National Centre for Research Methods Working Paper

06/13

Microsimulation model user guide

Flexible Modelling Framework

Kirk Harland, TALISMAN node, University of Leeds

NCRM National Centre for Research Methods

E·S·R·C ECONOMIC & SOCIAL RESEARCH COUNCIL

UNIVERSITY OF LEEDS

Microsimulation model user guide

(Flexible Modelling Framework)

Kirk Harland

Version 1.0
September 2013

School of Geography, University of Leeds,
Leeds, LS2 9JT, United Kingdom

This Working Paper is an online publication and may be revised.

Contents

Figures

1 Introduction to static spatial microsimulation

Microsimulation as an analytical approach dating back to the work of Orcutt (1957) and Orcutt *et al*. (1961). The modelling approach constructs a dataset of individual units (persons, households or firms etc.) over a large area. To accomplish this, microsimulation uses a sample population to copy or clone individuals to match constraints specified using known aggregate data. Therefore, the distribution of constraint attributes in the synthesised population should match those observed in the constraint data.

Static spatial microsimulation takes this approach and effectively adds a strict spatial constraint to the synthesis process. Populations are created within smaller geographical areas specified by the user. This spatially explicit approach has had a number of diverse applications including the analysis of disease prevalence in health studies (Brown & Harding 2002; Smith, Pearce, & Harland 2011; Tomintz, Clarke & Rigby 2008), transportation analysis (Beckman, Baggerly, & McKay 1996; McFadden, Cosslett, Duguay, & Jung 1977) and studies associated with demand estimation such as Williamson and Clarke (1996) water demand estimations.

Static spatial microsimulation uses survey data to sample from and aggregate data, Census data for example, to constrain the sampling process. The resulting synthetic population contains the attributes from both aggregate and survey data. This allows estimates of demand or prevalence to be made for small geographical areas using attributes not collected at that spatial resolution, a powerful tool for policy formation and local or regional planning. Alternatively, the synthetic population produced by a static spatial microsimulation model can be used as the base population for other individual level models such as dynamic microsimulation, which moves the synthetic population through space and time, or agent-based modelling for either enriching the model environment or to produce the acting agents.

1.1 Idealised example

To exemplify how the process works a simple idealised example is worked through below. Table 1 shows the example survey population which will be used as the sample population in this example.

id	sex	age
1	m	20
2	f	30
3	m	30
4	f	20
5	m	30

Table 1: Sample population

There are two constraint tables, age and sex. The constraint table for age is shown in Table 2 and sex is shown in Table 3. Both tables contain a zone field identifying the geographic zone for the aggregate counts for each row. Each row represents one zone. Each field following the zone field contains information relating to the attributes within that constraint. For Table 2, the first field contains the counts of people aged up to 20 for each zone, the second field contains the counts of people aged between 20 and 30. For Table 3 the counts of males (m) and females (f) are shown for

each zone. Both tables contain information for three zones a, b and c. The total population for each zone in this simple example is very small, zone a = 20, zone b = 10 and zone c = 12.

Zone	20	30
a	7	13
b	4	6
c	9	3

Table 2: Age constraint table

Zone	m	f
a	5	15
b	5	5
c	4	8

Table 3: Sex constraint table

The constraint tables can be matched against the fields in the sample population, in this example the fields in the constraint tables match the attributes within the sample population perfectly i.e. the sex field in the sample population has values of 'm' and 'f' which match the fields in the constraint table, the same is true of the age table. In reality this is rarely the case and pre-processing is generally required to get the constraint and sample population information to match.

The static spatial microsimulation process works through a zone at a time and clones individuals from the sample population into the synthetic population while matching the sum of individual's attributes to the constrained counts. For zone a, the synthetic population may look like that contained in Table 4.

Zone	id	sex	age
a	1	m	20
a	1	m	20
a	1	m	20
a	1	m	20
a	1	m	20
a	4	f	20
a	4	f	20
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30
a	2	f	30

Table 4: Example synthetic output for zone a

Counting the individuals in the example solution for zone a shows that we have 5 males, 15 females, 13 20 to 30 year olds (30) and 7 individuals aged 20 or below (20). Table 4 is only one configuration of cloned individuals from the sample population that match the constraint counts. There are many different configurations for a simple problem such as this that would match the constraints. However, it is unlikely that any real world model would be as simple as this and as the model complexity increases the number of suitable solutions decreases compounded by issues of real world data quality.

Furthermore, to increase the model accuracy cross-tabulated attribute information can be used as the constraint increasing the information captured in the model and passed to the resulting synthetic population. In this example this would involve using counts of individual for all age and sex combinations, m-20, f-20, m-30 and f-30. It is easy to imagine how the number of constraining attributes can escalate quickly increasing the complexity of the model. It is for the researcher to establish a balance between model complexity and accuracy.

2 Installing the software

Installing the software is just a matter of copying all of the files in the distribution onto your local machine. The software does not need administrator privileges to install, you just need to have read and write access to the area of the machine where you copy the program files to.

The software requires the Java Runtime Environment (jre6) or later to operate, please make sure it is installed before trying to start the application.

The software can be downloaded from https://github.com/MassAtLeeds/software/releases

3 Starting the software

Starting the software can either be accomplished by double clicking on the exec.jar file if jar files are set to execute on the machine you are using. If double clicking causes no action it is likely that .jar files are not associated with java, to create the association on a windows machine see:

http://windowstipoftheday.blogspot.co.uk/2005/10/setting-jar-file-association.html

If configuring .jar files to execute with java cannot be achieved, start a command window navigate to the folder where the exec.jar file has been copied and type in 'java -jar exec.jar' and press return.

As the application starts the splash screen shown in Figure 1 will be displayed while all of the code modules are loaded and checked.

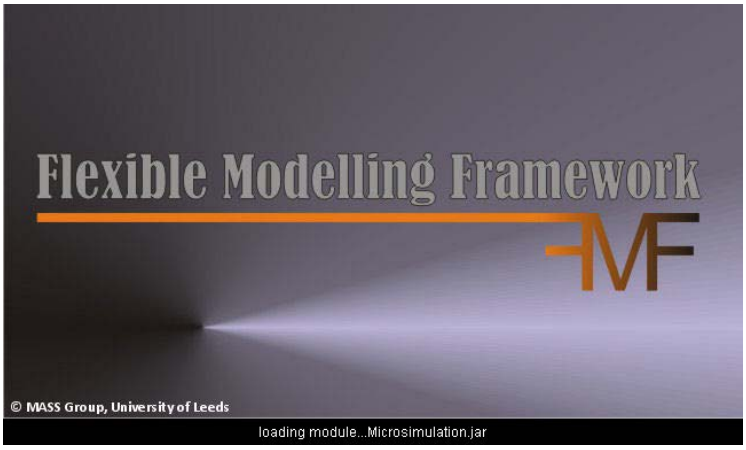


Figure 1: Splash screen for Flexible Modelling Framework

Once the application has successfully loaded the main screen will be visible, Figure 2.

Figure 2: Main screen of the Flexible Modelling Framework

3.1 Navigating the screen

The screen has four main areas:

1. The 'menu ribbon' across the top of the screen, contains the Application and Microsimulation menus in the screen shot above.

2. The large dark grey 'action area' below the menu ribbon which is where any action windows appear.

3. The light grey tabbed pain to the right containing three tabs:
    a. Processes – this is where the progress of any operations in progress can be observed
    b. Windows – a list of all open windows in the main action window are displayed here
    c. Data Sources – contains a tree of all registered data sources tables and fields
4. The white strip at the bottom is the reporting window and where any non-critical information notices will be displayed.

3.2 Application menu

On the menu ribbon the first option is Application. This menu can be dropped down by clicking on it or by holding down the 'Alt' key and clicking 'A' at the same time. The menu options can be seen in Figure 3 below.

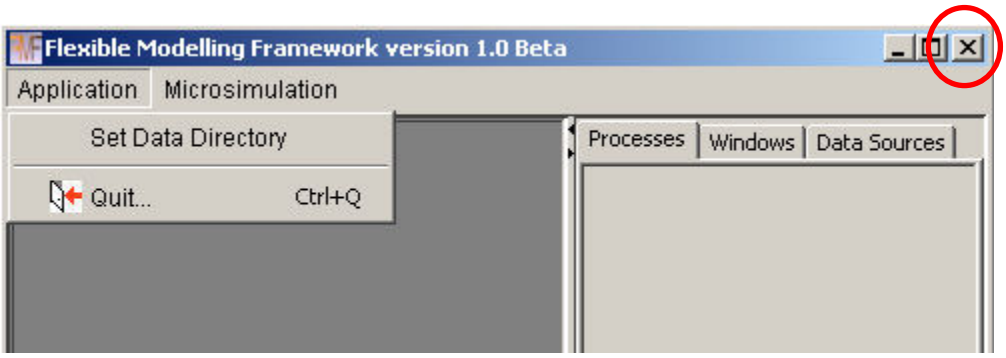


Figure 3: Application menu

3.2.1 Shutting down

To quit the application select the bottom option 'Quit' or click 'Q' while holding down the 'Ctrl' key note that the application menu does not have to be displayed for this shortcut combination to cause an exit to occur. Alternatively, exiting the application can be achieved by clicking on the cross hair box in the top right hand corner of the application main screen highlighted in Figure 3 above.

3.2.2 Setting the data directory

The user has the option to set the root data directory. Clicking on 'Set Data Directory' will launch a directory chooser dialog box shown below in Figure 4. Simply navigate to the directory on your computer that you would like to use as the root source for your data selection and click open. Please note that this is a convenience option to eliminate long repetitive file searches and can be overridden when actually selecting your data. This option simply means that when data is being selected the selection dialog will open at your preferred data directory. This option can also be changed at any time without any adverse effects.

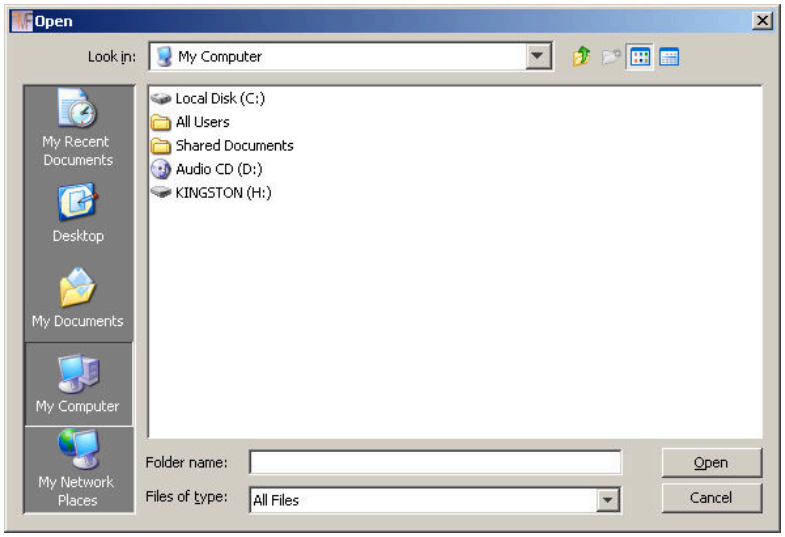


Figure 4: File chooser dialog

4 Data Sources

Before we can use the application to do any modelling activities we need to link to the data required to undertake the activities. In the application there is the concept of a data source. This is the location of data to be used in a modelling process or a location where data is to be saved.

4.1 Registering and unregistering a data source

To register a data source click on the Data Source tab at the top right hand side of the main screen highlighted in Figure 5 below.

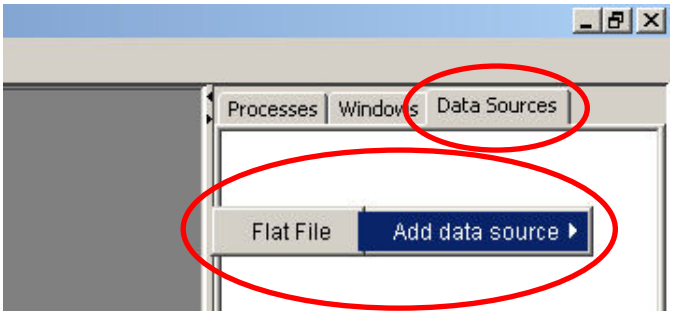


Figure 5: Data sources tab

Right click anywhere in the white area to see the context menu to register a data source, displayed in Figure 5. As the 'Add data source' option is entered a second menu with the data source types becomes visible, currently the only option is to read and write flat files such as .csv. Selecting the 'Flat File' option will open a window in the main action area of the screen for importing data from a flat file format, Figure 6. The application links to the data source, storing information about the location and format of the file rather than copying the data into a local area.

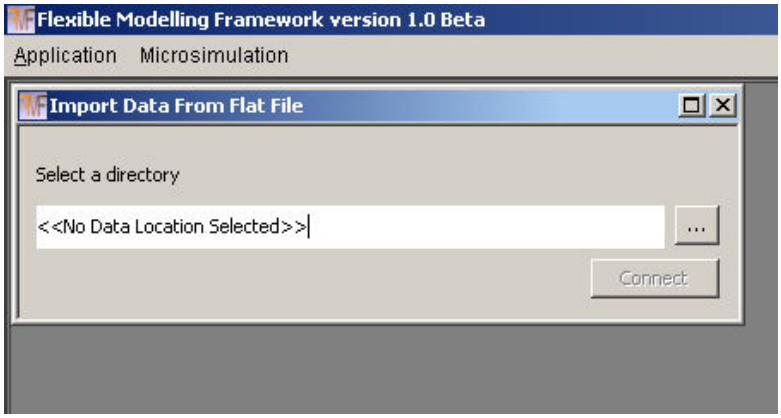


Figure 6: Import data window

To link to a data source you can either type directly into the text area denoted by '<<No Data Location Selected>>' or click on the button with '…' on it to launch a directory chooser dialog window. If you choose to type the location of the data source directory into the text area and the path or directory is incorrect then the statement 'could not connect to data source <<data source name entered>>' will appear in the reporting area at the bottom of the application. The safest and most flexible way to register a data source is to click on the '…' button. A dialog like the one below in Figure 7 will appear. The dialog will open at the folder set through the 'Set Data Directory' option in the 'Application' menu, below this is 'C:\Work\PostDoc\Population Synthesis' on a windows machine. If the data directory set does not exist or has not been entered the dialog will open at the default location for the local machine.

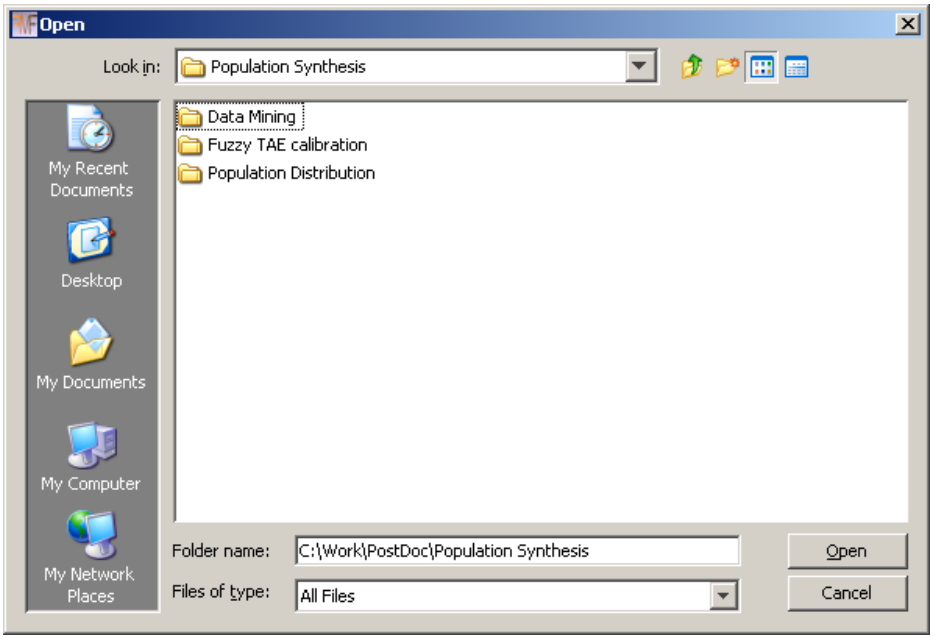


Figure 7: Open file dialog

The dialog can be navigated in the normal way with new folders being added and moving up or down levels until the desired folder has been located. Once this is highlighted click open and the full path and name of the folder should be displayed in the 'Import Data From Flat File' windows text area. If the correct path is shown click connect otherwise launch the file chooser dialog again and attempt to navigate to the correct folder.

