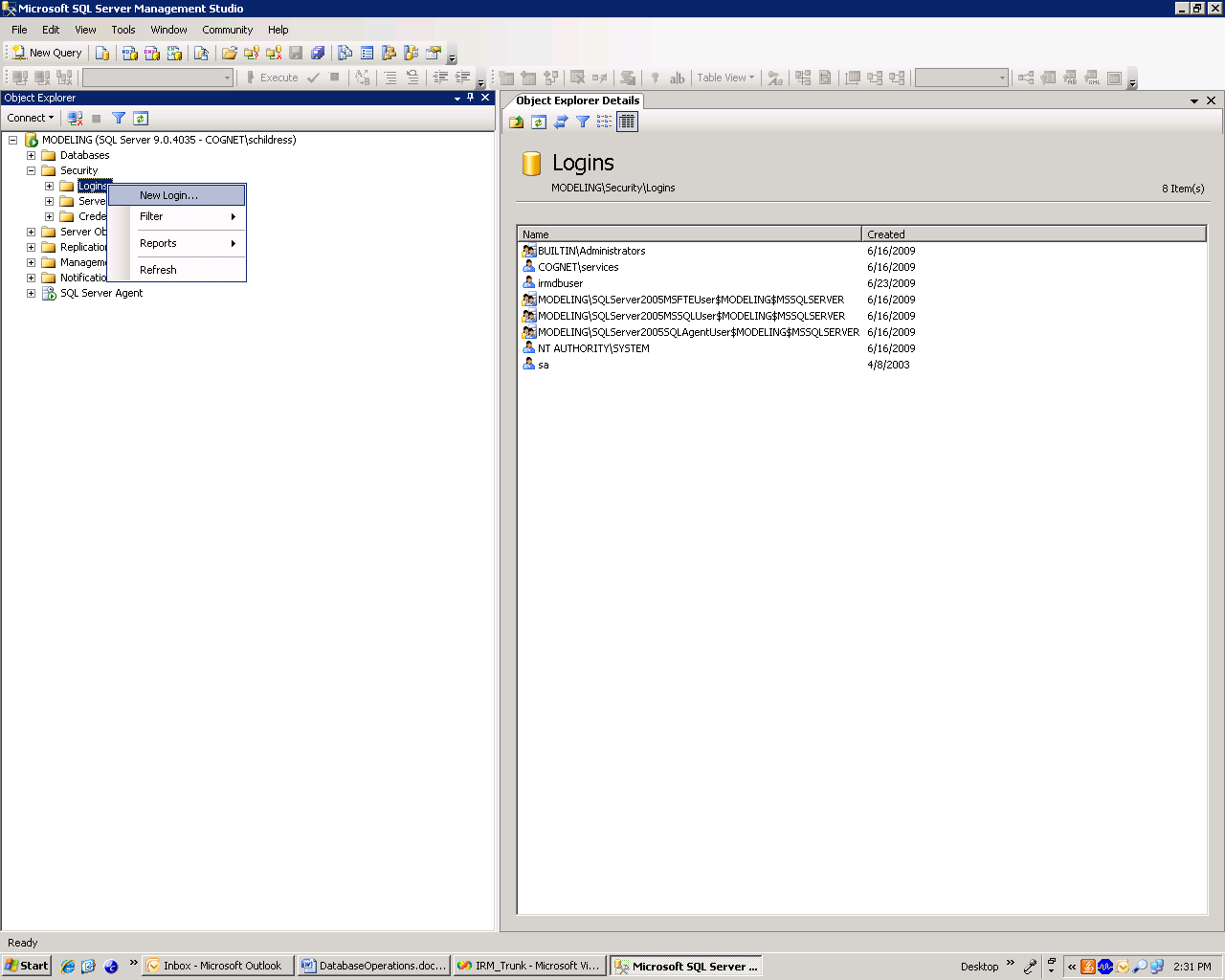
#### Attaching Databases

You will be sent databases in the form of .mdf (for the data) and .ldf(for the log file). Store them in a directory by themselves. Once you have logged into SQL Server, right click on databases and select **Attach Databases. Add** your IRM database and the data database to the server.

#### Adding Login for irmdbuser

The code accesses the database via the login name “irmdbuser” with the SQL Server Authentication password IRM

Under your server, left click **Security**, then right click **Logins** and select **New Login**.



For the **Login Name** type **irmdbuser** . Click the **SQL server** **authentication,** and type in and confirm the password **IRM** . For **default database** choose IRM, and default language English.

On the server roles page, check the boxes for all server roles. On the user mapping page, map the irmdbuser to the IRM and the data databases. Give the **membership type db\_owner** for irmdbuser by checking the box at the bottom.

**Refresh** the server by right clicking on the server name and selecting refresh.

The code should now be able to connect to the database.

## Filling IRM\_Data Database

The tables in the IRM database can be created, filled, and keyed using a set of stored procedures executed in the order prescribed in this section. These stored procedures require six input comma-separated text (.csv) files located and named as specified within the procedures. These csv files should be saved into a common location and will have their names set through SQL Server.

