

*Editor: Bart Van Herbruggen*

*Last updated: 14 December 2007*

## ***TREMOVE v2.7 User Manual***

This document focuses on the technical aspects of running the TREMOVE model. The aim is to enable a user with basic IT skills to run the model and offer him a reference manual for making modifications.

The reader will need a basic knowledge of DOS batch scripting, as well as an understanding of the GAMS language, in order to understand certain parts.

# Table of contents

<b>TABLE OF CONTENTS</b> .....	<b>2</b>
<b>1. SYSTEM REQUIREMENTS</b> .....	<b>3</b>
1.1    HARDWARE REQUIREMENTS .....	3
1.2    SOFTWARE REQUIREMENTS .....	4
1.2.1 <i>Operating system</i> .....	4
1.2.2 <i>Microsoft Office</i> .....	4
1.2.3 <i>Additional software</i> .....	4
A WinZip.....	4
B SmartDraw.....	4
1.2.4 <i>GAMS software</i> .....	5
<b>2. GETTING STARTED</b> .....	<b>6</b>
2.1    INSTALL TREMOVE.....	6
2.2    RUN TREMOVE .....	6
<b>3. MODEL IMPLEMENTATION</b> .....	<b>8</b>
3.1    GAMS.....	8
3.2    ONE MODEL, THIRTY-ONE COUNTRIES.....	8
3.3    MODULAR STRUCTURE.....	8
3.3.1 <i>Special case: the demand module</i> .....	9
A Folders .....	10
B Structure of the demand module code.....	10
C Calling the demand module in the vehicle stock modules .....	11
D Data exchange with vehicle stock modules.....	11
3.4    TREMOVE FOLDERS .....	12
3.4.1 <i>Override mechanism</i> .....	12
<b>4. INPUT PROCEDURES FOR VEHICLE STOCK, EMISSIONS AND WELFARE</b> .....	<b>14</b>
4.1    OVERVIEW .....	14
4.2    INPUT DATABASE .....	15
4.2.1 <i>Updating data</i> .....	15
4.2.2 <i>Database design</i> .....	15
4.2.3 <i>Table parameter documentation</i> .....	18
4.3    GENERATING GAMS INCLUDE-FILES.....	20
4.4    INPUT FILES FOR THE DEMAND MODULE.....	22
<b>5. OUTPUT PROCEDURES</b> .....	<b>23</b>
5.1    OVERVIEW .....	23
5.2    CONVERTING GAMS MODEL OUTPUT TO STRUCTURED OUTPUT DATA .....	24
5.3    IMPORTING STRUCTURED OUTPUT DATA IN THE OUTPUT-PURE DATABASE.....	24
5.4    QUERIES TO GENERATE OUTPUT_PROCESSED DATABASE TABLES.....	25
5.5    GENERATING PIVOT TABLES .....	26
<b>6. ADDING SIMULATION SCENARIOS</b> .....	<b>28</b>
<b>7. FILE STRUCTURE</b> .....	<b>30</b>

# 1. System requirements

## 1.1 Hardware requirements

The minimal hardware requirements for using TREMOVE are :

- Any computer that is able to run at least Windows 2000, should be able to run TREMOVE. In the past, Transport & Mobility Leuven has run TREMOVE on laptops with Pentium-III processor and 512MB RAM.
- However, for best results it is best to run TREMOVE on recent hardware, otherwise the run time will be very long.
- See below for recommended specifications for CPU, RAM and hard drive.

The runtime of TREMOVE depends on :

- CPU: Most of the TREMOVE run is completely processor limited.  
So the faster the CPU, the better.  
Note: All calculations are single-threaded, so only one CPU-core is used.
- RAM: A minimum of 1GB is needed to avoid swapping to disk, as this severely slows down the calculations. Some calculations use a bit more than 1GB of RAM, so 2GB (or more) is recommended.
- STORAGE: Some parts of the TREMOVE runs heavily depend on the speed of the hard drive. Laptop drives are considerably slower than desktop hard drives. If several scenario runs for many countries are envisaged, running TREMOVE on a desktop computer is recommended.  
Fast 7.200rpm SATA desktop drives are recommended, drives with higher spindle speeds (10.000rpm or higher) are even better.

For example, following hardware configurations are used by Transport & Mobility Leuven :

- Configuration 1 : TREMOVE run time for one country : 15 minutes
  - CPU: Intel Core 2 Duo E6850 (3GHz - 4MB L2-cache)
  - RAM: 8GB (dual channel 667MHz DDR2)
  - STORAGE: 10.000rpm SATA-drive  
(Western Digital Raptor 1500ADFD – 150GB – 16MB buffer)
  - Operating system: Windows XP Pro 64 bit edition SP2
- Configuration 2 : TREMOVE run time for one country : 30 minutes
  - CPU: Intel Pentium IV 540 (3,2GHz - 1MB L2-cache)
  - RAM: 1GB (dual channel 533MHz DDR2)
  - STORAGE: 7.200rpm SATA-drive  
(Seagate Barracuda 7200.7 – 160GB – 8MB buffer)
  - Operating system: Windows XP Pro SP2

## 1.2 Software requirements

### 1.2.1 Operating system

We only tested TREMOVE on the following operating systems:

- Windows XP (Professional)
- Windows XP (Professional) 64 bit edition

*We presume that TREMOVE will also work with Windows 2000 (Professional).*

Notes:

- TREMOVE has only been tested on these Operating Systems running in “native mode”. Windows running in a virtual environment (such as Parallel Desktops for Mac or VMware Workstation) is not recommended.
- TREMOVE has only been tested on “pc hardware”. Tremove *should* be able to run on Mac Hardware if it is running Windows in native mode. But since this has not been tested, it is not recommended.