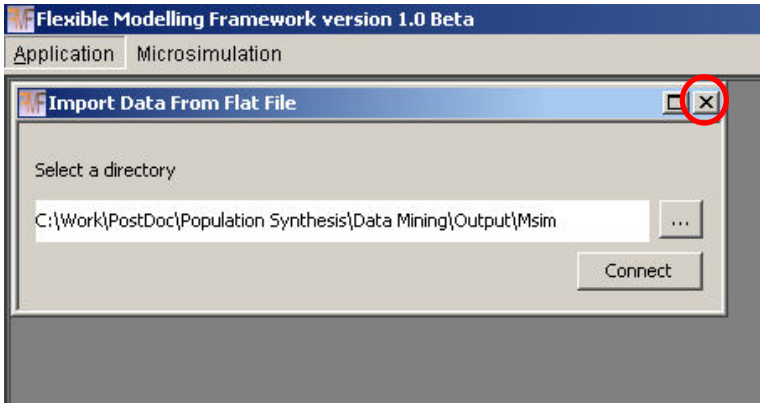


Figure 8: Import data window with file path specified

Once a directory has been chosen and the connect button clicked the 'Import Data From Flat File' will reset. More data sources can be added at this point or the import window can be closed using the cross hairs in its upper right corner, highlighted in Figure 8. The data source(s) added should appear in the 'Data Source' tab to the left as shown in Figure 9 below.

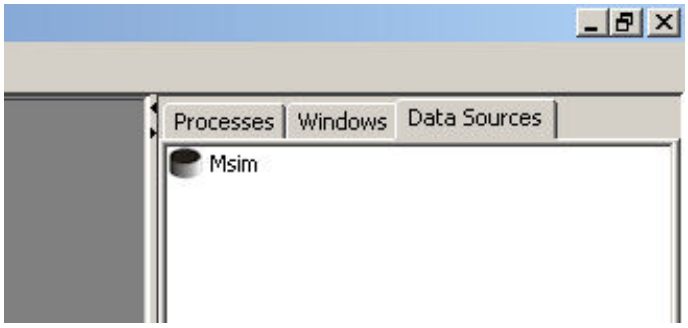


Figure 9: Data source in the data sources tab

The black cylinder shape denotes a correct connection to the underlying data source has been established. Double clicking on the data source will expand it showing the 'Properties' and 'Tables' that belong to this data source. Currently this data source does not have any tables therefore there is a large dot displayed next to the 'Tables' whereas the 'Properties' has a folder and a '+' symbol next to it, Figure 10.

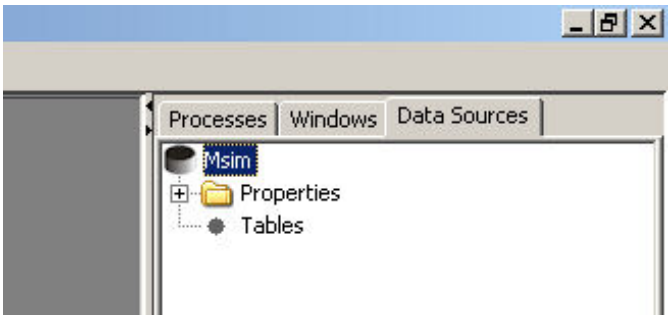


Figure 10: Expanding the data source

The properties for the data source can be expanded by clicking the '+' symbol, Figure 11. The fully qualified file name is displayed and the data type although, currently, only Flat File is available.

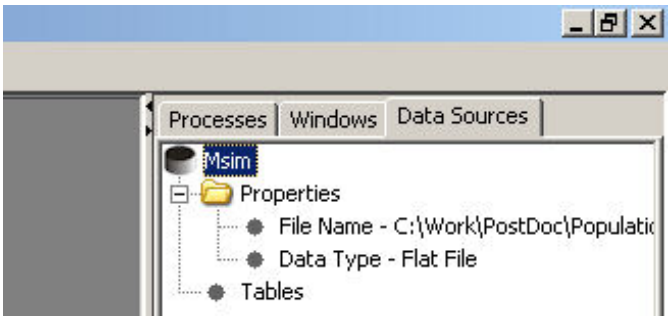


Figure 11: Data source properties

Highlighting a data source and right clicking again brings up a context menu, Figure 12 this time a few more options are available. The 'Register file' option is considered in section 3.2 below. The clicking the 'Remove data source' option will remove all record of the data source from the application. This does not affect the underlying directory and files on the computer these will remain in place, this action simply removes the link to the files from the application.

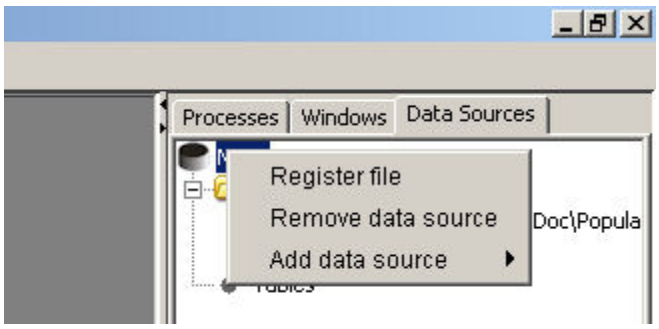


Figure 12: Data source context menu

4.2 Registering and unregistering files

Within the data source we need to register the data files. This is purely the process of telling the application what format the files are in so that the data can be loaded correctly. The 'Register file' option can be accessed by right clicking on the data source as shown above or by right clicking on the 'Tables' node, Figure 13 below.

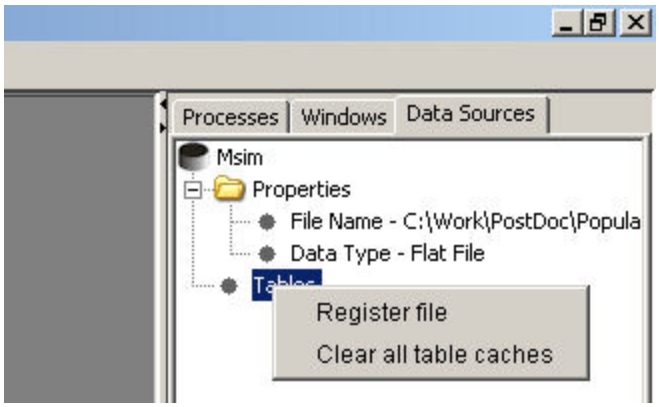


Figure 13: Tables context menu

Once the 'Register file' option has been selected a window called 'Register Flat Files – <<data source name>>' will be displayed in the action area, Figure 14.

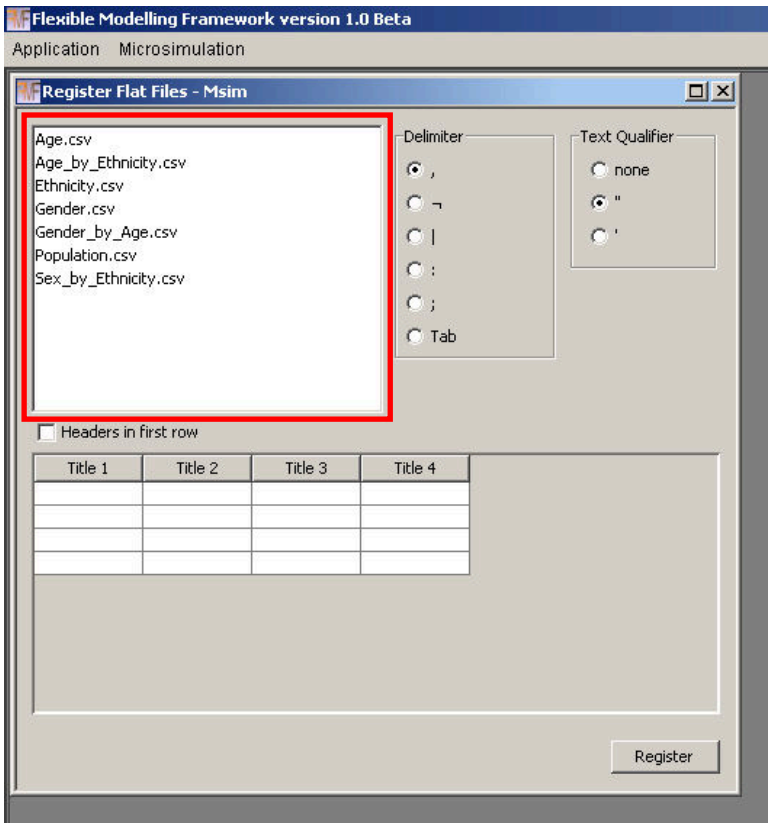


Figure 14: Register tables window

A list of files available to be registered in the data source is shown in the area highlighted by the red box in Figure 14. Selecting a file from the list will load the top 10 rows of the data and display them in the sample are shown in Figure 15 below. In this case the file Age.csv has been selected and the sample data is shown in the area highlighted by the red box.

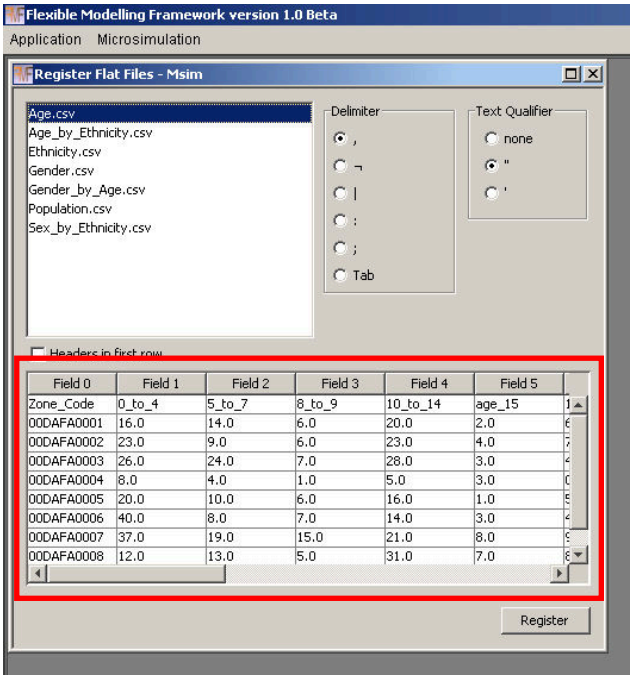


Figure 15: Register flat file window with file selected

Notice that the field names have been automatically added as ‘Field 0’, ‘Field 1’ etc. If your data has field names in the first row select the ‘Headers in first row’ check box as shown in Figure 16 below to use these headers.

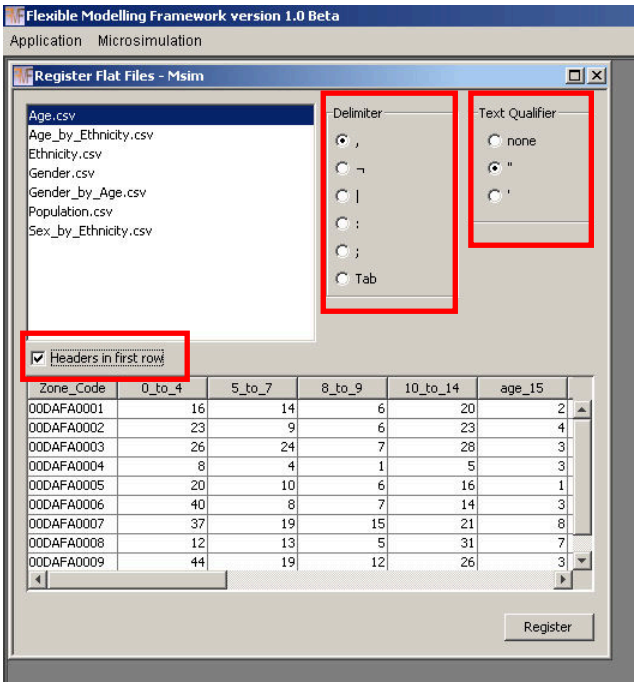


Figure 16: Register flat file headers delimiter and text qualifiers

Notice that the headers are now named as contained in the first row of data. A selection of common file delimiters and characters used to enclose text fields can be chosen in the area highlighted to the top right in Figure 16. The default for these two options is a comma for the delimiter and double quotes for text encapsulation, this is the format used to save all flat files generated by the application.

The field types will be suggested by the application from the sample of top 10 rows. If a text value is found in these rows the type suggested for the field is text, otherwise it is suggested as a numeric of double precision. Notice in Figure 15 the sample data is all left aligned in the columns, indicating text. This is because the first row does indeed contain text for the field headings. In Figure 16 when the first row is used as a header, only the first field has text detected and is therefore left aligned, all other fields are suggested to be numeric and are aligned to the right to show this. To override the suggested typing of the field you can right click anywhere in the data area for a field and a context menu showing the current data type will appear and allow you to alter this to a different type if required, Figure 17.

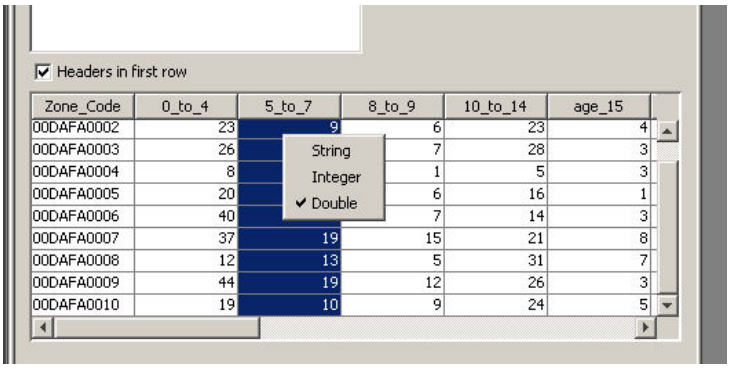


Figure 17: Data type context menu

Once all settings have been correctly selected click the 'Register' button in the bottom right of the window and the file will be registered and should appear in the 'Tables' area of the data source, Figure 18. Note the 'Tables' icon has now changed to a folder and it has been expanded in the same way as explained for the properties option above.

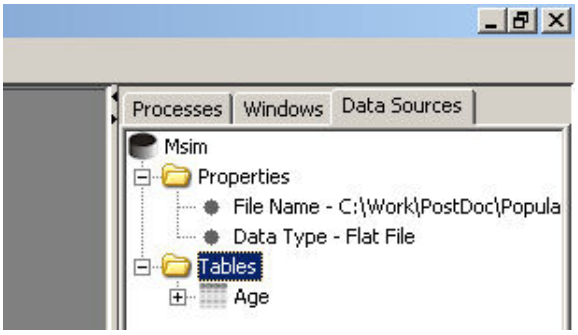


Figure 18: Registered table appears in data sources tab

Clicking on the cross to the right of the table ‘Age’ displays the fields in the table and how they have been defined, as demonstrated below in Figure 19. Each field has an icon displaying either ‘abc’ or ‘123’ next to it demonstrating if the field type is registered as a text field or a numeric field.

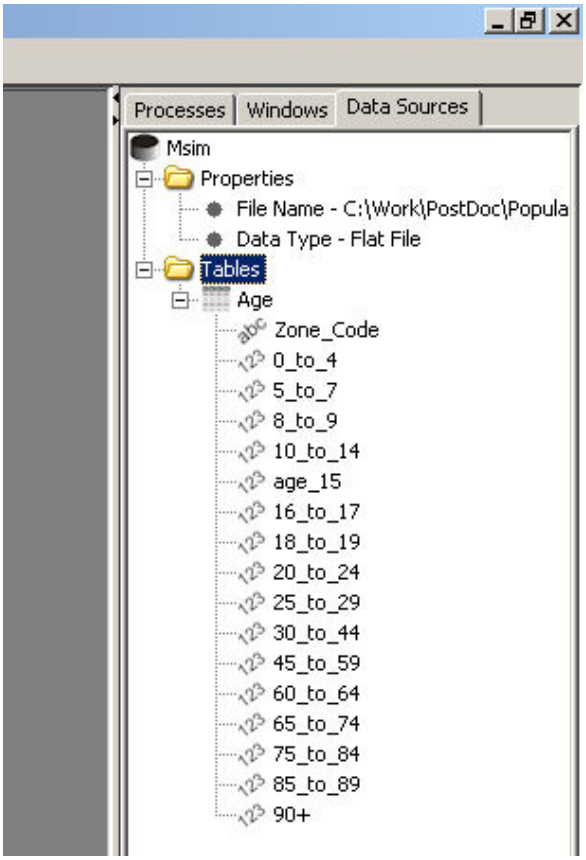


Figure 19: Expanded table showing field types

The registered file is no longer available in the ‘Register Flat Files’ window as shown below in Figure 20 the Age.csv file is no longer visible. The last files registration details remain set in the window and the ‘Register’ button will be disabled until another file is selected to be registered.