The **six input files** are the:

1. **Zonal File**
2. **Job Points File**
3. **Housing Points File**
4. **School Points File**
5. **Households File**
6. **Persons File**

The input files and their fields are listed below in greater detail. Most of these files are bulk-inserted as temporary files during the stored procedures, then used to fill the tables in the database structure, and deleted afterwards (with the exception of the households, persons, and employment category tables). In the following pages, first the fields are listed for each table, and the definitions for each field are provided.

## Input.csv files and fields

Six input .csv files will be created with the following heading names. Begin by copying templates from earlier runs to identify how your files should look. Below each of the six files and their expected fields are depicted in detail. Each field will described as too where to find the data.

All these files can found within the directory: S:\ModelRunNameHere\SQLInputDataCollection

## Zones

This file is created by using the zones.bin file from the COMPASS file and gathering GIS and land use forecast data.This file is always saved in the directory: S:\ModelRunNameHere\SQLInputDataCollection\ZonalData

To do a model run you will need to create a .csv file, but before that there is an .xlsx file you will make. The reason you need it to be an .xlsx file is that it performs calculations.

Fill your.xlsx file completely, then export to csv.

The .xlsx file is called ZonalDataTemplate\_WithCalculations and it is in the S:\ModelRunNameHere\SQLInputDataCollection\ZonalData directory. This file auto-creates some derived variables, i.e. say employment density divided by 100 ; it divides by 100 for you.

**FIELD DEFINITIONS: ZonalDataTemp-- All the following fields are created by exporting the zones.bin file to a csv and copying this into your ZonalDataTemplate\_WithCalculations.**

ZoneID- Table Primary Key, unique identifier for the zone, foreign keys to the Places table field PlaceID

The following fields are from the Compass model zones file. (This documentation is copied from the Compass Model Metadata)

TAZ05\_ID: TAZ ID in the 2005 numbering system

ZONE\_ID: TAZ ID used in COMPASS model

DISTRICT: district names used in k-factors.

DIATime: pseudo travel time used for distribution trips to DIA.

Zone Type: Values 0,1: not Boulder & Boulder.

Area Type: values 1 to 5.

Data Level: Not used

TAZ\_ID: TAZ\_ID by RSA\_ID (The 5 digit-ID used in economic modeling).

Centroid X: Not Used

Centroid Y: Not used

ACREAGE: zonal acreage (units of acres).

HH POP: total household population, not including population in group quarters.

LOW INC. HH: number of households low-income (bottom 11%)

MED. INC. HH: number of households medium-income (middle 64%)

HIGH INC. HH: number of households high-income (top 25%)

PROD./DIST. EMP: total employment in production and distribution

RETAIL EMP: total employment in retail.

SERVICE EMP: total employment in service.

The rest of the zonal data is generated for the IRM exclusively. Each of these fields, unless otherwise noted, needs to be filled in by hand by the modeler. The data comes from several sources including: the GISDK Parking Cost model, GIS processes, UrbanSim or BabySim, and the intersection density spreadsheet.

**SquareHalfSideLength -** Square root of the area of the zone divided by two; *This never changes unless your zone system changes. If it does re-calculate it as SQRT(AREA)/2*

**DIADummy –** 1 if Denver International Airport is in the Zone, 0 otherwise*; This never changes*

**TotalHHZone** – Total Number of Households in the Zone; Fill this from the land use model outputs

**HHVirHHBuffer–** Number of Households located within the 0.5 mi household virtual centroid buffer

Using the housing units created in BabySim or UrbanSim, first find the geographic virtual weighted by households centroid location. That is the "center of gravity" for the zone. Then count the number of housing units within a 0.5 mile radius of the virutal household centroid.

**HHVirEmpBuffer-** Number of Households located within the 0.5 mi employment virtual centroid buffer,

Using the jobs created in BabySim or UrbanSim, first find the geographic virtual weighted by employment centroid location. That is the "center of gravity" for the zone. Then count the number of jobs within a 0.5 mile radius of the virutal household centroid.

**NumTransStops-** Number of Transit Stops in the Zone--**DO NOT FILL; NOT USED**

**AverageDailyParkingCost-** Average Daily Parking Cost in the Zone; this is a result of the running the parking cost model from GISDK

**TotalEmployment-** Number of Employees in the Zone- Copy Land Use Model Output

**EducationEmployment**- Number of Education Employees in the Zone - Copy Land Use Model Output

**EntertainmentEmployment-** Number of Entertainment Employees in the Zone - Copy Land Use Model Output

**ProductionEmployment-** Number of Production Employees in the Zone- Copy Land Use Model Output

**RestaurantEmployment-** Number of Restaurant Employees in the Zone- Copy Land Use Model Output

**RetailEmployment-** Number of Retail Employees in the Zone- Copy Land Use Model Output

**ServiceEmployment-** Number of Service Employees in the Zone- Copy Land Use Model Output

**UniversityEmployment-** Number of University Employees in the Zone-- The spreadsheet calculates unviersity employment for you- Look at the cell formula; based on university enrollment