*We invite TREMOVE users to share experiences if they would test TREMOVE on other systems..*

### 1.2.2 Microsoft Office

TREMOVE uses Microsoft Office. The model is developed with Office XP and is expected to work also with Office 2003. The extensive output procedures of the model do not work with Office 2000.

### 1.2.3 Additional software

#### A WinZip

TREMOVE uses WinZip (and the Winzip Command Line Support Add-On) to compress the intermediate output of the model, which is stored, to limit the burden on the user's hard disk. It is not a vital part of the model, however, without WinZip the model will not automatically store the intermediate results. You can download and purchase WinZip and the Command Line Support Add-On at <http://www.winzip.com>.

Please be aware that WinZip is not free software, but the purchase cost is limited (in October 2007 it was about 30 euro). As an alternative, you can request TML to provide you an alternative batch file for running TREMOVE which will not require WinZip. This batch file will not delete the uncompressed intermediate model output.

#### B SmartDraw

The documentation of TREMOVE uses a tool called SmartDraw to illustrate the structure of the GAMS model code. There are two versions of this documentation available. The first version is chapter 7 in this manual, just in text format. In order to view this, you do not need SmartDraw. The second version is a SmartDraw-document. The advantage of the second version over the Word version is that it includes hyperlinks that allow you to navigate through the hierarchical structure of the GAMS code.

SmartDraw can be downloaded freely from [www.smartdraw.com](http://www.smartdraw.com)

#### 1.2.4 GAMS software

TREMOVE is programmed in GAMS (General Algebraic Modeling System). The TREMOVE v2.7 model version can be executed with the free demo version of GAMS v22.0. No GAMS license is needed.

This version of GAMS can be downloaded at [http://www.gams.com/download/download\\_old.htm](http://www.gams.com/download/download_old.htm). On this page follow the link to the GAMS ftp site (<ftp://ftp.gams-software.com/>) and download the folder `"/22.0/systems/win/"`.

GAMS installation notes are available on the same website (<http://www.gams.com/docs/gams/win-install.pdf>). Actually, only 2 of the procedures described in these installation notes need to be executed :

1. Run setup.exe
2. Add the GAMS directory to your path

Note that TREMOVE has been developed in GAMS v22.0, compatibility with other versions of GAMS is not assured. Also, some of the earlier TREMOVE versions (e.g. v2.5) can not be run without a GAMS license.

## 2. Getting started

### 2.1 Install TREMOVE

TREMOVE is developed for the Microsoft Windows platform. There is no installation program for the TREMOVE model. Basically, all you have to do is download the TREMOVE archive file from <http://www.tremove.org/model/> and unzip it in a chosen folder.

However, the scripts which are used for running TREMOVE rely on a number of other applications:

- *Microsoft Access*
- *Microsoft Excel*
- *Winzip*
- *GAMS v22.0 (free demo version)*

TREMOVE needs to know the location of the program files of MS Office and Winzip. These are set in the `TREMOVE.bat` file (which can be found in the folder `\Model\Run`). The following lines in `TREMOVE.bat` ensure that every time Access, Excel or Winzip is called, these programs are searched for in the specified paths:

```
set Oldpath=%PATH%
PATH =%OldPath%;C:\Program Files\Microsoft Office\Office10\;C:\Program
Files\WinZip\
```

The first line reads the current Path settings of your system (which you do not want to lose). The second line adds the location of your MS Office programs and of Winzip to the existing folders in your system's Path settings. The adapted settings are valid during the entire model run, the settings return to their original state after the `TREMOVE.bat` (DOS) window is closed.

The next 3 code lines in the `TREMOVE.bat` file make that, if Access, Excel or Winzip can not be found in the Path folder locations, the model will not run but display an error message. In that case you have to change the specified paths in the code lines mentioned above. Of course, you will get a similar message if MS Office or Winzip are not installed on your pc.

TREMOVE uses macros in Excel to generate the pivot tables that display the final results. In Excel the macro security level should be set at 'low', to enable the macro's to be executed. This can be done in Excel via `Tools->Macro->Security`.

To allow running GAMS models as TREMOVE it is needed to express decimals with a period (".") in all input and output files. Comma's (",") will not work. The easiest way to assure this, is to adapt the "regional and language settings" of your pc (`control panel > Regional and Language Options > Customize`). Set *decimal symbol*: ".", *digit grouping symbol* "." and *list separator* ",".

### 2.2 Run TREMOVE

To run TREMOVE, execute the DOS batch script `TREMOVE.bat` (by double clicking the file) in folder `\Model\Run`. Alternatively, one can start `TREMOVE.bat` from the DOS-prompt. The latter approach has the advantage that one can follow the proceedings of the calculations in the DOS-window.

Before starting a run, one has to specify the simulation scenario and countries by editing the `TREMOVE.bat` script in a text editor (e.g. in Notepad):

Choose the countries you want to run the model for by limiting the list of country codes in line  
`set REMOVE_COUNTRIES=AT BE CH CZ DE DK ES FI FR GR HU IE IT LU NL NO PL PT SE SI UK EE LT LV MT CY SK TR HR RO BG`

Choose the simulation scenario you want to run by editing the line  
`set REMOVE_SCENARIO=FUELTAX`

The possible simulation scenarios correspond to the subfolders in `\Model\Scenarios`. In the distributed TREMOVE version these are:

- NULL (“do nothing”, i.e. simulation equals basecase)
- FUELTAX (tutorial fuel tax policy scenario)

Choose a scenario variation by editing the line  
`set REMOVE_SCENARIO_VARIANT=HARMONISATION`

Each scenario can have a set of variations corresponding to the subfolders in the `\Model\Scenarios` folder. For example, scenario FUELTAX has only one variation:

- HARMONISATION

The `TREMOVE.bat` script ensures that the following output files are automatically created:

- Excel pivot tables in folder `\Pivots`
- Access output databases in folder `\Pivots`
- Zip archives containing intermediate files (`.lst`, `.csv`, `.gdx`, `.g..`) for each country in folder `\Model\Output`. These archives contain data that have been calculated during the run. These files can a.o. be used to display extra model output values that are not in the standard pivot tables (without having to redo the whole model run).