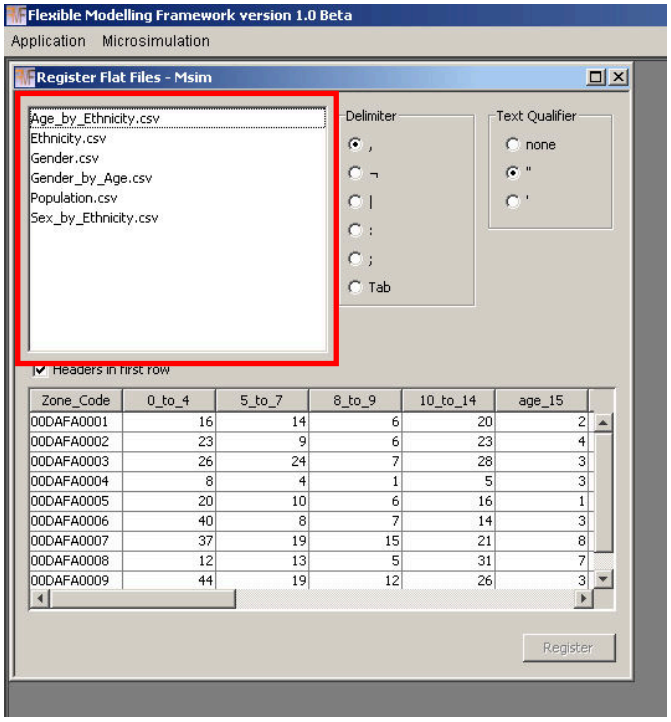


Figure 20: Files already registered no longer show in file selection area

To unregister a table right click on the table in the 'Data Sources' area to show a context menu, demonstrated below in Figure 21. Select the bottom option 'Drop table'.

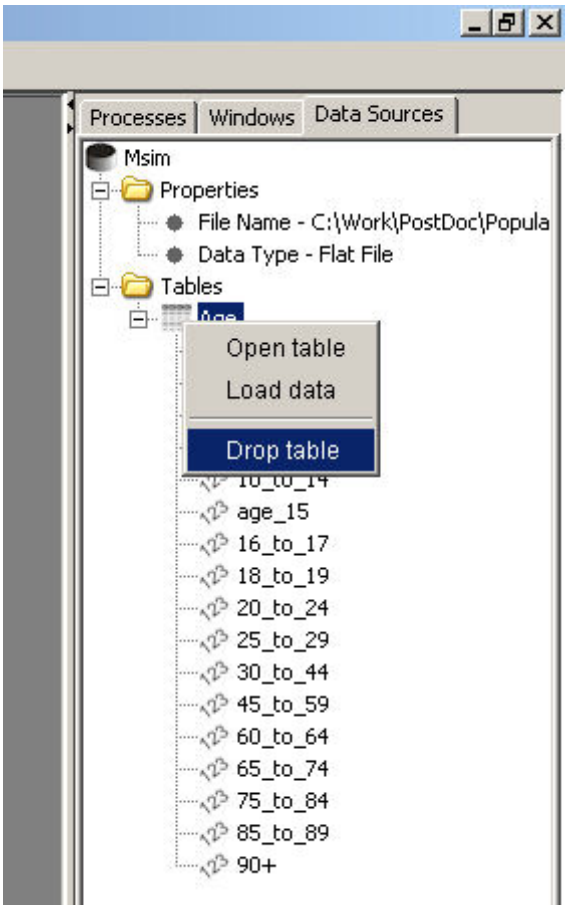


Figure 21: Un-registering a table

A dialog box asking if you wish to permanently delete Age appears, Figure 21. Click yes to remove the registration of the table; this will NOT remove the underlying data file. Click no to cancel the operation and retain the table registration.

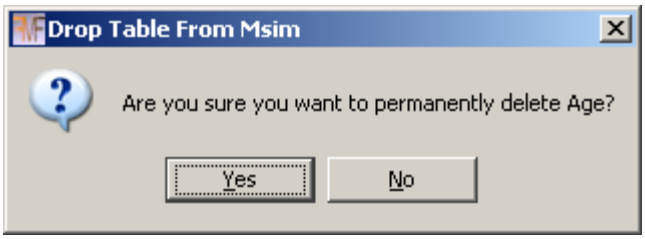


Figure 22: Un-register table confimation dialog

4.3 Loading data and opening tables

To load data into a table without displaying the data, right click on the table and select the 'Load data' option. The table will have a green tick through it to show that data has been cached in memory successfully, shown below in Figure 22.

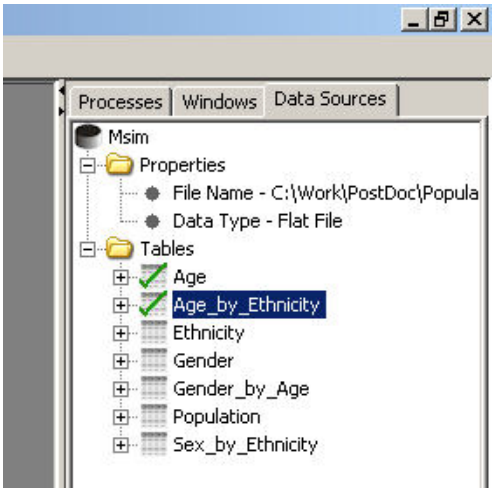


Figure 23: Data cached in sample tables

Right clicking on the table when it has data loaded shows the context menu below, Figure 23, giving an option to 'Clear cache'. Selecting this option will remove the data from memory and return the table to a normal state.

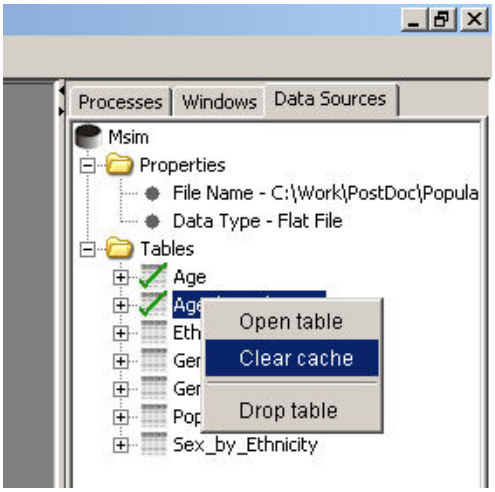


Figure 24: Table context menu

All context menus produced from right clicking on a table have an option to 'Open table'. Selecting this option will load data into memory and open a window containing the data in the action area of the screen, Figure 24. Opening a table can also be achieved by left clicking on the table and dragging it into the action area while holding down the left mouse button.

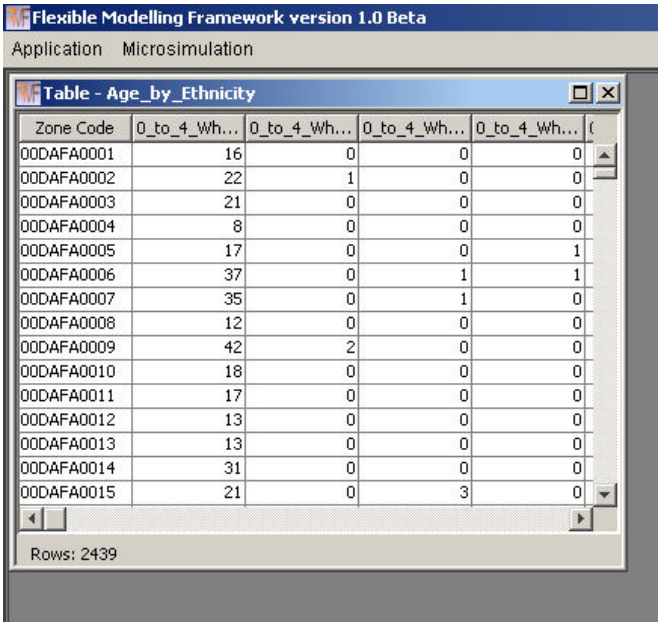


Figure 25: Table window opened in the action area of the screen

If you select to 'Clear cache' of a table that is open in the action area it is automatically closed. If you have several tables with data loaded you can right click on the table folder and select 'Clear all table caches'. This will remove the data from memory for all tables in the table folder.

4.4 What happens if file names or directories change between loads

4.4.1 Change in data source directory

If the path or directory name of a data source changes between loads then the verification process for the data source will fail and it will be displayed as shown below in Figure 26.

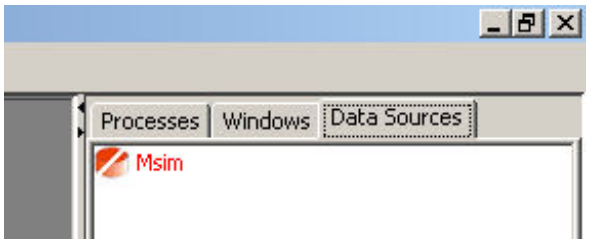


Figure 26: Invalid data source

The data source information can still be expanded to allow the user to find the location that is being searched to find the data source and enable the user to revert the name or location back to that expected by the application. If the directory is returned to the original name and location and contains all of the tables previously registered then the application will automatically validate the data source and tables the next time it restarts, currently restarting the application is the only way to restore the link to the data in these circumstances without re-registering the data source and all tables.

If the data source directory or path is changed externally while the application is open, tables will still be able to be opened but they will be empty and when closed or saved the current defined path in the application will be created with an empty file representing the table closed / saved.

4.4.2 Change in file name

If a file name (.csv file containing data) changes whether it is before or after the application is opened, the table will still appear in the data source. It will behave as described above, if opened it will contain no data and when closed or saved a new file will be created using the table name in the data source directory.

4.4.3 Change in file structure

If the file structure of the file you are trying to load does not match that registered in the application the statement 'The file you are trying to load <<table/field name>> is not registered correctly. fields = <<count of fields>> registration = <<count of registered fields>>' will appear in the reporting area of the screen. To change the registration of the table you will need to right click on the table and unregister it and then re-register it with the application or replace the underlying data file with one of the correct field format.

If text is found in a field defined as a numeric field the following statement will be displayed in the reporting area of the screen -'The input file format for table Age is incorrect. Possible cause text characters in fields defined as number fields.' The cause would need to be investigated and the data either corrected or the file unregistered and re-registered with the correct field types defined.

4.4.4 How *Null* is treated

Null values in underlying data will be allocated either an empty string or a 0 depending on whether the field is defined as a text or numeric type respectively.

5 Windows and Processes

5.1 The windows tab

The middle tab on the tabbed pain at the right hand side of the screen is called 'Windows'. This tab shows a list of all open windows in the action area of the screen. The window that is currently active is highlighted in the windows tab as shown in Figure 27 below, 'Table – Ethnicity' is the name of the window that is on top and active in the activity area and is also highlighted in the 'Windows' tab. Clicking on a different window in the action area will change the currently active window, alternatively selecting the window in the 'Windows' tab will also make the selected window active and move it to the front of the display if it is hidden by any other windows. If more than one window is opened with the same name a number is appended to the end of the window title in the form [1] as shown in Figure 27 below for the window 'Microsimulation', 'Microsimulation – [1]' and 'Microsimulation – [2]'. The window title, shown in the top banner of the window in the activity area, will always match the title displayed in the 'Windows' tab.

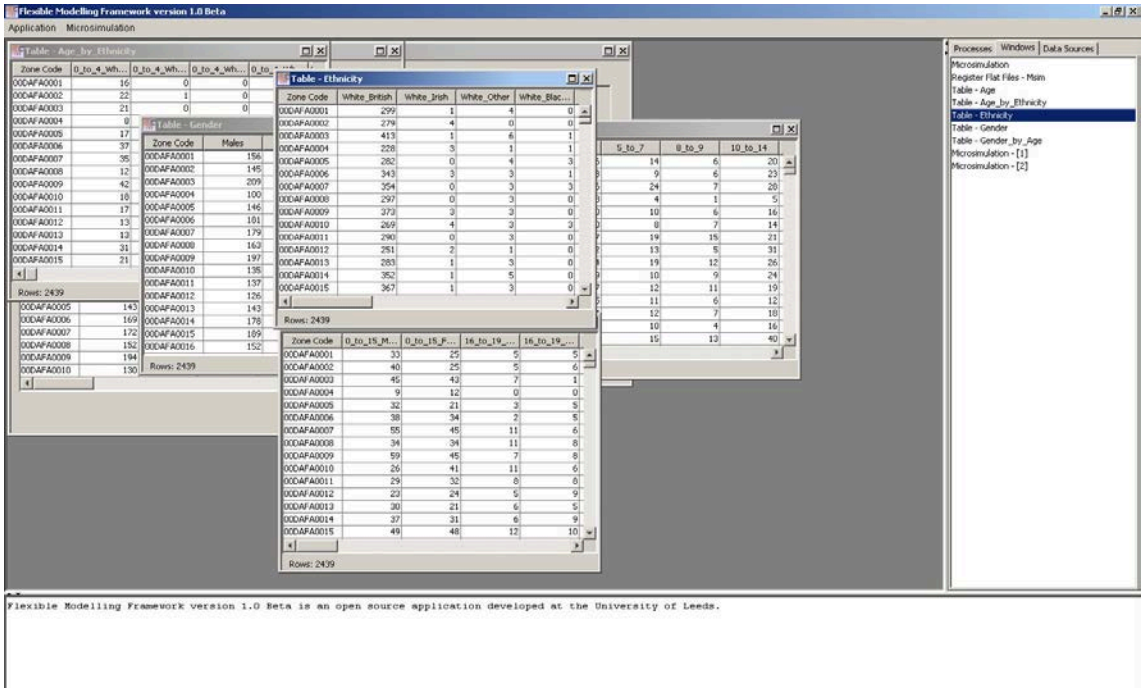


Figure 27: Interaction between windows tab and open windows in the action area

5.2 The processes tab

When the application is 'working' processes that are currently being executed are displayed in the 'Processes' tab as shown in Figure 28 below. The blue bar is the progress bar, which increments as a process moves towards completion. In the progress bar is displayed the process name and beneath each progress bar is a related cancel button. When a process is live the cancel button will be enabled but once the process reaches the end the button will be disabled and the progress bar and button will then be removed from the 'Process' tab. Clicking the cancel button indicates to the application that the particular process is no longer required, it is worth noting that the process may take some time (seconds through to a few minutes depending on the complexity of the work being undertaken) to register the cancelation, exit and remove the process from the tab. If an error occurs during the execution of a process, the text in the process bar will report the error and a message will normally be displayed in the reporting area and a detailed stack trace recorded in the log file in the root folder where the application has been deployed.

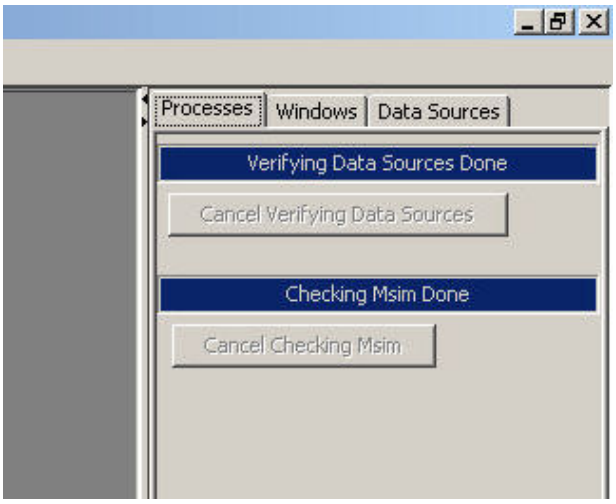


Figure 28: Progress bars for processes in the processes tab

6 Microsimulation

6.1 What is the technique and what does it do?

Static microsimulation is the process of creating a detailed population from aggregate data reproducing the relative known distributions within the population as accurately as possible. Several approaches can be taken and are discussed in the wider literature, for one example see Harland *et al.* (2012). The implementation discussed here uses a combinatorial optimisation approach called Simulated Annealing (see further reading for suggested explanatory texts).

6.2 Data format

To create the synthetic population the following are required:

- a population to sample from, hereon referred to as the sample population.
- constraint tables containing information about the aggregate counts for each type of person in each zone hereon referred to as constraints.
- To evaluate how well the population has been created verification tables are required. This is an optional but recommended post population creation stage. Tables not used as constraints but used available to evaluate the population will be referred to as evaluation tables.

6.2.1 Structure of the sample population

The format for the sample population is to have one record for each unit (person / household) type. Each record should have a unique identifier, the P_ID field in the example below. A field for each of the desired constraints and evaluation attributes should also be present (Gender, Age, Ethnicity, Age_Gender etc… in the example below, Figure 29).