**IntrDensHHBuffer-** Intersection Density (4-way intersections + 0.5\* 3-way intersections – 1-way intersections) around Household Buffer:

Fill by Using Regression Spreadsheet- The regression spreadsheet is located in the directory: SQLInputDataCollection\IntersectionData;

it is called IntersectionDensity2010\_Calcs.xlsx; In this spreadsheet "IntersectionDensity\_Project", Copy the fields:

Copy the values into the column E, IntersectionDensityEmpBuffer into the Column IntrDensHHBuffer of your zonal spreadsheet. It was decided to use Intersection Density in the employment buffer for both IntrDensHHBuffer because this regression is very rough anyway so our approximation is not fine enough to capture the difference between the employment and household buffer.

**IntrDensEmpBuffer-** Intersection Density (4-way intersections + 0.5\* 3-way intersections – 1-way intersections) around Employment Buffer: Fill by Using Regression Spreadsheet- The regression spreadsheet is located in the directory: SQLInputDataCollection\IntersectionData;

it is called IntersectionDensity2010\_Calcs.xlsx; In this spreadsheet "IntersectionDensity\_Project", Copy the fields:

**EdDensEmpCentroid-** Number of Education Employees in 0.5 mile buffer around zone employment centroid;

First the zone employment centroid is found in a GIS process that calculates the employment weighted centroid of the zone is found. Next the number of education jobs within a 0.5 radius of the centroid is found by counting the points.

**EntDensEmpCentroid-** Number of Entertainment Employees in 0.5 mile buffer around zone employment centroid

First the zone employment centroid is found in a GIS process that calculates the employment weighted centroid of the zone is found. Next the number of education jobs within a 0.5 radius of the centroid is found by counting the points.

**ProdDensEmpCentroid-** Number of Production Employees in 0.5 mile buffer around zone employee centroid

First the zone employment centroid is found in a GIS process that calculates the employment weighted centroid of the zone is found. Next the number of production jobs within a 0.5 radius of the centroid is found by counting the points.

**ServDensEmpCentroid** – Number of Service Employees in 0.5 mile buffer around zone employee centroid

First the zone employment centroid is found in a GIS process that calculates the employment weighted centroid of the zone is found. Next the number service jobs within a 0.5 radius of the centroid is found by counting the points.

**RetailDensHHCentroid-** Number of Retail Employees in the 0,5 mile buffer around the zone household centroid

First the zone household centroid is found in a GIS process that calculates the household weighted centroid of the zone is found. Next the number of retail jobs within a 0.5 radius of the centroid is found by counting the points.

**RetailDensEmpCentroid-** Number of Retail Employees in the 0,5 mile buffer around the zone employment centroid

First the zone employment centroid is found in a GIS process that calculates the employment weighted centroid of the zone is found. Next the number of retail jobs within a 0.5 radius of the centroid is found by counting the jobs.

**RestaurantDensHHCentroid**- Number of Restaurant Employees in the 0,5 mile buffer around the zone household centroid

First the zone household centroid is found in a GIS process that calculates the household weighted centroid of the zone is found. Next the number of restaurant jobs within a 0.5 radius of the centroid is found.

**RestaurantDensEmpCentroid-** Number of Retail Employees in the 0,5 mile buffer around the zone employment centroid

First the zone employment centroid is found in a GIS process that calculates the employment weighted centroid of the zone is found. Next the number of restaurant jobs within a 0.5 radius of the centroid is found.

**Private\_PK8Enrollment-** Number of students enrolled in private PreK-8th grade schools in the zone

Using the forecast enrollment from the school modeling process, sum the total private pk-8 students enrolled in schools located within the zone. This is after you have placed all the future school locations and what the enrollment will be in each school.

**Public\_PK8Enrollment**- Number of students enrolled in publicPreK-8th schools in the zone

Using the forecast enrollment from the school modeling process, sum the total public pre-8 students in located within the zone. This is after you have placed all the future school locations and what the enrollment will be in each school.

**Private\_912Enrollment-** Number of students enrolled in private 9th-12th grade schools in the zone

Using the forecast enrollment from the school modeling process, sum the total private 9-12 students in located within the zone. This is after you have placed all the future school locations and what the enrollment will be in each school.

**Public\_912Enrollment-** Number of students enrolled in public 9th-12th grade schools in the zone

Using the forecast enrollment from the school modeling process, sum the total public 9-12 students in located within the zone. This is after you have placed all the future school locations and what the enrollment will be in each school.

**UniversityEnrollment-** Number of students enrolled in universities in the zone

Using the forecast enrollment from the school modeling process, sum the total university students in located within the zone. This is after you have placed all the future school locations and what the enrollment will be in each school.