Note that these output files are automatically overwritten if a new run is executed. It is good practice to copy and store outputs elsewhere after each run.

### Expert Tips:

If you are an ‘expert user’, you will probably want to *short-cut* some steps in the script code. For example, you might want to generate pivot tables directly, starting from the archived output of an earlier TREMOVE run. A custom script can be made for such a task. This will, however, require a good understanding of the individual steps which are described in the different chapters of this document.

If changes in model or input files are made, it might be worth to perform a *compilation test for one country* before running the complete model. A compilation test can be performed as follows:

1. Open GAMS via the GAMS-IDE
2. Open project `\Model\Run\run.gpr`
3. Open file `\Model\Run\run.gms`
4. Adapt the file paths in the `idir` statement for the relevant country, scenario and scenario variation (see 3.4.1)
5. Click simultaneously Shift+Run-button (or Shift+F9)

### 3. Model implementation

The aim of this chapter is to give an overview the software used for building the TREMOVE model.

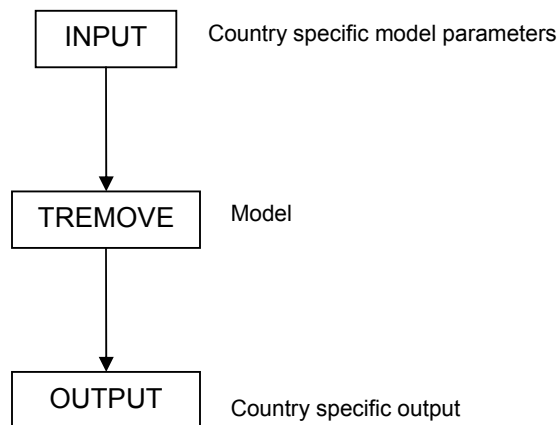
#### 3.1 GAMS

The TREMOVE model calculations are implemented in the GAMS language. GAMS is short for “General Algebraic Modeling System” and is particularly well suited for doing simulations that involve large scale non-linear optimisation problems. GAMS user guides and tutorials can be found on [www.gams.com](http://www.gams.com). This chapter is not intended to go into the details of the language. We restrict to some conceptual issues, in order to make it easy to understand the TREMOVE model for a reader with a minimal background in the GAMS language.

#### 3.2 One model, thirty-one countries

A common model is used for all countries. Thus, there is one TREMOVE model code that is used for all countries. As a consequence, modifications and developments of the model always apply to all countries, which results in guaranteed consistency between the countries. This common model then is executed for a specific country by feeding it with country-specific (exogenous) input parameters (see figure below).

Figure 1: A common model is run with country specific parameters

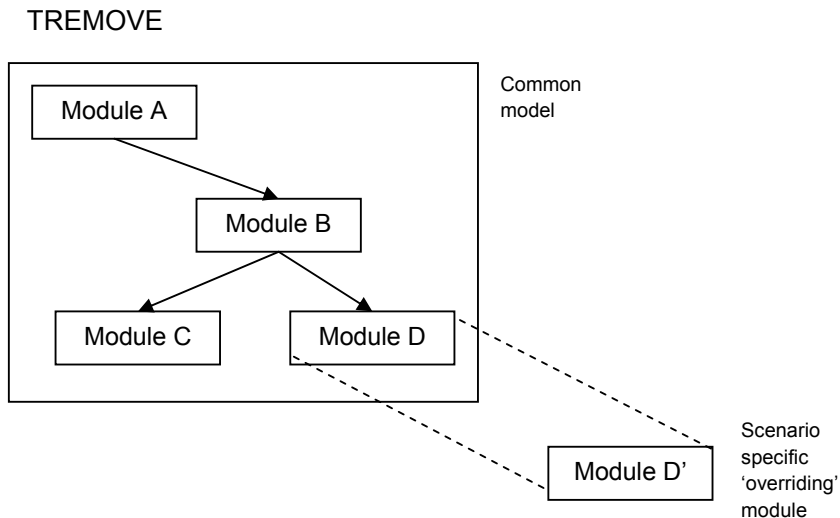


#### 3.3 Modular structure

The TREMOVE model is composed of several smaller pieces, called *modules*. Each module captures a single coherent piece of model behaviour. A modular structure offers a number of advantages:

- The software conceptual structure is emphasised, thereby improving readability;
- The same module can be called at different locations, thus avoiding to write the same lines of code more than once and lowering the probability of making mistakes;
- When the common model is subject to minor changes, e.g. for a specific scenario simulation, the necessary modifications can be restricted to one or a few modules, leaving the rest of the model unaltered. These modified modules from the common model then are *overridden* by an alternative formulation for these modules (see Figure 2 and 3.4.1).