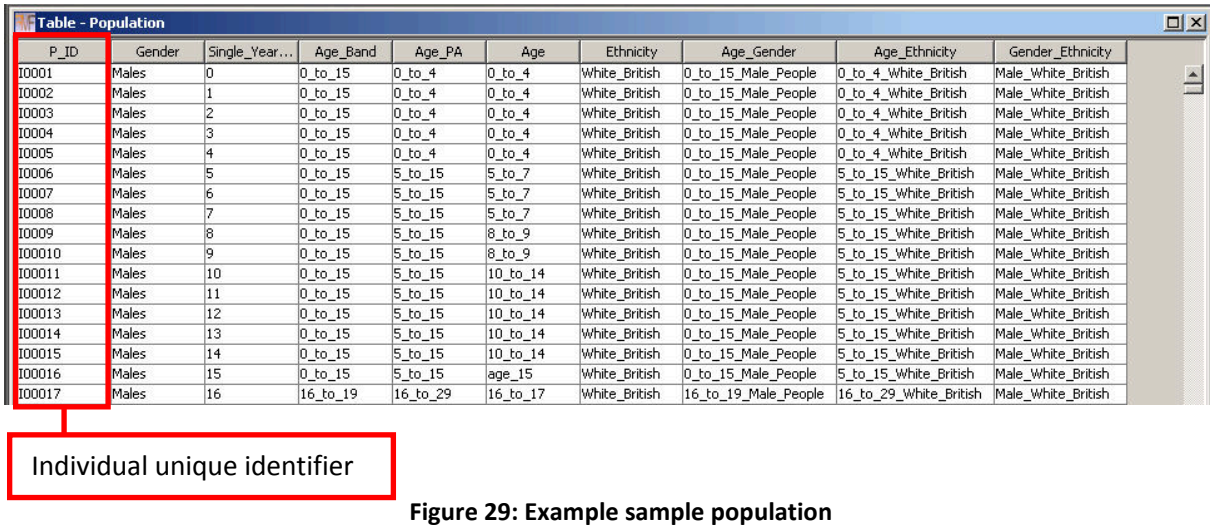


Figure 29: Example sample population

6.2.2 Structure of the constraints and evaluation tables

Constraint tables should contain a zone field and counts for each category within the attribute for each zone. All constraint tables should contain the same number of zones and have the same zone identifiers although the order of the zone records within the tables is unimportant. The structure of a constraint table is shown in Figure 30 below (gender by age). The left field in the example is the zone code identification field and the remaining fields are all categories for individuals within this constraint containing the count of individuals within each category. It is important that all of the fields containing counts for the constraints are registered as numeric fields.

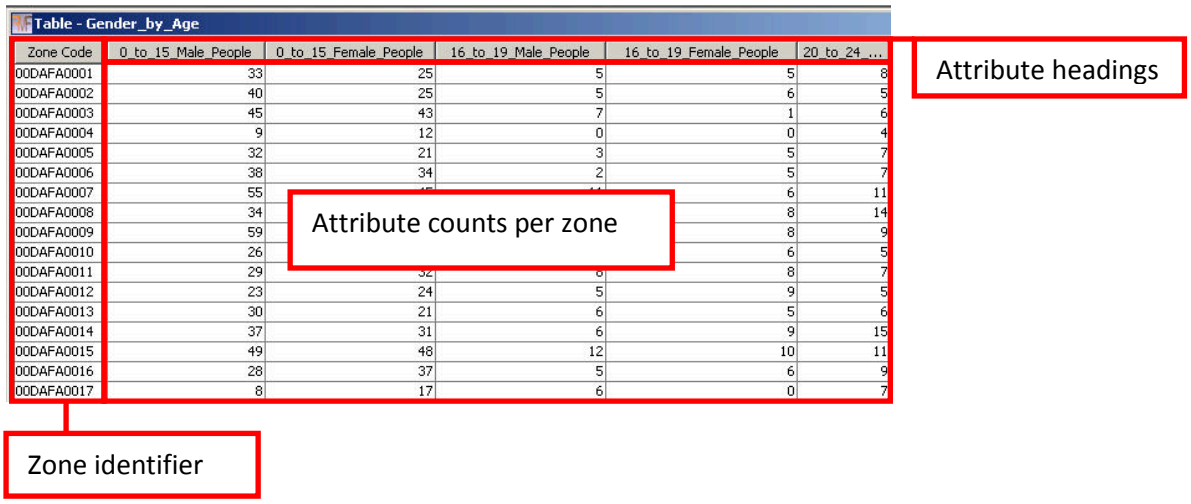


Figure 30: Constraint table example structure

It is worth noting that both constraint and evaluation tables are structured in exactly the same way, the only difference between the two tables is that evaluation tables do not necessarily need to be included as constraints during the model run.

6.2.3 How the tables relate

The sample population table should contain all of the different valid constraint category combinations such that all distributions within the data can be considered and simulated by the

microsimulation algorithm. Some combinations are obviously not valid, we would not expect to find children in an employment category for example. Each constraint should relate to one field in the sample population table and each category in the constraint that has a count of greater than 0 for any zone should be represented in the corresponding field in the sample population table. Constraints should not relate to more than one field in the sample population and sample population fields should only refer to one constraint or evaluation table.

A relationship between two categories in the sample population table and the corresponding constraint table is highlighted in Figure 31. It is worth noting that both the sample population and constraint tables contain many more records than displayed in the figure.

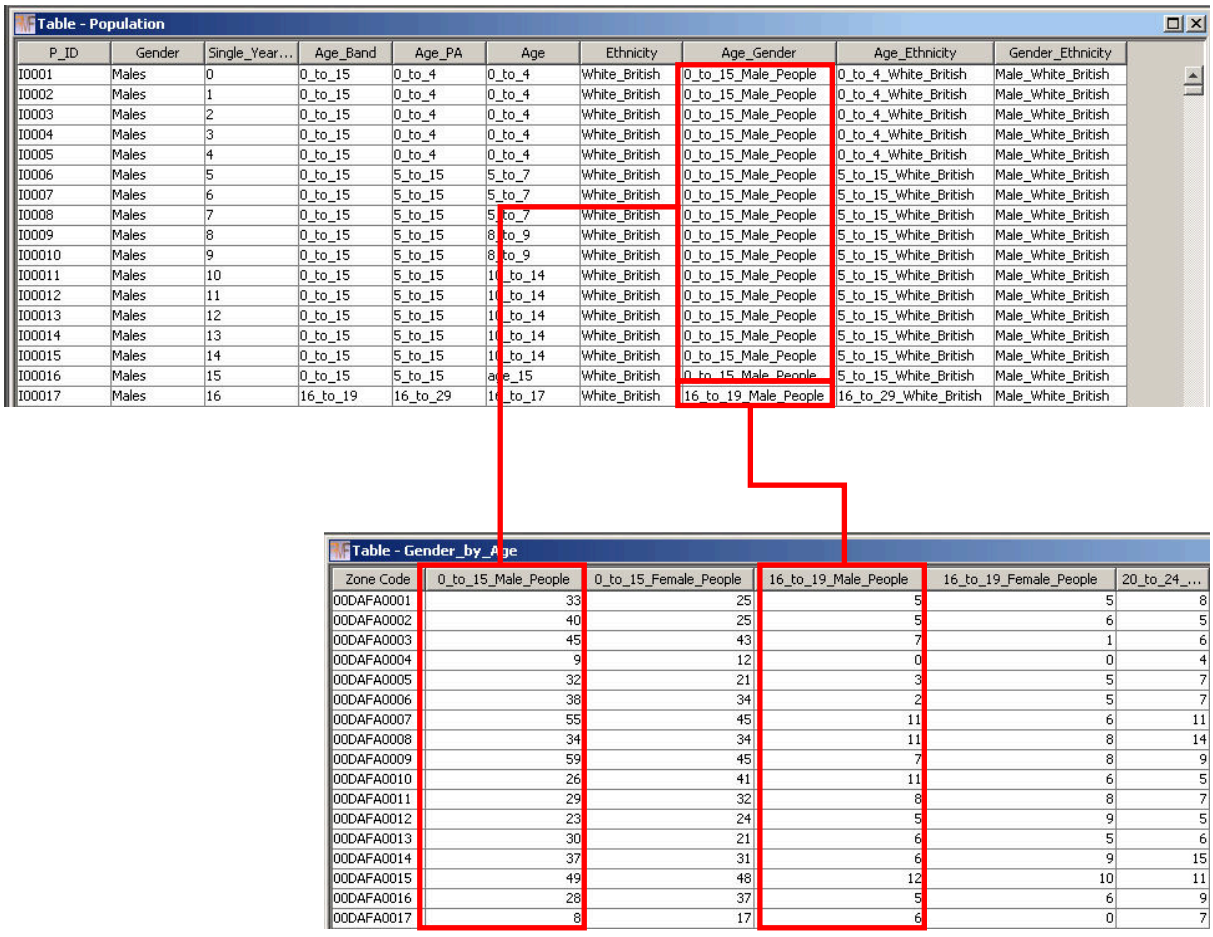


Table - Population

P_ID	Gender	Single_Year...	Age_Band	Age_PA	Age	Ethnicity	Age_Gender	Age_Ethnicity	Gender_Ethnicity
I0001	Males	0	0_to_15	0_to_4	0_to_4	White_British	0_to_15_Male_People	0_to_4_White_British	Male_White_British
I0002	Males	1	0_to_15	0_to_4	0_to_4	White_British	0_to_15_Male_People	0_to_4_White_British	Male_White_British
I0003	Males	2	0_to_15	0_to_4	0_to_4	White_British	0_to_15_Male_People	0_to_4_White_British	Male_White_British
I0004	Males	3	0_to_15	0_to_4	0_to_4	White_British	0_to_15_Male_People	0_to_4_White_British	Male_White_British
I0005	Males	4	0_to_15	0_to_4	0_to_4	White_British	0_to_15_Male_People	0_to_4_White_British	Male_White_British
I0006	Males	5	0_to_15	5_to_15	5_to_7	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0007	Males	6	0_to_15	5_to_15	5_to_7	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0008	Males	7	0_to_15	5_to_15	5_to_7	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0009	Males	8	0_to_15	5_to_15	8_to_9	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0010	Males	9	0_to_15	5_to_15	8_to_9	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0011	Males	10	0_to_15	5_to_15	10_to_14	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0012	Males	11	0_to_15	5_to_15	10_to_14	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0013	Males	12	0_to_15	5_to_15	10_to_14	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0014	Males	13	0_to_15	5_to_15	10_to_14	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0015	Males	14	0_to_15	5_to_15	10_to_14	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0016	Males	15	0_to_15	5_to_15	age_15	White_British	0_to_15_Male_People	5_to_15_White_British	Male_White_British
I0017	Males	16	16_to_19	16_to_29	16_to_17	White_British	16_to_19_Male_People	16_to_29_White_British	Male_White_British

Table - Gender_by_Age

Zone Code	0_to_15_Male_People	0_to_15_Female_People	16_to_19_Male_People	16_to_19_Female_People	20_to_24_...
00DAFA0001	33	25	5	5	8
00DAFA0002	40	25	5	6	5
00DAFA0003	45	43	7	1	6
00DAFA0004	9	12	0	0	4
00DAFA0005	32	21	3	5	7
00DAFA0006	38	34	2	5	7
00DAFA0007	55	45	11	6	11
00DAFA0008	34	34	11	8	14
00DAFA0009	59	45	7	8	9
00DAFA0010	26	41	11	6	5
00DAFA0011	29	32	8	8	7
00DAFA0012	23	24	5	9	5
00DAFA0013	30	21	6	5	6
00DAFA0014	37	31	6	9	15
00DAFA0015	49	48	12	10	11
00DAFA0016	28	37	5	6	9
00DAFA0017	8	17	6	0	7

Figure 31: Relationship between the sample population and

6.3 How the algorithm works

The algorithm is an iterative optimisation algorithm simplified from simulated annealing (Kirkpatrick *et al.* 1983) which in turn is an implementation of the Metropolis Algorithm (Metropolis *et al.* 1953). A flow diagram representation is shown in Figure 32 below. The optimisation algorithm is executed for each zone individually. It starts by randomly sampling a population of the correct number of people / households from the sample population table. The fitness of the population is then calculated, measured as the Total Absolute Error, equation 1 (Voas and Williamson, 2001), between the count of individuals in each category in the synthetic population and the expected count from the constraint tables summed across all constraint tables.

$$TAE = \sum_i \sum_j |T_{ij} - E_{ij}|$$

A change to the synthetic population is suggested, a randomly identified individual from the synthetic population is replaced by a randomly selected person from the sample population. The fitness of the synthetic population is again calculated. If the change improves the population fitness the change is automatically accepted. If the change degrades the fitness the exponential of the fitness difference (a negative number) divided by the 'temperature' (the distance travelled down the optimisation path) is compared to a randomly generated number between 0 and 1. If the randomly generated number is less than the exponential of the change calculation, the change is accepted, if it is greater the change is rejected. As the algorithm proceeds down the optimisation path and the annealing schedule reduces the likelihood of accepting a change for the worse reduces.

Once a change has either been accepted or rejected the maximum number of improvement attempts and the maximum number of improvements made are checked to see if either limit for these progress counters has been reached. If the limits have not been reached the algorithm suggests another random change and continues processing from this stage looping back to the change suggestion stage until one of the limits is reached or a perfect fit for the fitness calculation is achieved.

When one of the limits is reached the 'temperature' is reduced by multiplying it by the annealing factor (this multiplier is set by the user and controls how quickly the 'temperature' threshold will reduce and therefore how quickly the algorithm will move towards a solution). Once the 'temperature has been adjusted the count of annealing stages (the large outer loop) is checked to see if the maximum number of stages has been reached. If there are more stages left, the counters for maximum number of attempts and maximum number of improvements are reset and the algorithm loops back to suggest another random change. If there are no more stages left the algorithm has reached the end.

Figure 32: The Simulated Annealing algorithm

There are three ways the algorithm can reach the end. 1) when all iterations have been exhausted and an optimal solution has not been found. 2) if when the fitness is tested after a change, the fitness statistic represents a perfect fit (0 for the Total Absolute Error statistic) the algorithm will break straight to the end, no matter what stage the different counters and limits are at. 3) if the algorithm either makes a complete circuit of the outer 'annealing' loop without taking an improving step (this is difficult to do if a backwards step is taken) or 10 outer loop iterations are completed without the fitness statistic changing the algorithm will exit. The latter conditions have been

included to allow the algorithm to exit early when it has reached a point where no further progress is made and a perfect solution has not been reached.

6.4 What all the bits of the screen mean

Figure 33 below shows the Microsimulation configuration screen. Each area of the screen is highlighted and explained in more detail below. All of the controls on the screen have help text associated with them which appear if the mouse is hovered over the item.

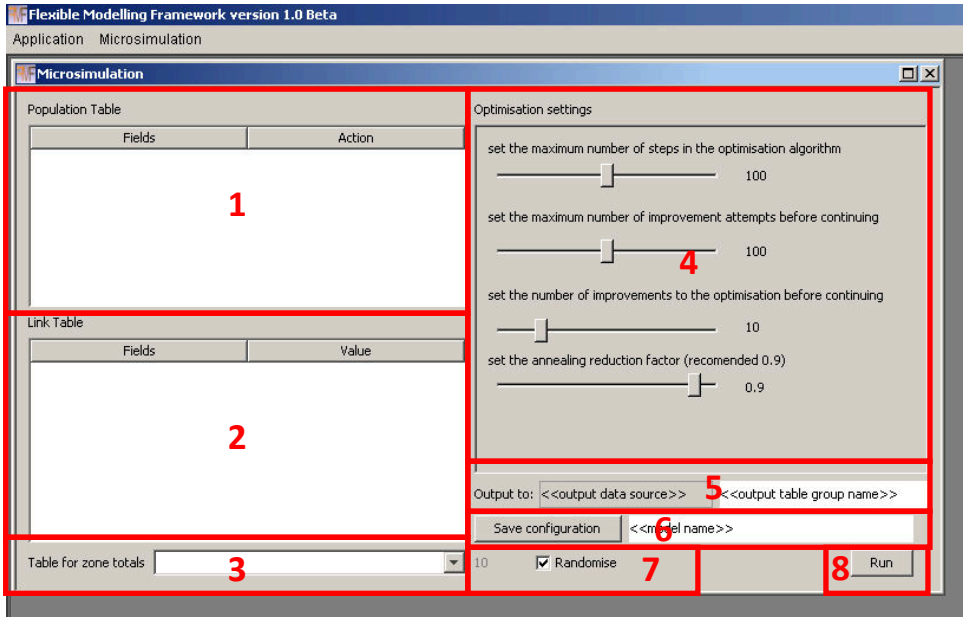


Figure 33: Microsimulation configuration screen

1. Population table area is where the sample population table for creating the synthetic population is configured.

2. Link table area is where each of the constraint tables is configured and the direct relationship between each attribute in the sample population table and the fields in the constraint (or link) tables are defined.