**MixedUseDensityHHBuffer- YOU DO NOT NEED TO CALCULATE THIS, THE SPREADSHEET WILL CALCULATE FOR YOU**

Let resden:= # of households within 0.5 mile of household centroid of zone / 1000 = onhhh1/1000

Let retden:= Number of employees employed in retail or restaurants within 0.5 mile of household activity centroid / 1000 = (oempih15+oempih14)/1000

Then mixed use density:= (retden\*resden)/max(0.001,retden+resden)

**MixedUseDensityEmpBuffer- YOU DO NOT NEED TO CALCULATE THIS, THE SPREADSHEET WILL CALCULATE FOR YOU**

Let resden:= # of households within 0.5 mile of employment centroid of zone / 1000 = onhhh1/1000

Let retden:= Number of employees employed in retail or restaurants within 0.5 mile of employment centroid / 1000 = (oempih15+oempih14)/1000

Then mixed use density:= (retden\*resden)/max(0.001,retden+resden)

**RuralAreaType-** 1 if area type is rural, 0 otherwise **YOU DO NOT NEED TO CALCULATE THIS, THE SPREADSHEET WILL CALCULATE FOR YOU**

**CBDAreaType-** 1 if area type is CBD, 0 otherwise **YOU DO NOT NEED TO CALCULATE THIS, THE SPREADSHEET WILL CALCULATE FOR YOU**

**LNServDens-** Natural Log of number of jobs in 0.5 mile buffer around zone employment centroid

**YOU DO NOT NEED TO CALCULATE THIS, THE SPREADSHEET WILL CALCULATE FOR YOU**

**SchoolDistrictZone –**the school district ID the zone belongs to **THIS DOES NOT CHANGE**

**ParkingCostLowInc –** Daily Parking Cost for low income

COPY FROM THE OUTPUTS OF THE GISDK PARKING COST MODEL

**ParkingCostMedInc –** daily Parking cost for med income

COPY FROM THE OUTPUTS OF THE GISDK PARKING COST MODEL

**ParkingCostHighInc –** daily parking cost for high income

COPY FROM THE OUTPUTS OF THE GISDK PARKING COST MODEL

**NewDistrictName –** The name of the district of the zone (19 district system) **THIS DOES NOT CHANGE, unless you want to change the districting**

**NewDistrictID-** The id of the district of the zone (19 district system) **THIS DOES NOT CHANGE, unless you want to change the districting**

## Jobs

The Jobs Table holds the X, Y coordinates of every job in the region, as well as the type of job it is of six types. The file comes from QCEW data in the base year, economic/land use forecasts, and additional work to generate job points. See point forecasting documentation to see how to create these fields.

**FIELD DEFINITIONS: Jobs**

TempID – Temporary Job ID

X - X Coordinate of the Job Location in NAD-83 Colorado Central State Plane System

Y - Y Coordinate of the Job Location in NAD-83 Colorado Central State Plane System

TAZ05\_ID -TAZ ID in the 05 system where the X-Y point is found

JobTypeName – One of Six Types: Education, Entertainment, Production, Restaurant, Retail or Service

## HousingUnits

This file comes from electrical unit hook-up data , economic/land use forecasts, and additional work to generate points where the electrical units do not match the households by zone. Also it includes the distance from the housing unit to the nearest transit point. See point forecasting documentation to see how to create these fields.

**FIELD DEFINITIONS: HousingUnitTemp**

TempID- Housing ID on the Housing Unit File

X- X Coordinate where the housing unit is located in the NAD-83 Colorado Central State Plane System

Y- Y Coordinate where the housing unit is located in the NAD-83 Colorado Central State Plane System

TAZ05\_ID- TAZ ID in the 05 system where the X-Y point is found

DistanceToNearestTransit-The distance to the closest transit stop from the household point

## Schools

This table comes from a base year schools file and economic/land use forecasts. See point forecasting documentation to see how to create these fields.

**FIELD DEFINITIONS: SchoolTemp**

SchoolName- The name of the school

X – X Coordinate where the school is located in the NAD-83 Colorado Central State Plane System

Y- Y Coordinate where the school is located in the NAD-83 Colorado Central State Plane System

Enrollment- Number of Students enrolled at the school

TAZ05\_ID- TAZ ID in the 2005 system where the school is located

SchoolTypeID- Identifier for the type of school 1=Public Pre-K-8, 2= Private Pre-K-8, 3=Public 9-12, 4=Private 9-12, 5= University

# Households

This file is an output from PopSyn. For field definitions, see the PopSyn documentation.

## Persons

This file is an output from PopSyn. For field definitions, see the PopSyn documentation.