Figure 2: Scenario specific REMOVE change by overriding limited number of modules



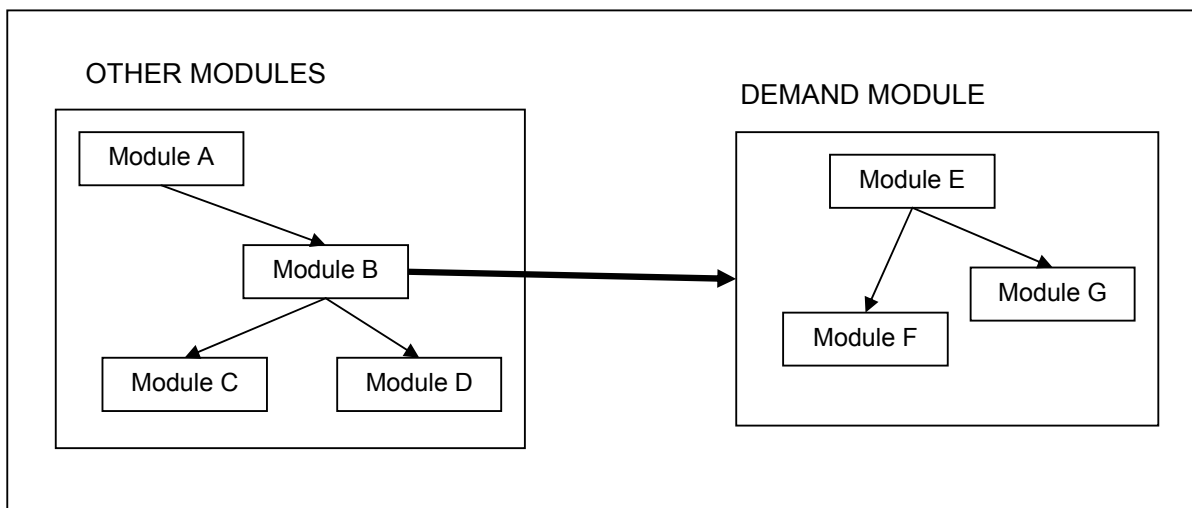
### 3.3.1 Special case: the demand module

Calling a module is straightforward: a special command (i.e. the GAMS `$include` or `$batinclude` command<sup>ii</sup>) ensures that the code of a called module is inserted in another module at the location where the call is made. The whole process takes place within one and the same program, which ensures automatic exchange of data between the modules.

There is, however, one module that is called in a totally different manner. As can be seen from Figure 3, this is the *demand module*, which is used for calculating price effects on transport demand. Actually, the demand module is not a module in the sense of those described above, but a full GAMS program on its own. And it consists of different smaller (sub)modules. This has two important consequences: Firstly, the `$include` and `$batinclude` commands are not suitable for calling the demand module (instead the special `execute` GAMS command is needed, as explained below). Secondly, because the calling module and demand module are no longer in the same program, specific procedures have been written to exchange data between the demand module and the other modules.

The following sections discuss the structure of the demand module in more detail.

Figure 3: Call to demand module (thick arrow) is different from calling other modules  
TREMOVE



## A Folders

The demand module consists of three folders:

- `\Model\Demand Module\Country Input`: contains subfolders for each modelled country with country-specific demand input data stored in an `.inc` file (the `.inc` file can be generated automatically from the `Demand_Input Database` in the same folder (see 4.4));
- `\Model\Demand Module\Demand`: contains the GAMS code for the actual demand module calculations, shared by all countries;

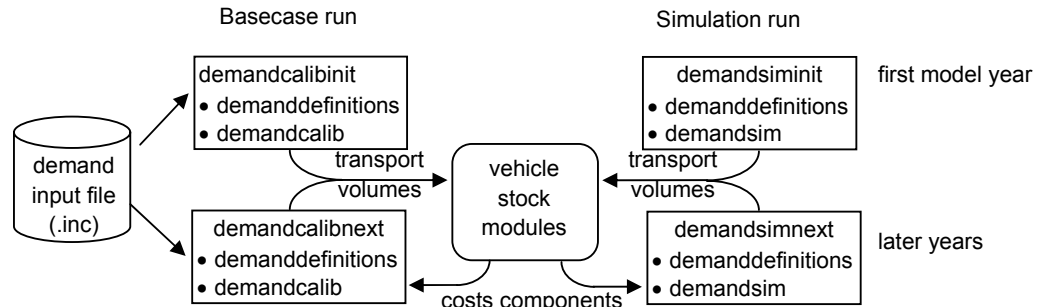
## B Structure of the demand module code

The demand module code in `\Model\Demand Module\Demand` contains seven GAMS files:

- `Demanddefinitions.gms`
- `Demandcalibinit.gms`
- `Demandcalibnext.gms`
- `Demandcalib.gms`
- `Demandsiminit.gms`
- `Demandsimnext.gms`
- `Demandsim.gms`

Figure 4 gives an overview of how these different files are used during a TREMOVE run for one country.

Figure 4: Demand module GAMS code structure



### C Calling the demand module in the vehicle stock modules

As mentioned earlier, the demand module is called within the other modules by making use of the GAMS `execute`-command. More specifically, it is called from the modules that perform vehicle stock calculations. This is done in the `Call_Demand_Module.gms` file which differs (slightly) for each country and can be found in the country subfolder of `\Model\Input`. This file contains the following code line.

```
execute 'gams %1 wdir="..\Demand Module" idir="Country Input\AT";Demand';
```

This code line expects a parameter “%1” indicating the name of the specific GAMS file to be executed. Possible values for this parameter are `demandcalibinit`, `demandcalibnext`, `demandsiminit` and `demandsimnext`. These parameter values are passed through automatically during the execution of the vehicle stock module.

Next, the folder in which the demand module operates, thus in which intermediate and output files are stored, is indicated (parameter `wdir`)

The code line also specifies the paths in which the `.inc` input files and the `.gms` GAMS code files are located (parameter `idir`).