3. Drop down box for identifying the constraint table to be used to calculate the over population in each zone. If all constraints sum to the same value for each zone (the ideal situation) this is not critical to set and can be left as the first defined constraint. In some cases the population is most reliable in one table and is less reliable in other tables, for example the 2001 UK Census has disclosure control applied to tables which mean that not all constraint tables will add up to the expected total. Pre-processing could be undertaken to resolve this situation or alternatively, this drop down box can be used to select the most likely constraint to contain the correct population counts for each zone.

4. Optimisation control area. Each slider is used to control an aspect of the simulated annealing algorithm. Figure 34 below demonstrates how the adjustment sliders relate to the flow diagram introduced earlier. Increasing the values for each of the limits will mean extends the algorithms potential to cover more search area. Increasing the annealing

reduction factor reduces the amount that the 'temperature' decreases on each major iteration. This increases the search space covered by the algorithm, as it takes longer for it

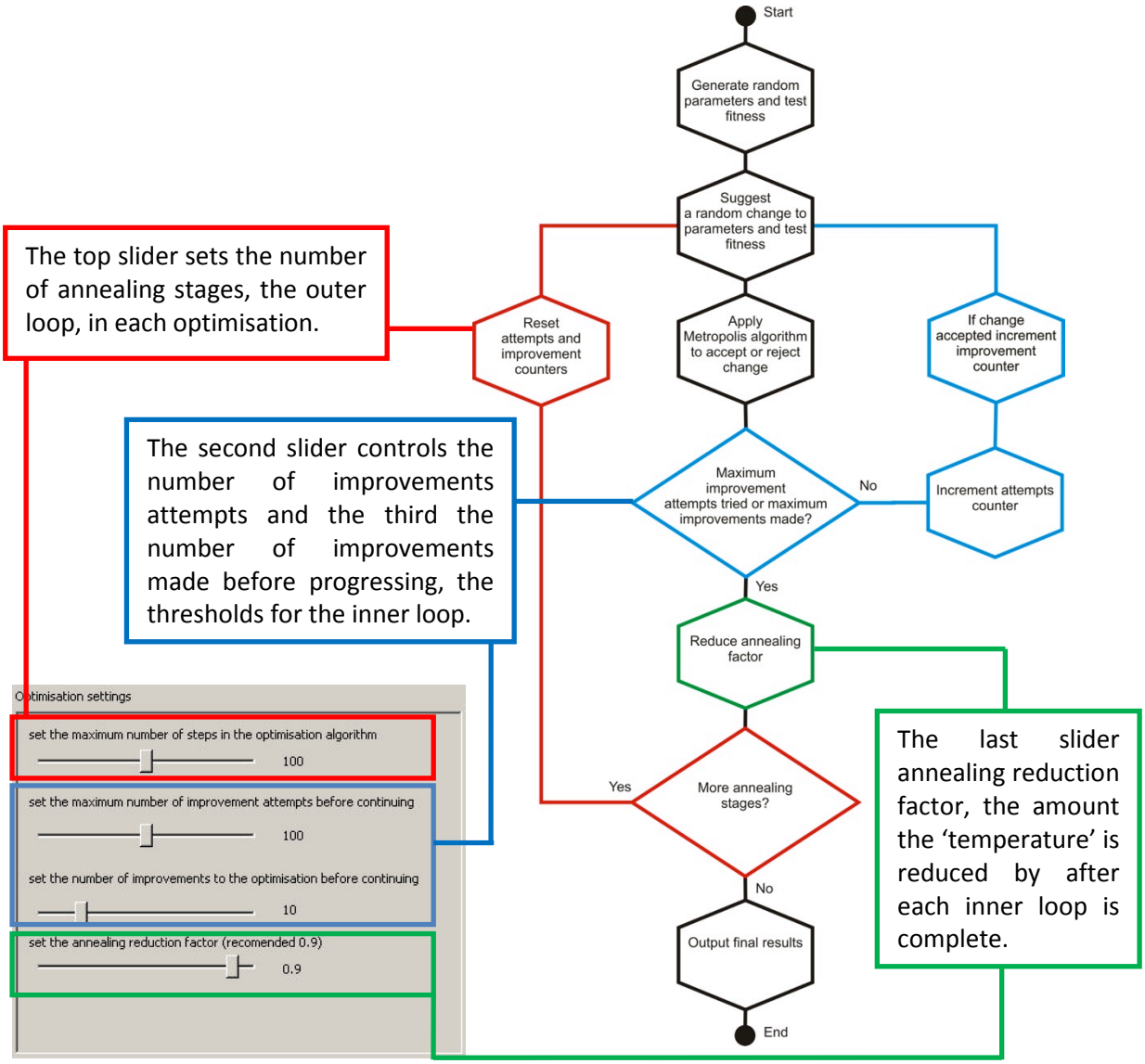


Figure 34: Relationship of Simulated Annealing algorithm and configuration controls

to move towards accepting improving steps only and settle towards a solution.

5. Output area is the registered data source where the output tables will be stored and the prefix that will be used to identify the tables created by this model run.

6. Save configuration provides the opportunity to name the configuration and save it for later use or edit.

7. Randomise settings are where the user can specify the model to be run in random mode or provide a seed to be used. Specifying a seed to be used ensures the results are reproducible

so long as the same input data and random seed are used with the same version of the algorithm.

8. The run button will start the microsimulation model.

6.5 Setting up a model configuration

6.5.1 Creating links between the sample population and constraint tables

To begin creating links between the sample population and the constraint tables, first locate the sample population table in the data sources, left click to highlight it and while keeping the left mouse button pressed drag it to the population area of the microsimulation screen and release the mouse button. The fields from the population table will now be displayed in the 'Fields' column in the population table area of the screen and the label above will display the name of the table, in this

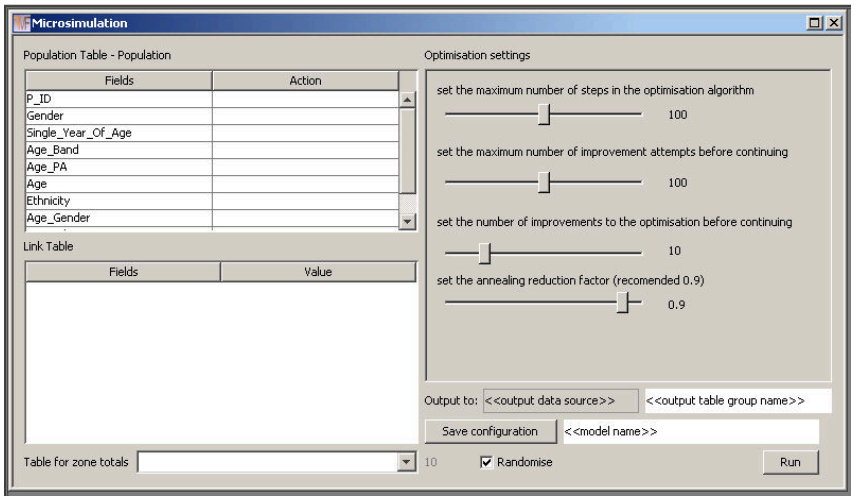


Figure 35: Population table displayed

example, Figure 35, 'Population Table – Population' because the table used here is called Population.

We now need to tell the model which field in the sample population table is the unique identifier. Locate the field, in this example it is P_ID. First left click on the row with the field name in to highlight it and then right click to show the context menu as shown in Figure 36 below. Select 'Set as ID field' and the text 'Pop id field' should appear in the 'Action' column as shown in Figure 37.

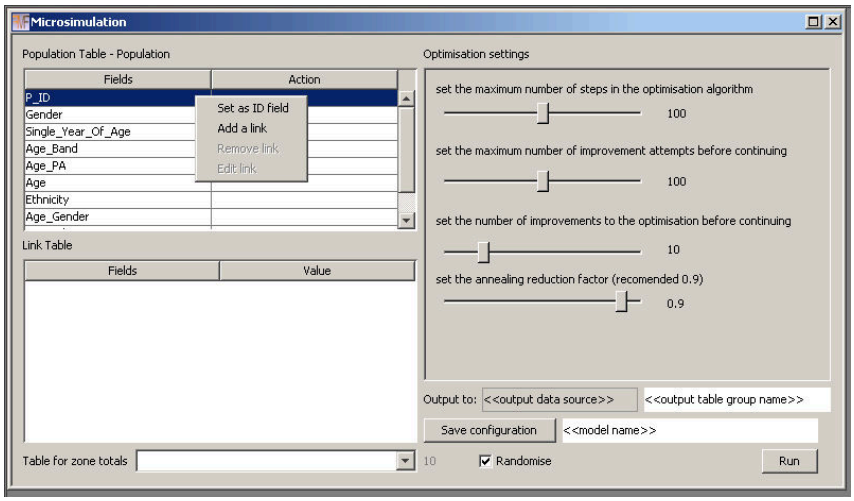


Figure 36: Selecting the unique identifier

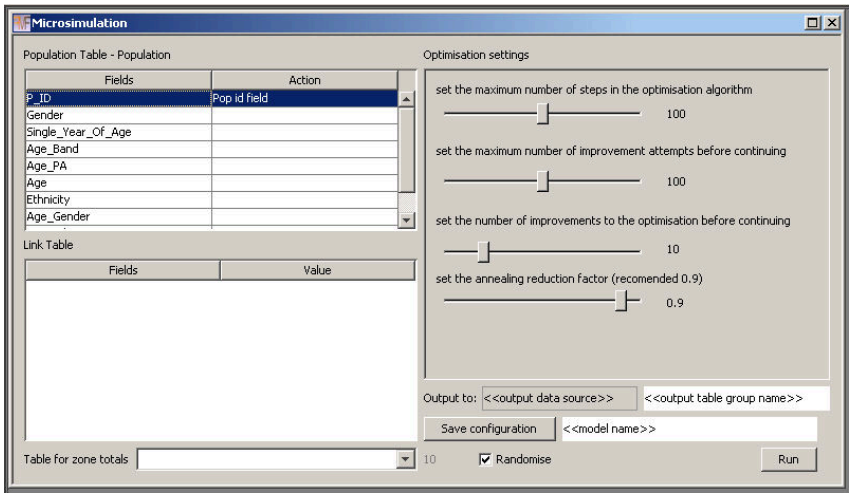


Figure 37: Pop id field displayed in action

To add a link we locate the field we wish to constrain in the population table as we did above to set the ID field, left click to highlight it and then right click to show the context menu again. This time we select 'Add a link' from the context menu. The example in Figure 38 shows a link being added to

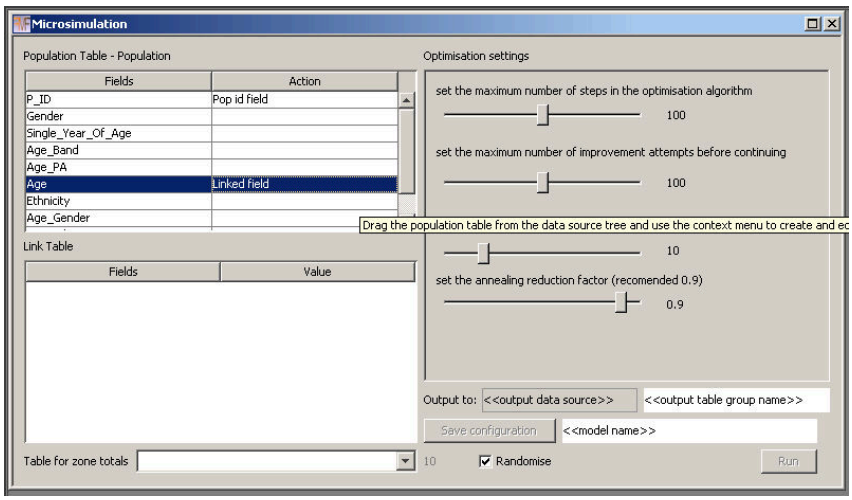


Figure 38: Link added to population table

the 'Age' field.

Notice that the text 'Linked field' is now displayed in the Action column and the 'Save configuration' and 'Run' buttons are disabled, additionally the context menu accessed by right clicking in the population area is no longer available. This is because we have only half finished creating the link. We have identified the field in the sample population that we want to link to a constraint table but we have not yet identified the constraint table or the attribute category to constraint field relationships. To do this locate the constraint table to be associated with this population field in the data sources area and left click on it, hold down the mouse button and drag the table into the 'Link Table' area. The 'Fields' and 'Value' columns should become populated as shown in Figure 39 below.

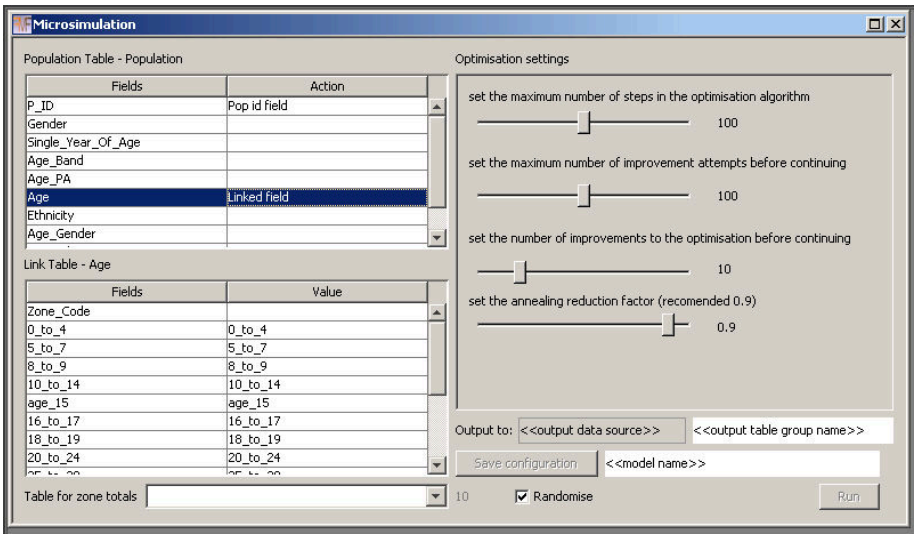


Figure 40: Link table area populated with field and values from constraint and sample population

The label at the top of the area now shows the table name, in this example the label reads 'Link Table – Age' because the age constraint is called, unsurprisingly, Age. The 'Fields' column contains all of the fields from the constraint table. The 'Value' column contains all of the attribute values found in the associated field in the sample population table. The application will attempt to match the field name to attribute where the two match. It is worth making sure that this has been done for all of the relationships between fields and values that you want to make. The top row in the constraint table is empty in the 'Value' side of the display because no match was found for this field, this is expected as this is the zone identification field. We need to tell the model that this field is the zone identification field. We do this by left clicking in the empty 'Value' cell to reveal a drop down of all the possible attributes values from the sample population field, with the option for 'Zone ID' at the top, Figure 40, select this option.