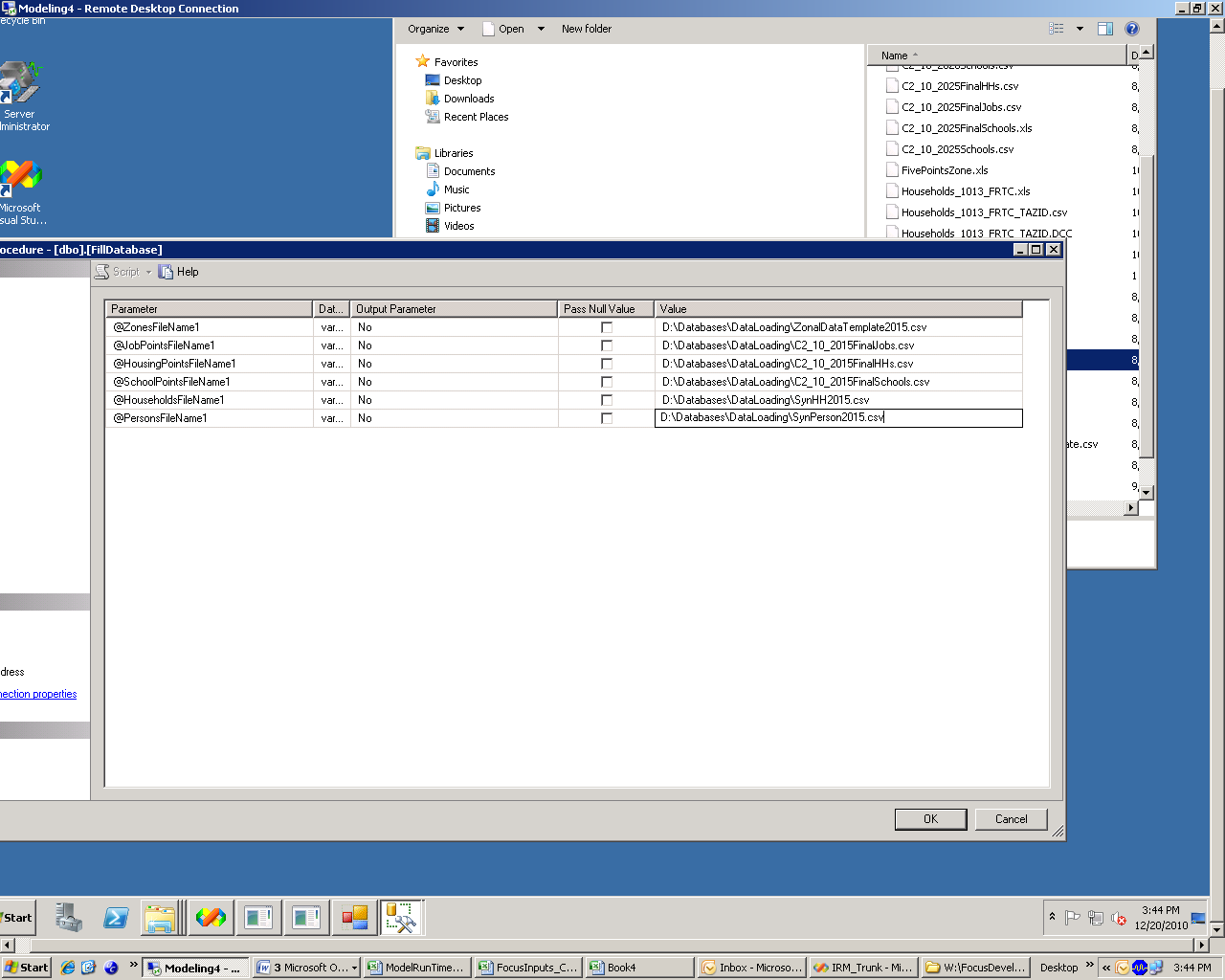
## Running the Input Initialization Stored Procedures

After you have created the six input files and saved them into a common directory, you can run the stored procedures to initially fill the database.

The stored procedures can be executed either in a step-wise fashion or by running the single procedure entitled FillDatabase.

For data items that will be filled later in the model operations, such as distance from home to work, the stored procedures fill the columns with zeroes to ensure that utility function values are zeroed out for the variables’ use in components.

To run the procedures, right click on the stored procedure entitled Fill Database.

A window will pop up, where you should type the directory and file name where each of the six files are located. An example of such a window is shown below. In the value field, you would type something what is shown below, in the value field:

Once you fill the file names, and begin executing the procedure the following input steps will be run and will take about an hour total. Then your database will be ready for running.