### D Data exchange with vehicle stock modules

The demand module requires also data from the vehicle stock modules to function correctly. And, after its execution, demand module results are returned to the vehicle stock modules. This data-exchange is done through the use of `.gdx`-files. GDX is the “GAMS Data Exchange” format.

The file `Write_Demand_Module_Input.gms` in `T:\Model\Vehicle Stock Module module` ensures that the necessary data is written to following `.gdx` files, which are stored in the `\Module\Demand Module` folder:

- `Costroad.gdx`
- `Costpublic.gdx`
- `Costplane.gdx`
- `Costis.gdx`
- `Networkcost.gdx`

These files contain price information for all transport modes that is calculated in the vehicle stock modules and that is read as input by the demand module.

The demand module writes its results to the following `.gdx` files, which are stored in the `\Module\Demand Module` folder:

- `Demandroad.gdx`
- `Demandpublic.gdx`
- `Demandplane.gdx`
- `Demandis.gdx`

In file `Read_Demand_Module_Output.gms` in folder `T:\Model\Vehicle Stock Module module` these `.gdx` files are read in by the vehicle stock modules.

These files contain transport volume information for all modes that results from the demand module and that is read as input by the other modules.

### 3.4 TREMOVE folders

The TREMOVE archive file that is downloaded has three main folders:

- `\Input DB`: contains the input database and scripts generate coded input files (see chapter 4 for more details);
- `\Model`: bundles all GAMS code and execution scripts;
- `\Pivots`: output in pivot tables and output databases (see chapter 5 for more details).

The `\Model` folder is further divided into:

- `\Vehicle Stock Module`: GAMS code for vehicle stock modelling;
- `\Demand Module`: GAMS demand code is stored in subfolder `\Demand`, country specific demand data in `\Country Input` and scenario specific code in `\Scenarios`;
- `\Welfare Module`: GAMS code for welfare calculations;
- `\Run`: GAMS project directory, containing execution batch scripts. `TREMOVE.bat` can be edited and executed to run a simulation (see section 2.2 for details);
- `\Input`: GAMS include input files (`.inc` extension) which are shared by all countries. Country specific input files are stored in the country subfolders, along with other country specific code;
- `\Scenarios`: contains one subfolder per scenario for scenario specific GAMS code (see chapter 6 for more details);
- `\Report`: GAMS code for TREMOVE output post-processing (see section 5 for more details);
- `\Output`: location of the zipped intermediate files after running TREMOVE with the `TREMOVE.bat` script.

#### 3.4.1 Override mechanism

The model code is organised in several folders. By default, GAMS only looks into the project folder for code. You can tell GAMS to look in other folders by using the `idir` call parameter. For example, suppose you execute the GAMS command in folder `\Model\Run` by typing

```
gams run.gms idir="..\Input\BE";"..\Input";"..\Vehicle Stock
Module";"..\Welfare Module"
```

Then, everytime GAMS encounters a `$include`-statement in the code (e.g. `$include Main`), it will look for the file (e.g. `Main.gms`) first in the working directory<sup>iii</sup> and then in each of the folders passed by `idir`.

*The folders are searched in the passed order. If a file with the same name exists in several locations, the first one found will be used.*

This is an important property. It allows the implementation of the override mechanism described in Figure 2: if you want to override `Main.gms`, just put a modified version of the file in a new folder, e.g. `Model\New_Main` and make sure it is searched before the folder `Vehicle Stock Module` (the original location of `Main.gms`):

```
gams run.gms idir="..\New_Main";"..\Input\BE";"..\Input";"..\Vehicle Stock
Module";"..\Welfare Module"
```

This feature is used to implement alternative scenario simulations in `TREMOVE`. Code lines for scenario simulations are stored in a subfolder of `\Model\Scenarios`. These subfolders are, in turn, passed as the first items in the `idir` folder string. For example the command:

```
gams run.gms
idir="..\Scenarios\FUELTAX\HARMONISATION";"..\Scenarios\FUELTAX";
"..\Input\BE";"..\Input";"..\Vehicle Stock Module";"..\Welfare Module"
is used to execute the HARMONISATION variant of a FUELTAX scenario for Belgium.
```

The default line in the standard batch file that calls the `TREMOVE` gams code is:

```
gams run
idir="..\Scenarios\%3\%2";"..\Scenarios\%3";"..\Input";"..\Input\%1";
"..\Vehicle Stock Module";"..\Welfare Module" s=save_%2
```

In which `%1`, `%2` and `%3` are command-line parameters:

`%1` = country

`%2` = sim variant, e.g. HARMONISATION

`%3` = sim group, e.g. FUELTAX

If you want to use country specific override files in your scenario run, you can modify this line as follows :

```
gams run
idir="..\Scenarios\%3\%2\%1";"..\Scenarios\%3";"..\Input";"..\Input\%1";
"..\Vehicle Stock Module";"..\Welfare Module" s=save_%2
```