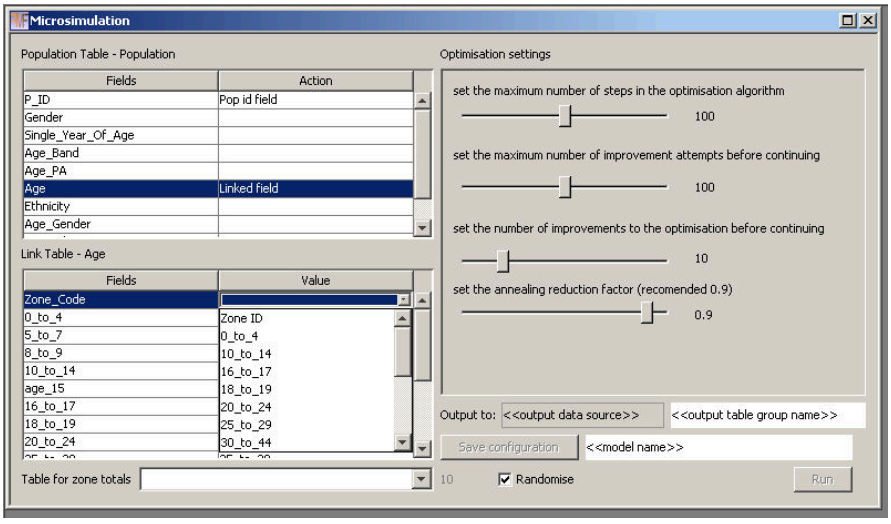


Figure 39:Manualling identifying the zone

The link is now ready to be saved. All of the relationships are defined and the zone identification field has been determined. To save the link right click in the link table area to reveal the context menu shown in Figure 41, select 'Save link'. The link table area resets, the 'Save configuration' and

'Run' buttons become enabled again and the context menu in the population table area is again accessible. You will also notice that the name of the link table has been added to the drop down box at the bottom of the screen 'Table for zone totals', we will come to this later. The 'Remove row' and 'Add row' options from the context menu allow for the correction of any mistakes made during the specification of relationships between the sample population table and the constraint. It is worth

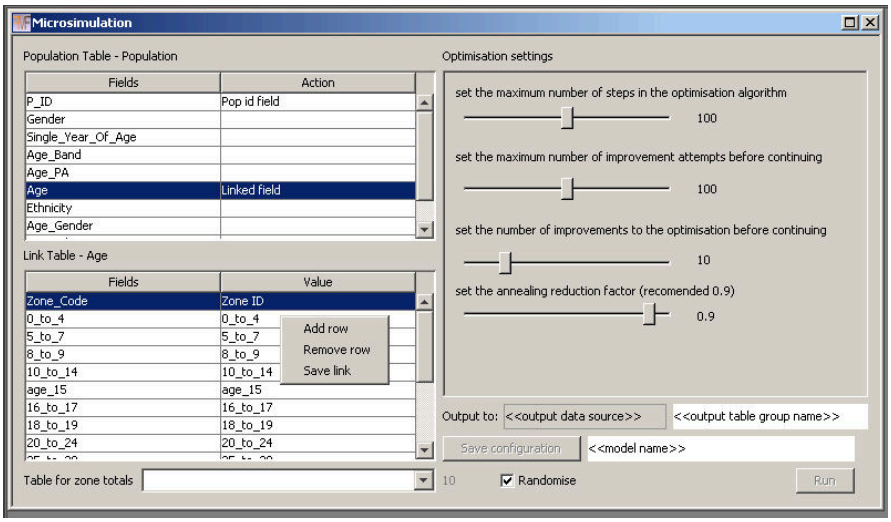


Figure 41: Saving the link

noting that any fields from the constraint table not included in relationships with the sample population will not be included in any model run.

Once a link has been saved, right clicking on it in the population table area will present the same context menu we used to start and add the link above, this time the 'Set ID field' and 'Add a link' options are disabled and the 'Remove link' and 'Edit link' options are enabled, Figure 42. Selecting the 'Edit link' option will reload the relationships and table information into the link table area allowing adjustments to the relationships to be made. Selecting the 'Remove link' option will delete the link from the configuration altogether.

One link can be defined for each of the fields in the sample population table, except for the field reserved as the unique identifier, but links do not have to be defined on all fields in the table. Only one constraint table can be linked to a field. If the field names in the constraint table and the attribute values in the sample population table are not the same, suggestions will not be made by the application and the user needs to manually create the relationships as shown in Figure 43 below for the Gender constraint.

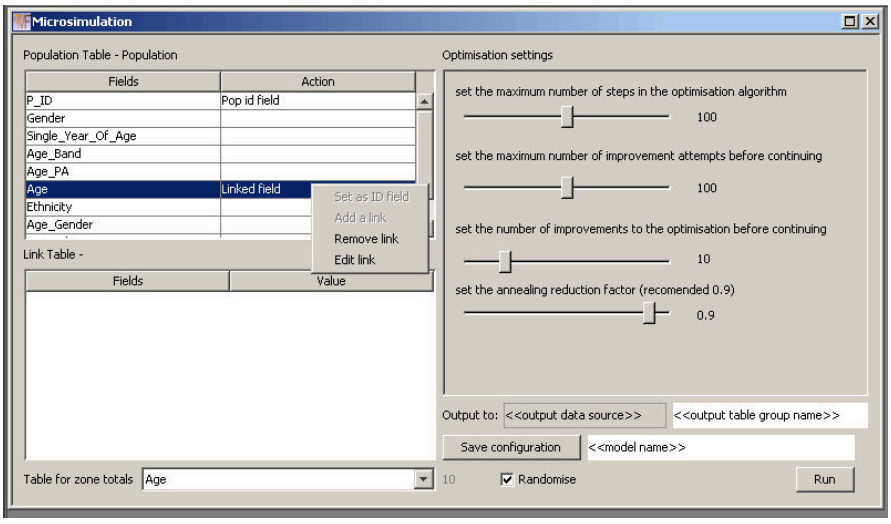
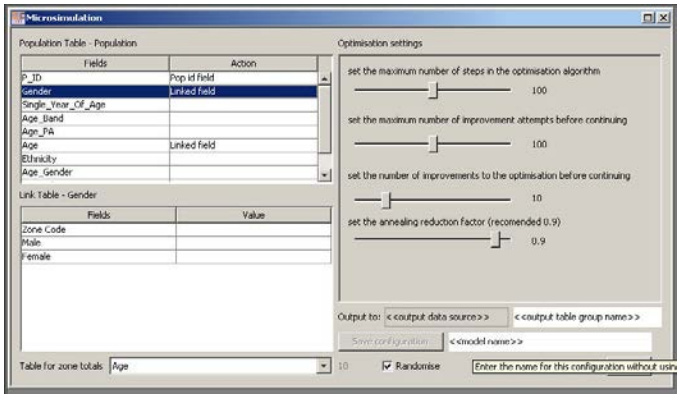
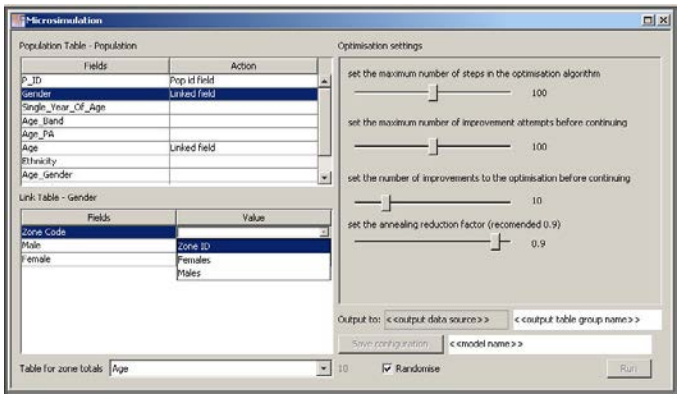


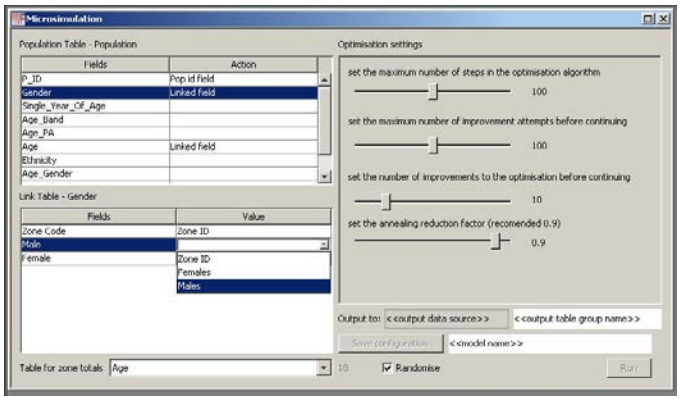
Figure 42: Link context menu in sample population table area



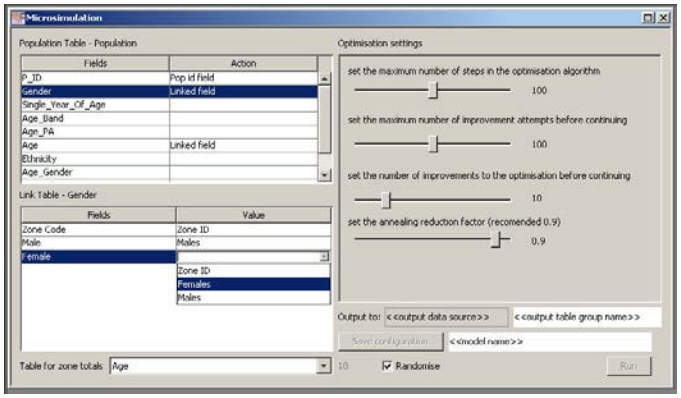
Add the constraint table to the link table area but no field – value relationships automatically identified.



Identify the field containing the zone unique identifier in the constraint table



Manually identify the ‘Male’ field to ‘Males’ attribute relationship



Manually identify the ‘Female’ field to ‘Females’ attribute relationship

Figure 43: Manually configuring a link

The process of linking the sample population fields to the corresponding constraint tables is repeated until all of the required constraints have been added. In the example used here, three univariate constraints have been added – Gender, Ethnicity and Age – and three cross tabulated constraints have been added – Age by Gender, Age by Ethnicity and Gender by Age.

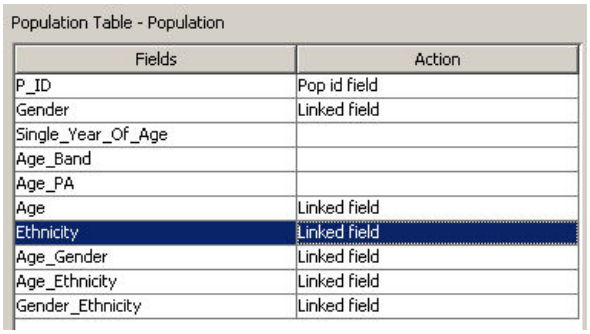


Figure 44: All constraints configured

6.5.2 Selecting the constraint to calculate total population values from

Once all of the links are added the constraint which is most likely to hold the actual number of population units for each zone is selected in the drop down box 'Table for zone totals' at the bottom of the screen. In this example we are using the Gender constraint because it has the smallest number of categories (male and female) and should therefore be subject to the least amount of adjustment for disclosure control, Figure 45.

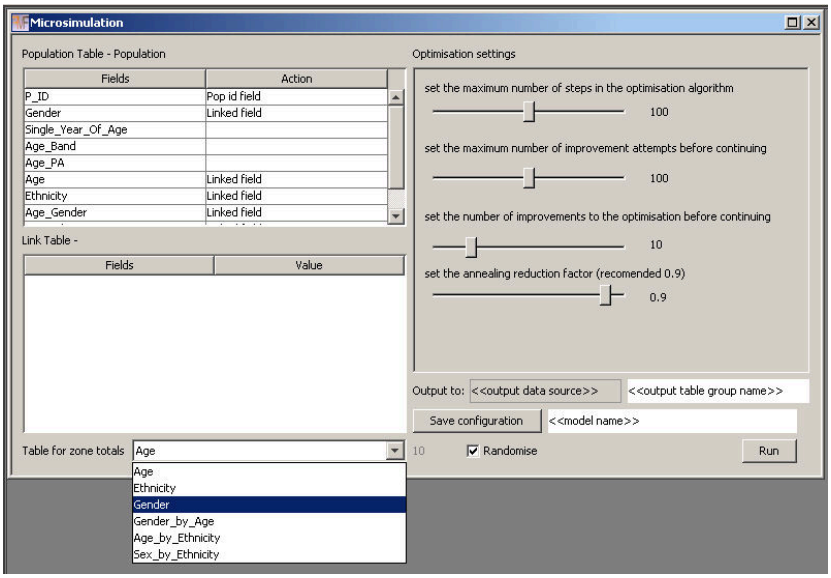


Figure 45: Selecting most trusted constraint

6.5.3 Adding output information

The next stage of configuration is to identify where the model outputs should be saved. First identify the data source where the tables should be saved in the data sources tab, click on it and while holding down the left mouse button drag it into the '<<output data source>>' area as shown in Figure 46 below. Next to the data source is a free text box to enter a prefix that all model outputs will use to identify them, in this case 'UG_Test'.

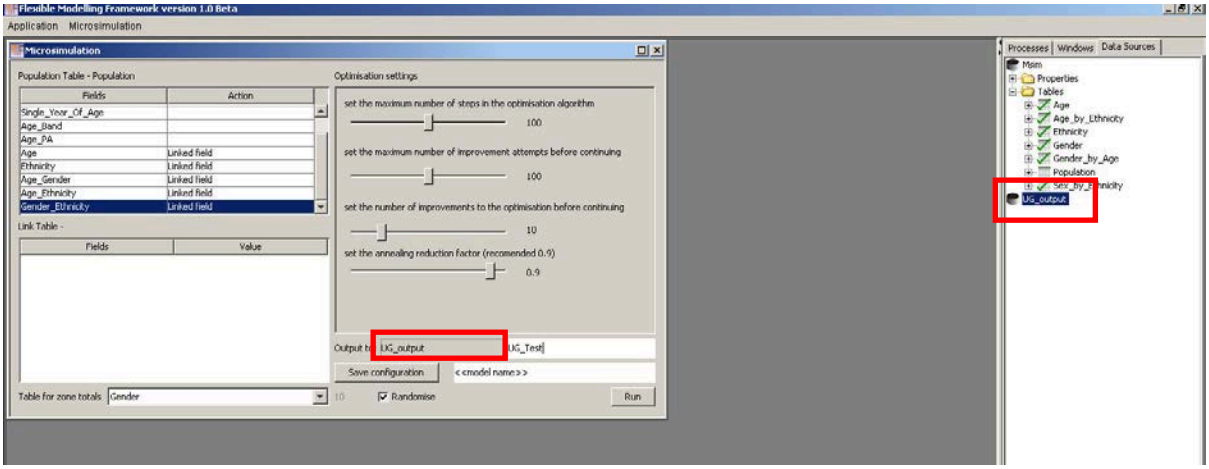


Figure 46: Setting output location

6.6 Saving, loading and deleting a model configuration

6.6.1 Save

To save a configuration, enter a name in the highlighted area of the screen in Figure 47, in this case 'User Guide Ex' and click the save button. This action will save all the configuration settings entered so far under the name specified.

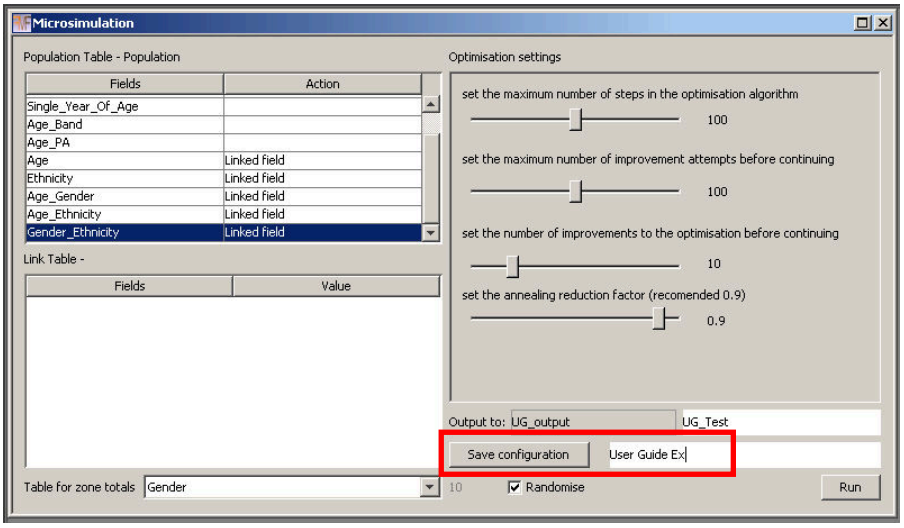


Figure 47: Saving a configuration

The name for the configuration should now appear in the expanded areas next to the option to 'Load microsimulation model' in the main 'Microsimulation' menu as shown below in Figure 48.

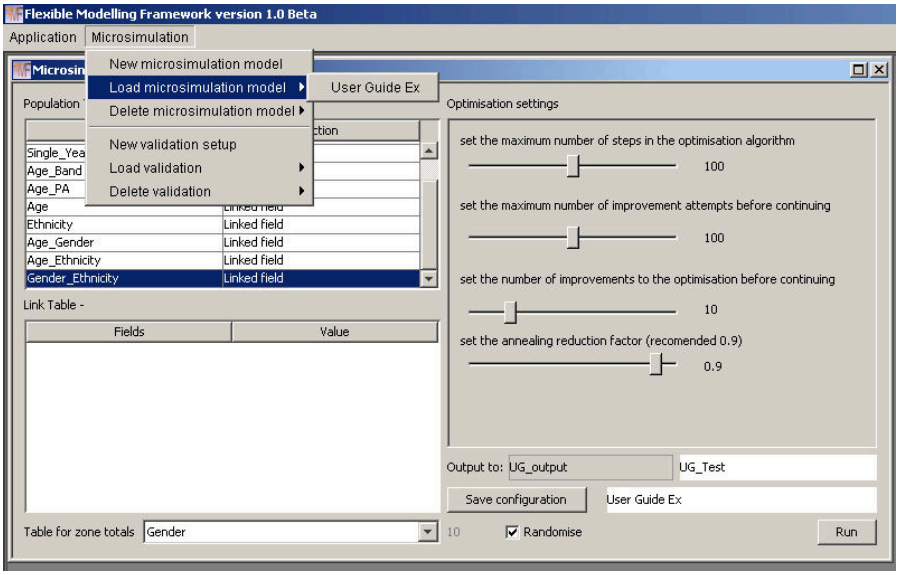


Figure 49: Saved configuration

If the configuration name already exists a dialog like the one shown in Figure 49 will be displayed. Clicking 'No' will cancel the save operation allowing the name to be changed on the main microsimulation screen before saving is attempted again. Clicking 'Yes' will overwrite the saved version of this configuration with the one that is currently displayed on screen.