**Input Stored Procedure Steps**

1. CreateTables

2. FillLookupTables

3. ImportZonesFillZonePlaces

4. ImportEmpPlaces

5. ImportEmpPlacesStep2

6. ImportHousingPoints

7. ImportSchoolPoints

8. ImportPersonsHHsEmpCodes

9. AddDerivedFieldsHouseholdsPersons

10. FillDerivedPersonHHStepOne

11. FillDerivedPersonsHHStepTwo

12. FillDerivedPersonHHStepThree

13. FillDerivedPersonHHStepFour

14. FillDerivedPersonHHStepFive

15. FillDerivedPersonsHHStepSix

16. InsertZoneCentroidsAsHousingPoints

17. FillZoneIDsonPointsTable

18. UnNullColumns

19. AddSomeKeys

20. AddSomeMoreKeys

21. CreateIndices

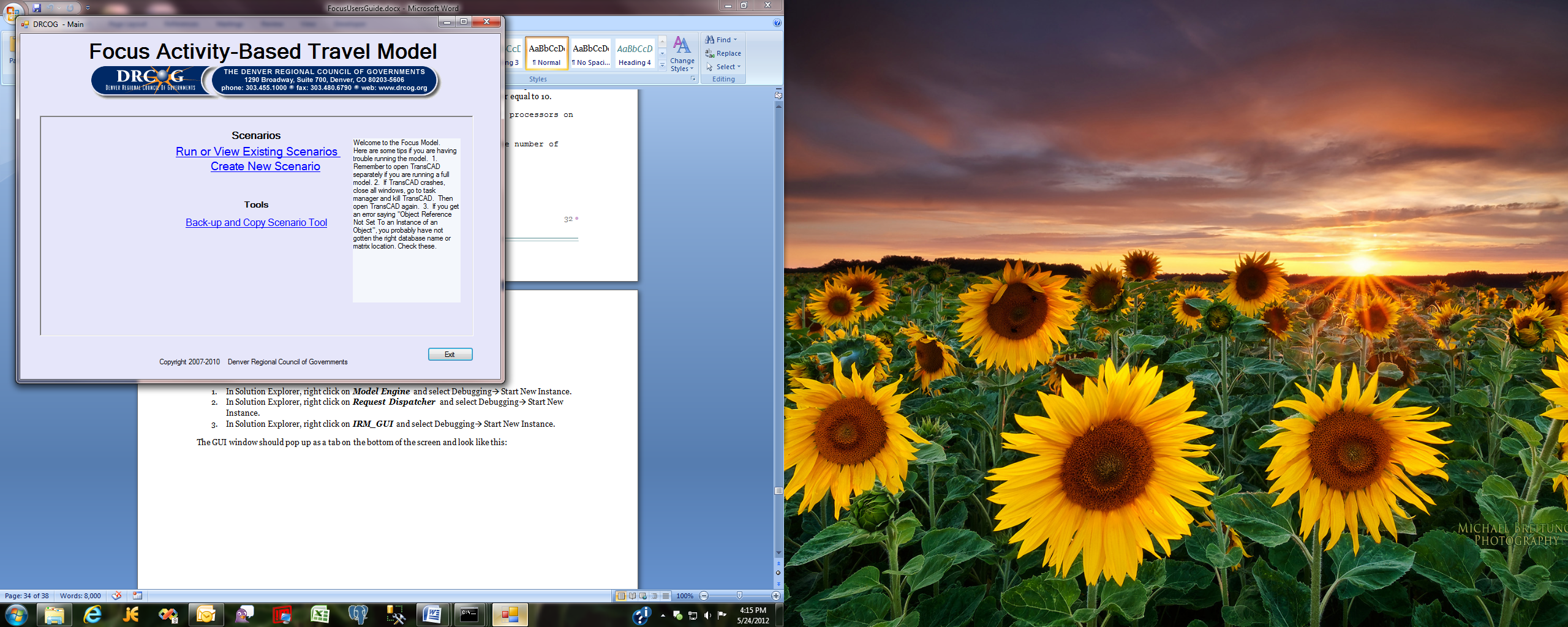
# Building the code and Starting a Model Run

There are two ways to start the user interface, either from the .bat file or from Visual Studio.