Of course, you then need to make a country-specific scenario folder,

e.g. : `Model\Scenarios\FUELTAX\HARMONISATION\AT`

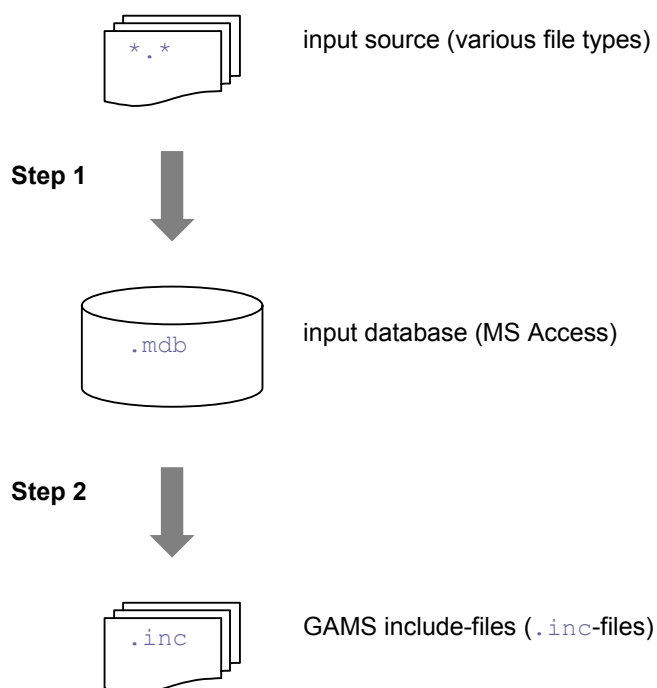
## 4. Input procedures for vehicle stock, emissions and welfare

### 4.1 Overview

Introducing new basecase input data into the TREMOVE model must be done in two steps (see Figure 5):

1. Import the data into the input database
2. Extract GAMS include (.inc) files from the input database

Figure 5: The different steps for producing TREMOVE input



Section 4.2 describes the structure of the input database in more depth. The second step, converting the database into GAMS include-files, is the subject of section 4.3.

## 4.2 Input database

### 4.2.1 Updating data

The input databases is file `TREMOVE_input.mdb` in folder `\Input DB`. This database contains

- Lists of the elements of sets used in the model code (e.g. the years, the vehicle types, etc.);
- All information related to the exogenous baseline, except those included in the demand module input database, see 3.3.1 (e.g. public transport tariffs, fuel resource costs, ...);
- Data for the baseyear (e.g. 1995 vehicle fleet in Austria) or timeseries for historic years (e.g. 1995-2004 vehicle fleet in Belgium);
- Fixed model coefficients (e.g. COPERT-coefficients for emission calculations).

Once you have located the Access table in which parameter you want to change is stored (see 4.2.3), updating data the database is easy: just replace the original data by the new data in the database tables.

#### Time Saver Tip:

Importing data into MS Access can be done in many ways but, in most cases, the fastest is with the help of MS Excel. Start from a sheet which contains the data in the same layout as the database table. Then simply copy the data, select the entire last row of the Access table and paste the data in it.

When modifying parameter data in the database, don't forget to update the `Parameter documentation` table (see 4.2.3).

Only *exogenous* model parameters and their dimensions (sets) are stored in the database, and can be modified in the database. The remaining *endogenous* sets and parameters are defined and calculated in the model code. This distinction between *endogenous* and *exogenous* data is reflected in the file structure of the TREMOVE model. *Exogenous* parameters and sets, which are stored in the database and converted into `.inc` files to be included by the GAMS code, are located in the `\Model\Input` folder. The remaining parameters and sets are *endogenous* and are directly defined and calculated in the GAMS code, which is located in other `\Model` folders.

Changes to the databases are not automatically propagated to the GAMS mode. They will only take effect after the generation of formatted data files ("include-files" with extension `.inc`) which can be imported by the GAMS model (see section 4.3).

A first glance on the input database might be a bit overwhelming. The structure will be much clearer when some underlying design principles are understood. Moreover, one table, named `Parameter_documentation`, serves as the road map for locating the GAMS parameters. Both this table and the design principles are explained below.

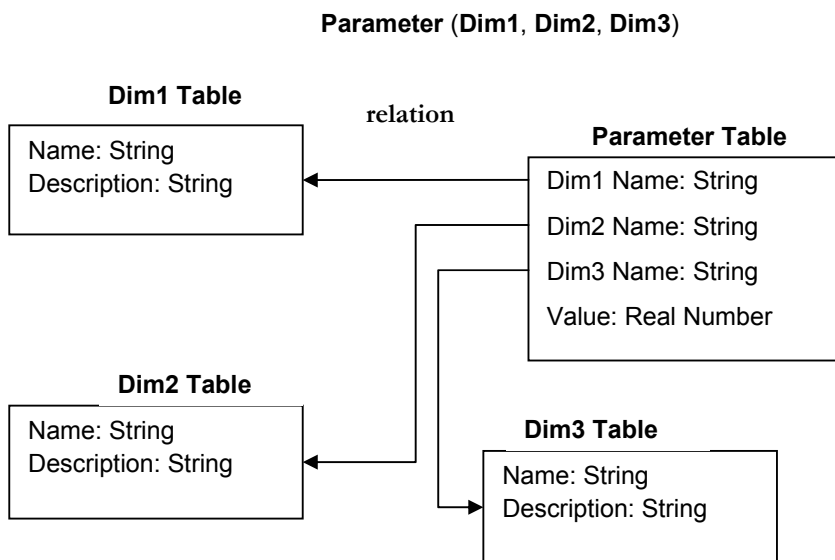
### 4.2.2 Database design

The TREMOVE input database is designed in MS Access using relational database principles, i.e. as a collection of tables and relations between them. Relational databases are used to store data *efficiently*, taking up the least amount of storage room and allowing fast retrieval using the Standard Query Language (SQL).

This usually means that the data end up scattered over many tables and are hard to locate without the help of design blueprints.

The design blueprints for the TREMOVE input database are fairly straightforward. Only exogenous GAMS *parameters* are stored, along with their dimensions. The dimensions, in turn, correspond to GAMS *sets* (see Figure 6 for an illustration).