Figure 48: Configuration name conflicts

6.6.2 Load

To retrieve a previously saved model configuration simply navigate to the 'Load microsimulation model' in the main 'Microsimulation' menu and hover over the option until it expands and then select the model configuration name you wish to load, Figure 50.

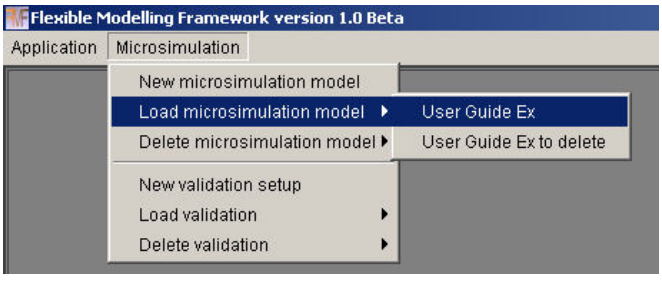


Figure 50: Loading a configuration

Clicking on the named model will load up the microsimulation screen with the options specified in the configuration as shown in Figure 51 below.

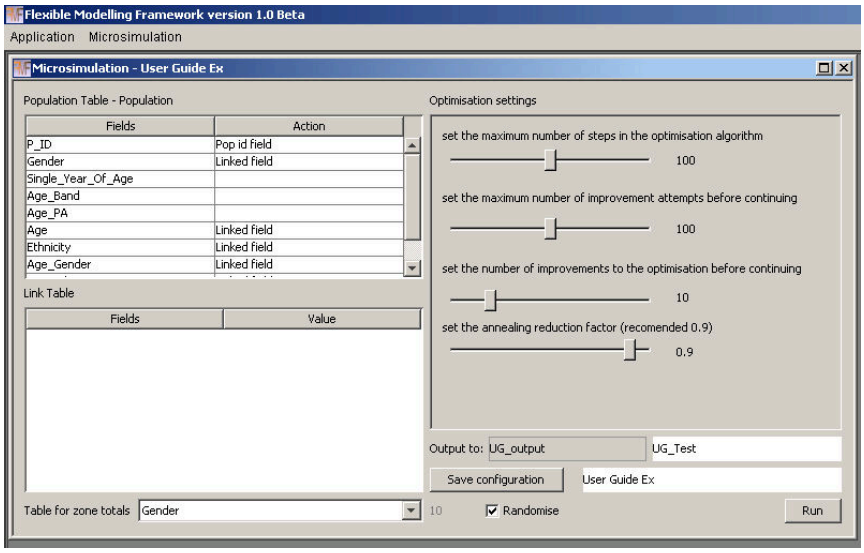


Figure 51: Reloaded configuration

Note: the optimisation settings, randomise selection status and seed settings are not stored in the configuration and need to be adjusted each time a model is run.

6.6.3 Loading a configuration where data sources or tables have changed

When a data source or table has been moved, renamed or altered you will be asked to locate the new location for the table before the configuration can be loaded as shown below. If the structure of the table has changed or field names are different you will be asked to locate the missing fields, Figure 52.

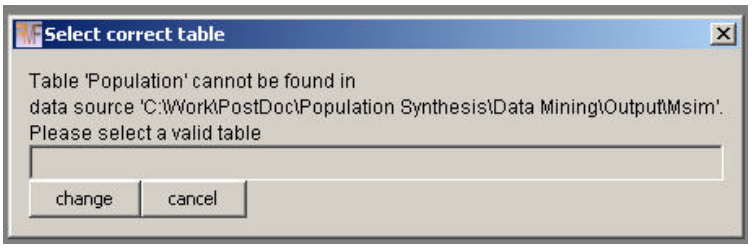
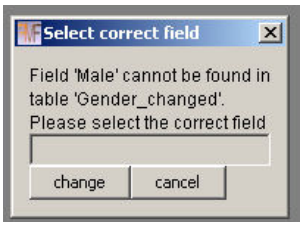


Table name changed dialog



Field name changed dialog

Figure 52: Locating missing data when reloading a configuration

Clicking cancel on either dialog will cancel the change to the location of the table or field. If the table or field is in the sample population table this will result in the load of the configuration being cancelled as a model run cannot be executed without the sample population table being present. If cancel is pressed during the search for an altered constraint table then the loading of the link associated with the constraint will be cancelled and the link will be removed from the configuration although the configuration will still be loaded. This is also the case if the values contained in the

sample population table's field cannot be matched to the fields in the constraint table as previously mapped.

To make an update to a table locate it in the data sources area left click on it and drag it into the grey box above the buttons, the name of the table will appear in the box as shown in Figure 53

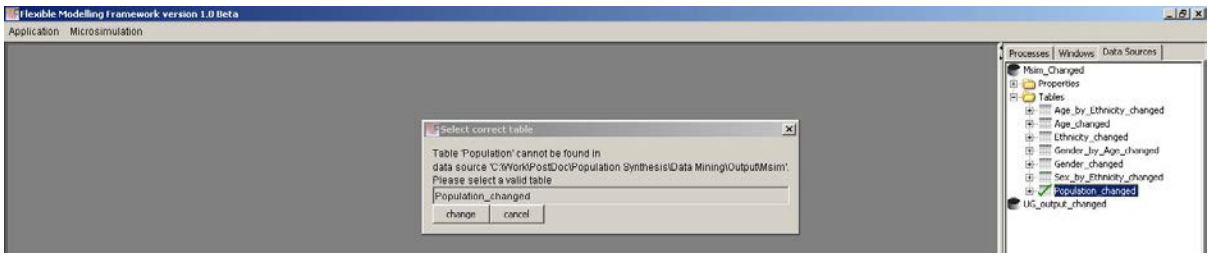


Figure 53: Adjusting the location of the sample population table

below. When you are happy you have selected the correct table click 'change'.

The fields in the table are now examined to see if they match those stored in the configuration. If they do no further action is taken. If the fields do not match the previously defined fields the field search dialog is displayed. To change the field locate the new field location from the field list by expanding the cross at the side of the table. Left click on the correct field and drag it into the grey box above the buttons, it name will appear as shown in Figure 54 below, and click 'change'.

Once all changes have been made the configuration will be loaded. If not all links can be loaded a warning will be displayed in the reporting area so that the user can take action. As shown below the warning contains information on the field from the sample population table and constraint table name forming the link that has been removed to assist in any updates that may need to be made. Reforming the link is simply a case of adding the link as described in the section 5.5.1 above.

"WARNING - Not all existing value to field pairs could be found.

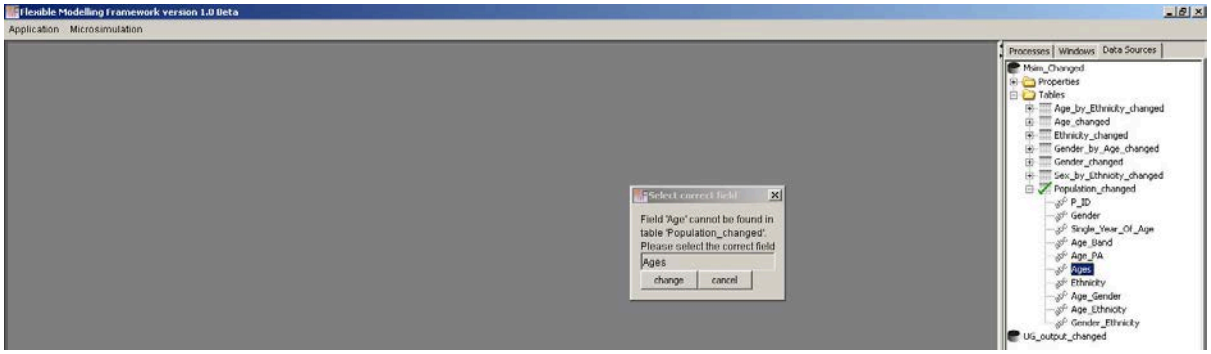


Figure 54: Altering the fields in the sample

link between Gender and table Gender_changed has been removed.
link between Age_Gender and table Gender_by_Age has been removed."

When the configuration has loaded a reminder that it has changed and that the changes have not yet been saved will be displayed.

“Changes have been made to the configuration 'User Guide Ex' save the configuration to ensure changes are available in the future.”

6.6.4 Delete

To delete a model configuration go to the ‘Delete microsimulation model’ option on the main ‘Microsimulation’ menu. Hover over the option until it expands and then click on the model name you wish to delete, Figure 55.

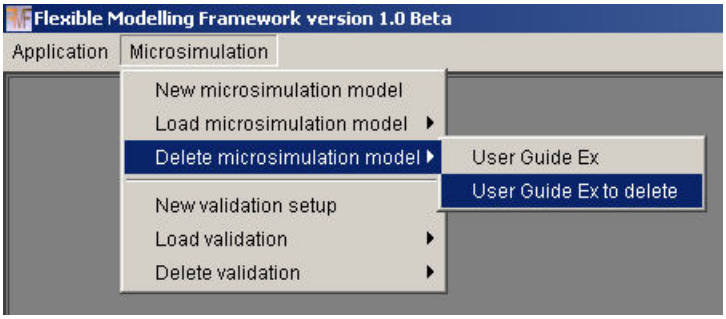


Figure 55: Deleteing a model configuration

When you have selected the model configuration to delete a confirmation dialog like the one in

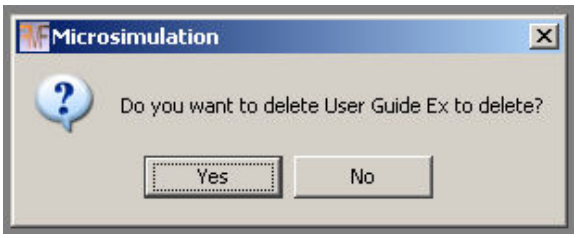


Figure 56: Delete configuration confirmation dialog

Figure 56 below will be shown.

To cancel the delete operation click ‘No’, to carry on and remove the configuration click ‘Yes’. Once deleted the model configuration will no longer be available to load or delete, as shown below.

6.7 Running a model configuration

6.7.1 Final setup and run

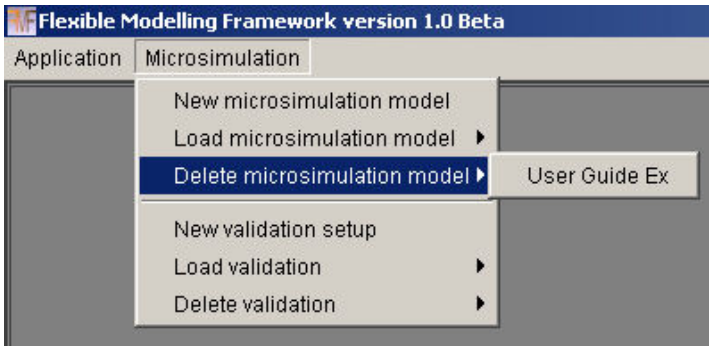


Figure 57: Model configuration deleted

To complete the final setup, select whether you would like the model to run using in a random mode or whether you would like to supply a seed to ensure the results can be recreated using the same data, seed and algorithm. The randomise options are in the bottom right of the microsimulation screen, Figure 58.

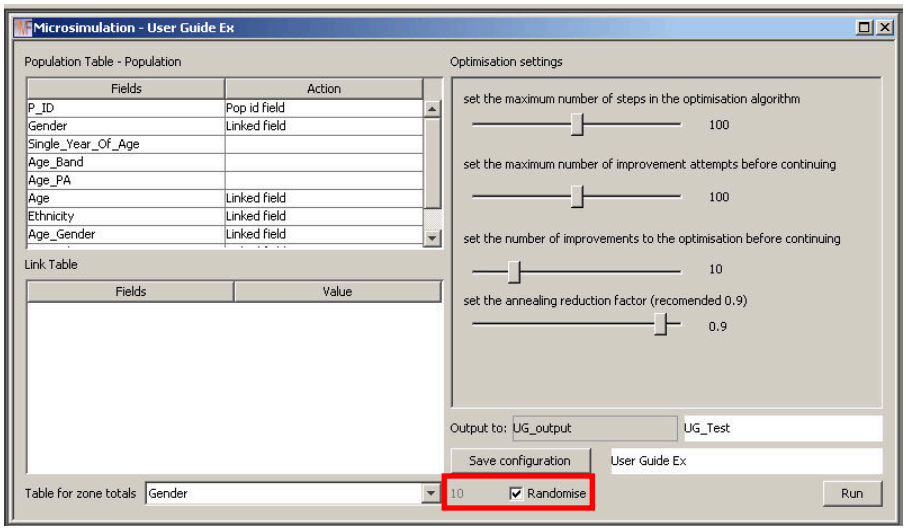


Figure 58: Selecting the random settings

If the randomise option is selected, the model will run in random mode. Unchecking the 'Randomise' option will enable the text box next to it. Any valid integer value can be entered. If you try to run a model with a non integer value as the seed a message 'The seed is not a valid Long number please correct and try again.' Will be displayed in the reporting area and the model run will be aborted.

The settings for the optimisation parameters default to 100, 100, 10, 0.9. It is recommended that you explore these with regard to your own model configuration as each simulation problem demands a different amount of work from the algorithm and these settings are relatively arbitrary.

When the run button is pressed the microsimulation screen is closed and the optimisation process begins. The 'Processes' tab should now display an overall progress bar called 'Optimising', highlighted in red in Figure 59 below. This progress bar shows the increments through the whole model run, when 100% is reached the model run is finished.

The progress bars highlighted in yellow show the progress through the optimisation for each zone in turn. The application will use as many processing cores as are available, therefore if the computer has two cores available, two zones will be optimised in parallel, if four cores are available, four zones will be optimised in parallel. Therefore, the more cores available on a computer the faster the model will complete.

To cancel the overall model run, click the 'Cancel Optimising…' button. The zones being optimised can either be allowed to complete or the cancel buttons underneath each of the progress bars can be clicked to cancel each zone optimisation process that has been left running.

When the model completes the approximate time taken is displayed in the reporting area:

“Processing Time is Weeks = 0 | days = 0 | hours = 0 | minutes = 17 | seconds = 55 | millie-seconds = 156

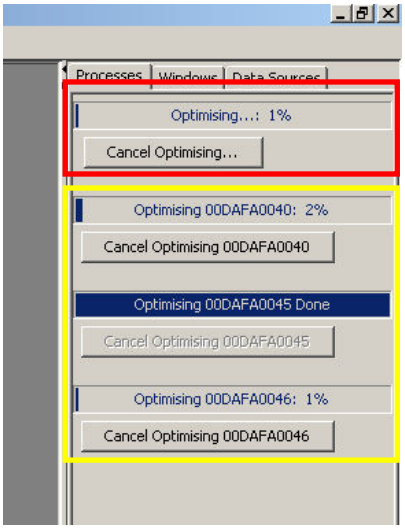


Figure 59: Model processing progress bars

Processing Time is 18 minutes.”

6.7.2 Output and interpretation

Results are saved into the data source specified in the output area of the microsimulation screen. Two tables will be created with the prefix entered on the microsimulation screen called <<prefix>>_population and <<prefix>>_stats. In the example used here the tables are UG_Test_population and UG_Test_stats, Figure 60.

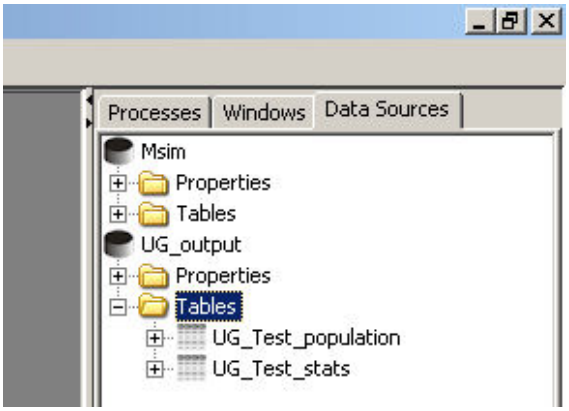


Figure 60: Output table generated

The population output table contains a list of all of the ‘cloned’ person records along with the zone identifier the person record has been cloned into. The structure of the population tables is just two fields one contains the zone code from the constraint tables and the other contains the population identifier from the sample population table, Figure 61. Using an external database the original sample population table can be joined back to the output population table using the unique population identifier (in this case P_ID) with the location for each synthesised person being identified by the zone code (in this case ZoneID).