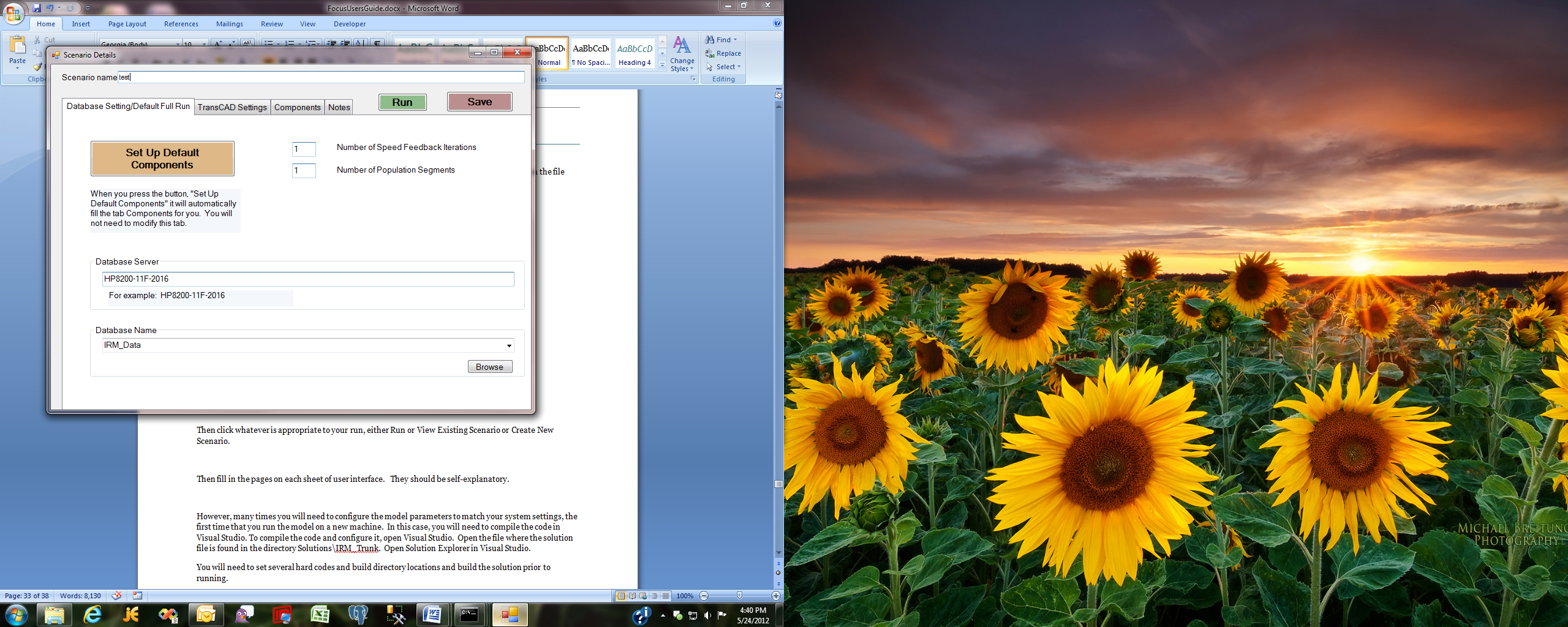
**Running from the .bat file**

If you are not making any changes to the code and do not need to figure the code, you can start the model execution by clicking on the file RunFocus.bat in the directory location where you C# is.

This will bring up the screen:



Then click whatever is appropriate to your run, either Run or View Existing Scenario or Create New Scenario.



Then fill in the pages on each sheet of user interface. They should be self-explanatory. If not, something needs to be written in this document about it :) I'm just guessing that it's straightforward.

**Running from Visual Studio**

However, many times you will need to configure the model parameters to match your system settings, the first time that you run the model on a new machine. In this case, you will need to compile the code in Visual Studio. To compile the code and configure it, open Visual Studio. Open the file where the solution file is found in the directory Solutions\IRM\_Trunk. Open Solution Explorer in Visual Studio.

You will need to set several hard codes and build directory locations and build the solution prior to running.

First you need to ensure that the location and name of the executable where you have TransCAD installed is:

C:\\Program Files (x86)\\TransCAD\\tcw.exe

If it is not, search the code for this string and replace it.

Next, you will need to configure the connection string to your database. Search the code for "Data Source". You will find a string like this:

Data Source=XYZ

Replace the name XYZ with the name of your database server.

You also will need to create a directory structure on your hard drive called:

c:\\inetpub\\wwwroot\\irm\\mcf\\

This directory is filled with the xml files for a scenario.

Finally, select a location that will hold component run time information about the run. It's default location is:

C:\\ TimeEstimate.txt

If you would like to change it, search the code for this string, and replace it.

In each component, two parameters control how the code is run in terms of how many threads and chunks of data are sent from the database. Find these strings in the code and set them equal to the desired values. This is sometimes hard and artful. If you don't know what to do, set the number of processors to be the number of processors on your machine and the number of chunks per processor equal to 10.

NumberOfProcesorsToUse- This should be at maximum the number of processors on your machine and minimum at 1.

NumberOfChunksPerProcessor- This can be any number from 1 to the number of obserations.

Then build your solution. You should be ready to run.

Next, start three services:

1. In Solution Explorer, right click on ***Model Engine*** and select Debugging🡪 Start New Instance.
2. In Solution Explorer, right click on ***Request Dispatcher***  and select Debugging🡪 Start New Instance.
3. In Solution Explorer, right click on ***IRM\_GUI*** and select Debugging🡪 Start New Instance.

The GUI window should pop up as a tab on the bottom of the screen and look like this.

# Current Unfortunate Hard-Codes

Eventually, all hard-codes should be removed from the code and put as configurable parameters in the user interface. At this point, several hard codes still exist and this section describes where they are located.

**Outer SpeedFeedback loops**

GISDKTransCAD.NetworkAssignment

line 28 outerSpeedFeedbackIterations

**Number of Segments (speed feed back iterations to divide the population into)**

Request.ScenarioRunRequest line 30

**Number of Zones**

several locations in C#

**Random Number Generation Seeds**

ModelComponent. MultinomialLogitModelComponent line 16

ModelComponent. NestedLogitModelComponent line 18

**Default Number of Chunks and Processors**

ModelComponent.ModelComponentWithThreads

chunks =1 ; line 19

line 84; number of processors on your machine

if (!nProcessorsToUse.HasValue) nProcessorsToUse = Environment.ProcessorCount;

**The Number of Chunks and Processor on Every Component**

Each component in the main component file, i.e. RegularWorkplaceLocation.cs the default number of chunks and processors to be used is overridden from the default.