Figure 6: Relational database representation of a GAMS parameter with three set dimensions (Dim1, Dim2 and Dim3)



Tables containing GAMS *sets* are easy to locate in the database. They have the same name as in the GAMS model or one that is very similar, e.g.

- T (set of years);
- N (set of vehicle ages);
- VEHFULL (full set of road vehicle types);
- TECHFULL (full set of road vehicle technologies);
- ...

All records are uniquely identified within the table by the *Name* field. This is, in relational database terms, the *primary key*.

Contrary to sets, no table with the same name as a GAMS *parameter* exists. The reason is that many GAMS parameters share the same dimensions, e.g.:

- RVAT (RVEH, T, RUN) VAT rate - by road vehicle type and year
- RMILnew (RVEH, T, RUN) Mileage of a new vehicle - by road vehicle type and year

Thus, it is more convenient to bundle these into one table with an additional *Name* field. For example, both parameters *RVAT* and *RMILnew* are stored in table `T_VEHICLE_PARAMETER` (see Figure 7).

Figure 7: Table **T\_VEHICLE\_PARAMETER** containing GAMS parameters **RVAT** and **RMLnew**. Also shown: Tables **VEHFULL** (corresponding to GAMS set **RVEHFULL**) and **T** (GAMS set **T**)

The screenshot shows the Microsoft Access interface with three tables displayed in Datasheet View:

- T\_VEHICLE\_PARAMETER : Table**

Country	Year	Veh	Name	Value
BE	1995	PCDB	RMLnew	30626
BE	1995	PCGB	RMLnew	23276
BE	1995	PCG2	RMLnew	20000
BE	1995	PCGS	RMLnew	13475
BE	1995	MP	RMLnew	1870
ES	2014	BUS	RVAT	0
ES	2013	PCDB	RVAT	0.16
ES	2013	PCL	RVAT	0.16
ES	2013	PCDHB	RVAT	0.16
ES	2013	PCGHM	RVAT	0.16
- VEHFULL : Table**

Name	Description
BUS	buses
BUS_CNG	compressed natural gas buses
COACH	coaches
HTD1	heavy duty trucks - diesel - <7.5t
HTD2	heavy duty trucks - diesel - 7.5-16t
HTD3	heavy duty trucks - diesel - 16-32t
HTD4	heavy duty trucks - diesel - >32t
HTG	heavy duty trucks - gasoline
LTD	light duty trucks - diesel
- T : Table**

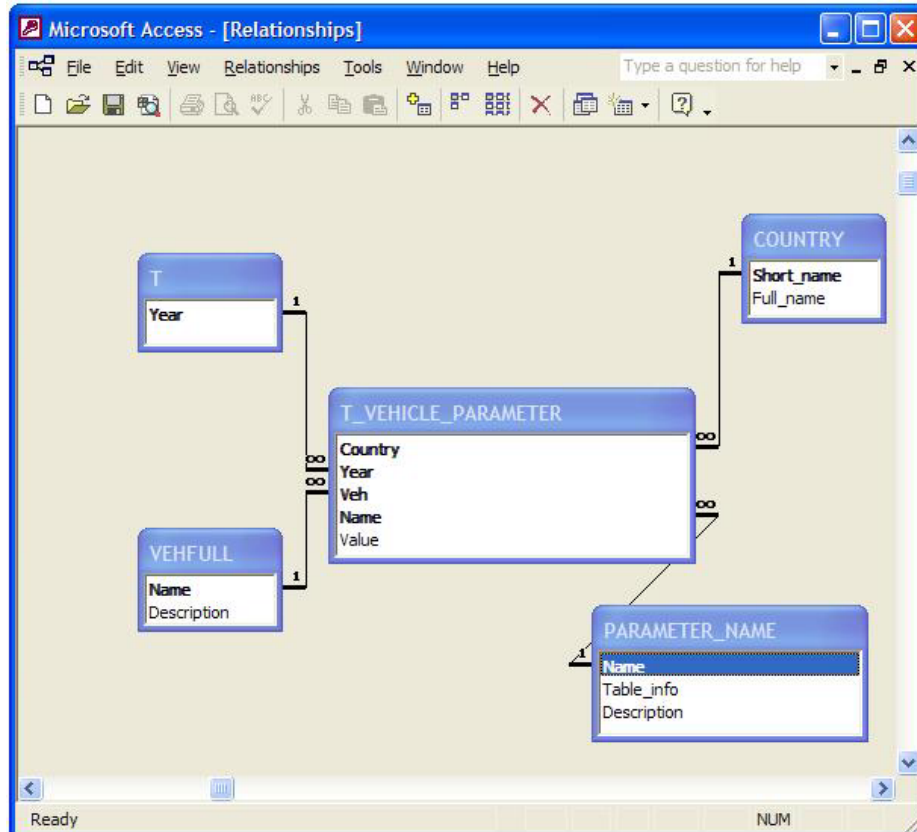
Year
1995
1996
1997
1998
1999
2000
2001
2002
2003

Figure 8 shows the relations between table **T\_VEHICLE\_PARAMETER** and its dimensions **T** and **VEHFULL** (this view is accessible via Access menu **Tools > Relationships...**). For example for every value for **Veh** in table **T\_VEHICLE\_PARAMETER**, a corresponding value of **Name** must exist in table **VEHFULL**.

Dimension **RUN**, e.g. in **RVAT (RVEH, T, RUN)** is used in the model to distinguish between data for the basecase run (**RUN = "BC"**) and for the scenario simulation (**RUN = "SIM"**) run. This dimension is always ignored in the input database, as it only contains basecase data.

Also note that the input database contains exogenous model data for *all* countries. Therefore, all parameter tables have an additional relation with table **COUNTRY**. When the value of a parameter is the same in all countries, as is the case the special value **ANY** is used in the **Country** field.