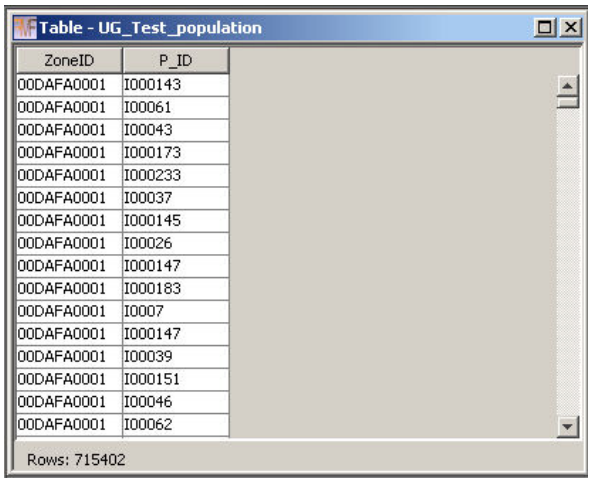


Figure 62: Structure of the output population table

The stats output table contains the fitness level recorded at each major iteration of the optimisation algorithm for each zone. It therefore has three fields, the first is the zone identification field (ZoneID), the second is the count of the iteration for which the fitness value was output (Order) and the third is the fitness, Total Absolute Error, value itself (Fit), Figure 62.

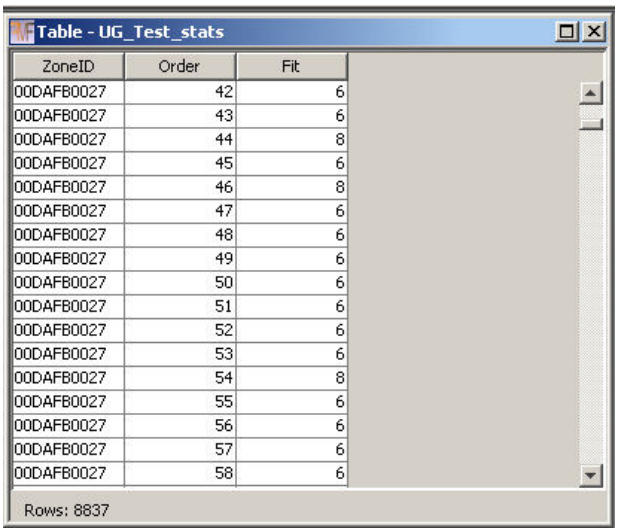


Figure 61: Structure of the output statistics table

Note: if an underlying file for the output table being created already exists but is not registered in the application, the values are appended into the file. If the file exists and is registered in the application it is removed and replaced. This is behaviour by design, files not registered and visible in the application are not deleted. If a file is registered and visible in the application it is assumed that the user would like to overwrite the information.

6.7.3 Evaluating the model

It is possible to evaluate the model fit. The options for doing this are present in the lower half of the 'Microsimulation' menu, Figure 63. The 'Load validation' and 'Delete Validation' options work in the same way as the 'Load microsimulation model' and 'Delete microsimulation model' options explained above. Once configurations have been created they are accessible through the menu areas.

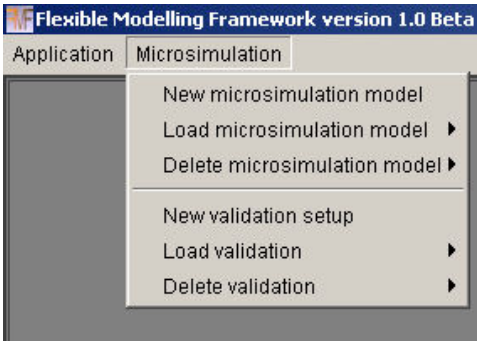


Figure 64: Evaluation menu below the microsimulation options

To create a new validation configuration click on the 'New validation setup' shown in Figure 63 and the screen show in Figure 64 will open. The screen is very similar to the main microsimulation screen and entering the population and creating links is done in exactly the same way as described in section 5.5. You are not limited to the same links as used in the microsimulation, if other information is available to be checked these data can be included here as evaluation tables.

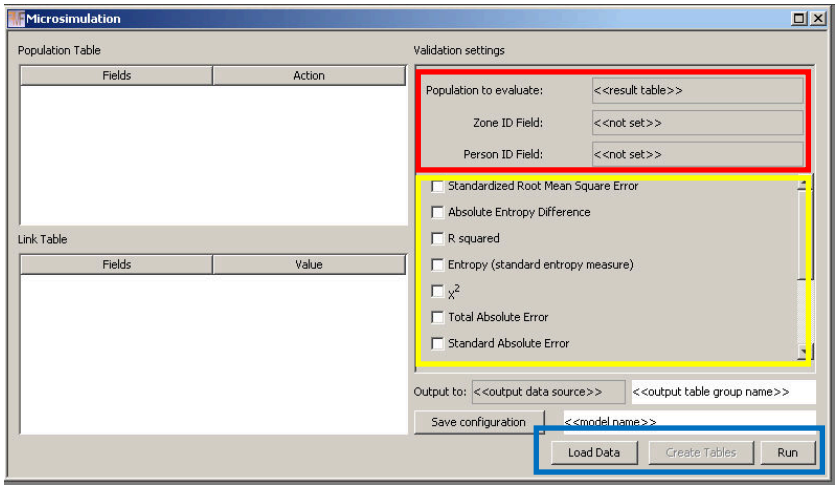


Figure 63: Evaluation window

There are three main differences to the main microsimulation screen:

1. The area at the top of the 'Validation settings' pane highlighted in red above. The top box in this area is where you enter the table name for the output population created by the model (in the example UG_Test_population), the second box is for the zone identification field (in the example ZoneID) and the third box is for the unique person identification field (in the example P_ID).

2. The area highlighted in yellow is for the selection of fit statistics to be evaluated against, the most useful of these is the Total Absolute Error measure which is near the bottom of the list. Simply locate the statistics you wish to use and check the boxes next to them.

3. The third difference is the three buttons at the bottom of the screen instead of one. This is so that different fit statistics can be added without having to run all of the evaluation stages again, after they have been run once. The three stages have been separated purely for convenience. Re-running the 'Create Tables' option is time consuming and is not required if the model has not been re-run or the number of links in the evaluation has not been

changed since a previous creation of the summary tables. If either of these conditions have changed (links or model run) then all three stages will be executed.

- 'Load Data' ensures that all of the data used to create the tables is loaded correctly.
- 'Create Tables' only becomes enabled when data has been loaded. This option creates a summary set of tables in the same structure as the constraint / evaluation tables using aggregate counts from joining the sample population table to the output synthetic population.
- 'Run' calculates the fit statistics and creates the evaluation tables requested. If additional fit statistics are required after an initial evaluation has been run and the summary tables have already been created the process can be shortened. Simply select all of the required fit statistics, click 'Load Data' and once data loading is complete click the 'Run' button missing out the time consuming 'Create Tables' stage.

The completed 'Validation' screen for the example used here can be seen in Figure 65 below.

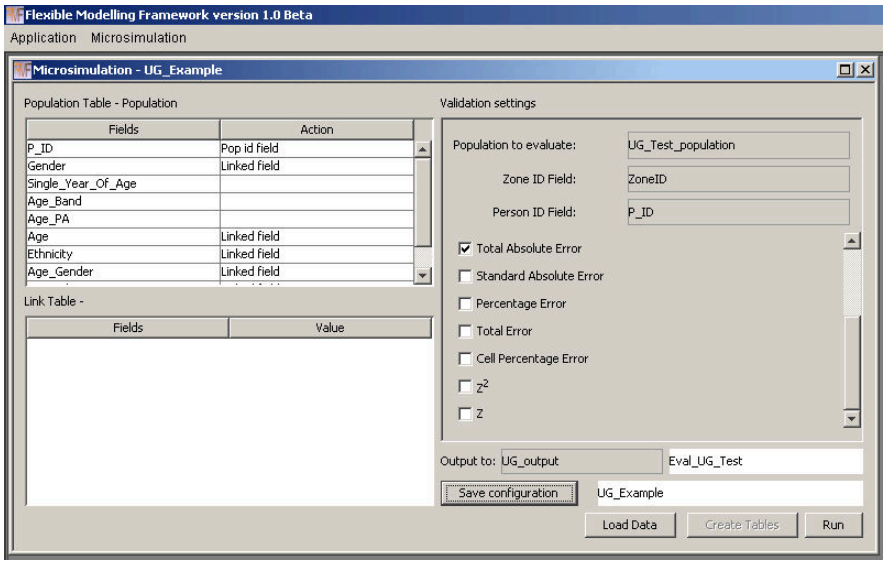


Figure 65: Completed evaluation window

Loading the data enables the 'Create Tables' button. Once the 'Create Tables' button is clicked a progress bar will appear in the 'Processes' tab showing how far through the summary creation the application is, Figure 66.

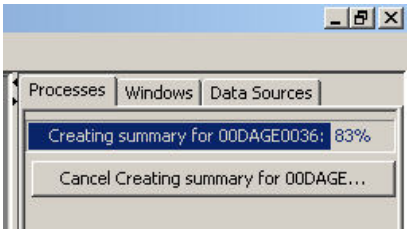


Figure 66: Create tables process running

This will take some time especially if there are many zones or a large population. The summary tables created for the example used here are shown in Figure 67 below.

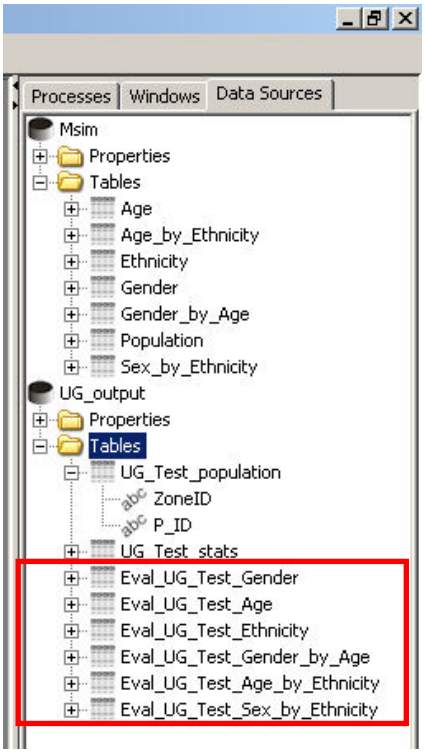


Figure 67: Example summary tables created

Once the summary tables have been created the evaluation can be run to generate the fit statistics and fit summary tables. The fit statistics are calculated by comparing the observed values in the constraint and evaluation tables to those created from the synthesised population in the summary tables. Three types of fit table are created (see Figure 68):

1. An overall summary fit table is created with the name <<output prefix>>_gof, in the example here Eval_UG_Test_gof. The table will contain a field called 'VariableName' which will have a record with the name of each table linked to the sample population table. It will also contain a field for each fit statistic chosen and an overall value for the statistic for each constraint / evaluation table in the relevant row.

2. A summary fit table by zone with the name <<output prefix>>_zones_gof, in the example here Eval_UG_Test_zones_gof. The table will contain a field with the full list of zone identifiers followed by a field for each constraint / evaluation table – fit statistic combination following the naming convention of <<table name>>_<<statistic abbreviation>>. This table provides a summary of the level of fit by zone.

3. The final level of fit table if for each constraint evaluation table and follow the naming convention of << output prefix>>_<<table name>>_<<statistic abbreviation>>_gof. For the age constraint and Total Absolute Error statistic in the example here this would be Eval_UG_Test_Age_TAE_gof. The structure of the table is identical to that of the original table with the first field containing the zone identifier followed by the field headings used in the link to the sample population. The values for each field are the values for the fit statistic chosen for the individual cell in the table.

These three levels of fit tables provide detailed information about where the model has performed well and poorly. The list of tables created in the evaluation process can be seen in Figure 68.

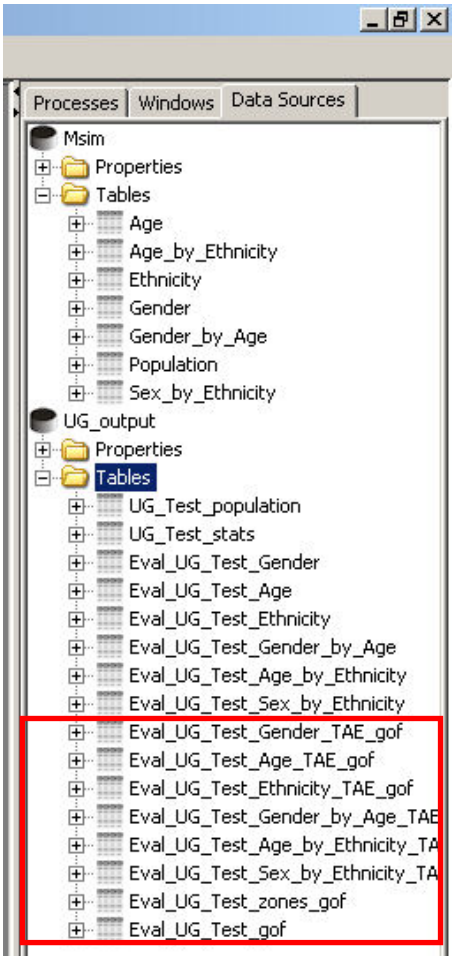


Figure 68: Summary fit tables for the example model configuration

7 References

Beckman, R. J., Baggerly, K. A ., & McKay, M. D. (1996). Creating synthetic baseline populations. *Transportation Research, 30,* 415-429.

Brown, L., & Harding, A. (2002). Social modelling and public policy: Application of microsimulation modelling in Australia. *Journal of Artificial Societies and Social Simulation,* 5(4) 6. http://jasss.soc.surrey.ac.uk/5/4/6.html

Harland, K., Heppenstall, A., Smith, D., and Birkin, M. (2012). Creating Realistic Synthetic Populations at Varying Spatial Scales: A Comparative Critique of Population Synthesis Techniques. *Journal of Artificial Societies and Social Simulation* **15** (1) 1. [Available From] http://jasss.soc.surrey.ac.uk/15/1/1.html

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science* 220, 671-680.

McFadden, D., Cosslett, S., Duguay, G. & Jung, W. (1977). Demographic Data for Policy Analysis . *Urban Travel Demand Forecasting Project, Final Report Series, Vol VIII.* University of California, Berkeley and Irvine: Institute of Transportation Studies.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M., Teller, A. H., & Teller, E. (1953). Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics, 21*, 1087-1092.

Orcutt, G. H. (1957), A new type of socio-economic system, *The Review of Economics and Statistics, 39,* 116-123

Orcutt, G. H., Greenberger, M., Korbel, J. and Rivlin, A. (1961), *Microanalysis of Socioeconomic Systems: A Simulation Study,* New York, Harper and Row.

Smith, D.M., Pearce, J.R., & Harland, K. (2011). Can a deterministic spatial microsimulation model provide reliable small-area estimates of health behaviours? An example of smoking prevalence in New Zealand. *Health & Place, 17,* 618-624. [doi:10.1016/j.healthplace.2011.01.001]

Tomintz, M.N., Clarke, G.P., & Rigby, J. (2008). The geography of smoking in Leeds: estimating individual smoking rates and the implications for the location of stop smoking services. *Area, 40,341-353.* [doi:10.1111/j.1475-4762.2008.00837.x]

Voas, D. & Williamson, P. (2001). Evaluating goodness-of-fit measures for synthetic microdata. *Geographical andEnvironmental Modelling, 5,* 177 - 200. [doi:10.1080/13615930120086078]

Williamson, P. & Clarke, G.P. (1996). Estimating small-area demands for water with the use of microsimulation. In *Microsimulation for urban and regional policy analysis.* Clarke, GP (Ed.). London: Pion.

8 Suggested Further Reading

Microsimulation:

TANTON, R., Vidyattama, Y., Nepal, B., & McNamara, J. (in press). Small area estimation using a reweighting algorithm. *Journal of the Royal Statistical Society: Series A.*

WILLIAMSON, P., Birkin, M., & Rees, P.H. (1998). The estimation of population microdata by using data from small area statistics and samples of anonymised records. *Environment and Planning A, 30*, 785-816.

MERZ, J. (1991). Microsimulation - A survey of principles, developments and applications. *International Journal of Forecasting, 7*, 77-104.

Simulated Annealing:

INGBER, L. (1993). Simulated Annealing: Practice Versus Theory. *Mathematical and Computer Modelling, 18*, 29-57.