**Location of TransCAD program files**

1. **transCAD.exe file**

ModelComponent.ModelComponent line 20-21

1. **drcog\_ui.dbd**

GISDKTransCAD.TransCAD line 14-15

# Model Steps

something

# Tricky Things

Occasionally the database will have a deadlock that crashes the code. The deadlock happens because two or more threads are trying to read or write to the same dataset at once. If this occurs, decrease the number of chunks and processors you are running on and rerun until you are successful.

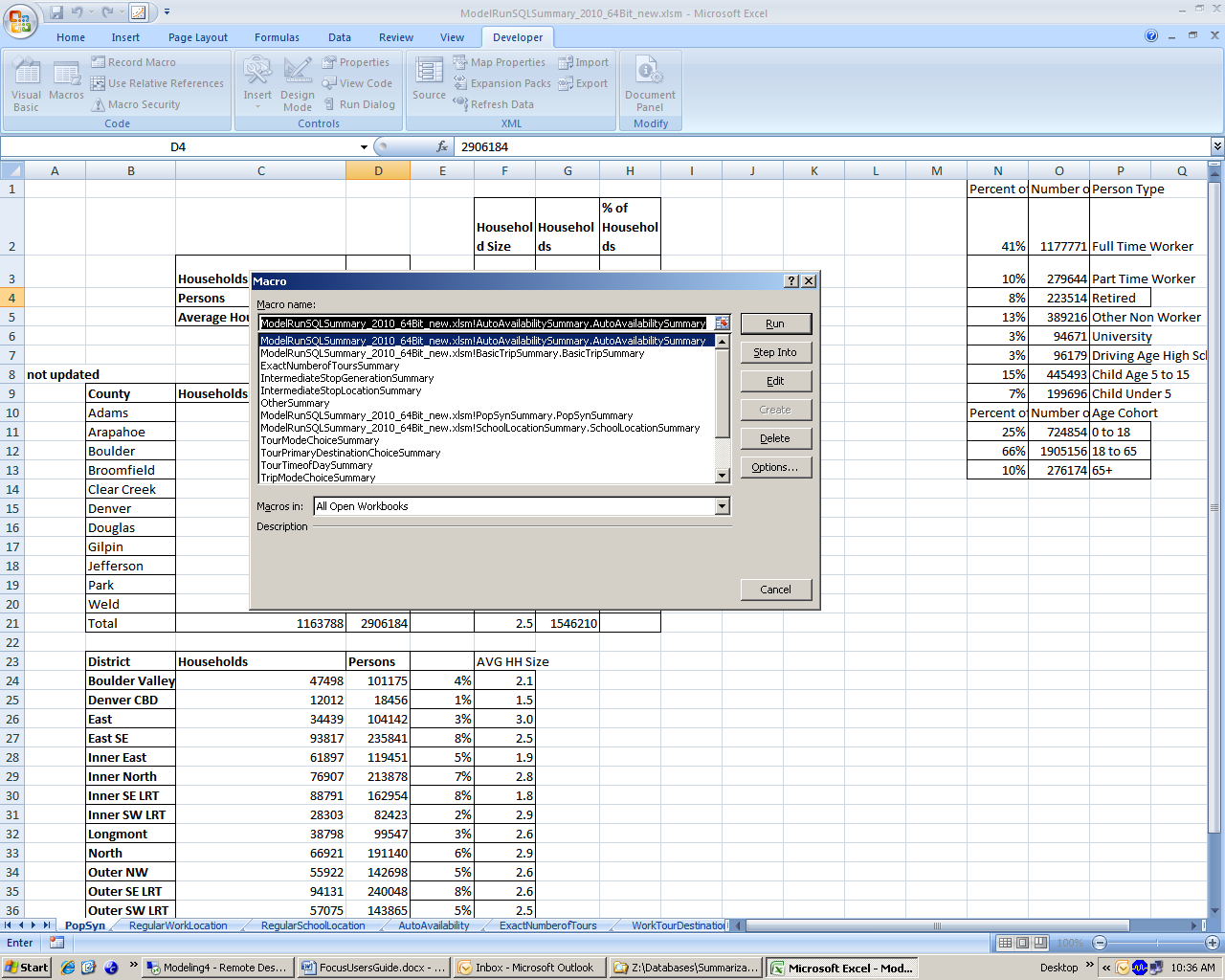
Be very careful about setting database and TransCAD location names. A typo will crash the code.

# Model Results Summarization

## Database Results

A Visual Basic Macro in an EXCEL workbook has been created to summarize model database results.

To begin the macro, copy the file ModelRunSQLSummary\_NameXYZ.xlsx onto a drive that can access the SQL server database. Open the file. Click on Developer ->Macros. A screen like this should pop up:



First, you will need to set the connection string. To do so , select the macro entitled “GettingConnectionString”. Then press the run button.

It will prompt you enter the connection string value. Type in the value in the following format:

PROVIDER=SQLOLEDB; Data Source=Modeling4;Initial Catalog= IRM\_Data\_2015\_C2\_10;User ID=irmdbuser; Password=IRM

This value will be written into the cell A1 of the first worksheet.

Next choose macros again, and select and run your macros. The results will be written into the corresponding worksheet.

## TransCAD Results