Figure 8: Relationships of T\_VEHICLE\_PARAMETER



### 4.2.3 Table parameter documentation

Table `Parameter_documentation` is the only table where all GAMS parameters are collected, making it very convenient as a road map for the input database. It is also used to keep track of the different data sources.

As can be seen in Figure 8, the name of the parameter (`Name` field) must correspond to a record in `Parameter_documentation`. This is true for `T_VEHICLE_PARAMETER` as well as for all other parameter tables.

The `Description` field repeats the GAMS parameter description, thus giving additional information about the parameter to the user (see Figure 9).

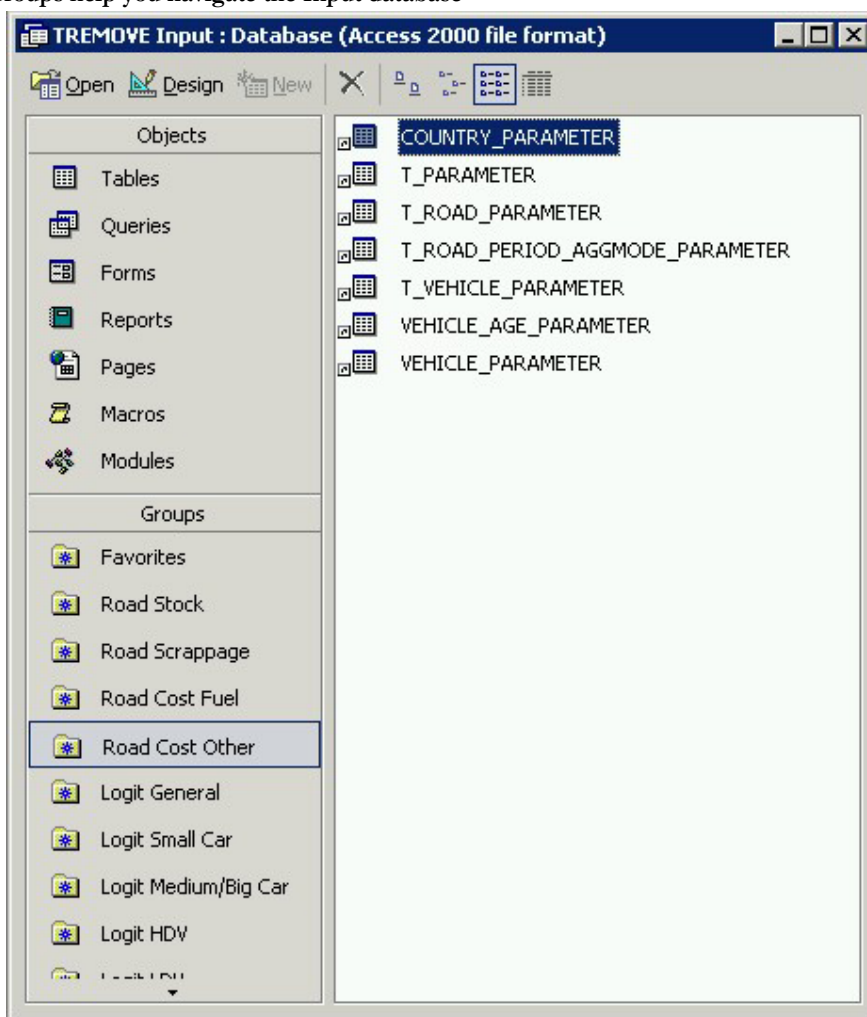
The field `Table_info` in table `Parameter_documentation` contains the table in which a parameter is stored. It has no other purpose than pointing the database user into the right direction.

Figure 9: Table **Parameter** documentation serves as a road map for the input database, pointing to the table where a GAMS parameter can be found

Name	Table_info	Description
+ FUELSPEC	T_FUEL_FUELPAR_PARAMETER	Fuel specification history
+ GAMMA1	COUNTRY_PARAMETER	Life expectancy parameter 1
+ GAMMA2	COUNTRY_PARAMETER	Life expectancy parameter 2
+ GAMMA3	COUNTRY_PARAMETER	Life expectancy parameter 3
+ GAMMA4	COUNTRY_PARAMETER	Life expectancy parameter 4
+ GAMMA5	COUNTRY_PARAMETER	Life expectancy parameter 5
+ IWCONFIG	T_IWVEH_PARAMETER	Indicates for each configuration if a
+ IWCONSffuel	IWVEH_PARAMETER	Inland Waterway Fuel consumption
+ IWEMIF	IWVEH_POLL_PARAMETER	Inland Waterway Emission factor -
+ IWENGCOST	IWVEH_PARAMETER	Additional cost of the engine type c
+ IWEQCOST	IWVEH_PARAMETER	Additional cost of the engine type c
+ IWFCOST	T_GENERAL_PARAMETER	Inland waterway fuel resource cost
+ IWREDUC	IWVEH_PARAMETER	Reduction in fuel consumption of th
+ IWFTAX	T_GENERAL_PARAMETER	Inland waterway fuel tax - in EURO
+ IWFUELdens	T_GENERAL_PARAMETER	Density of inland waterway fuels - c
+ L	MOTIVE_TAX_PARAMETER	Marginal cost of public funds

Another way to find the location of the data you are looking for, is by navigating the 'groups' in the database. The **groups** are listed on the bottom left side of the main window in the database (see Figure 10). Every group (eg 'Road Cost Other') contains a set of hyperlinks to the tables that contain data related to the topic (in case of 'Road Cost Other' these are the tables 'COUNTRY\_PARAMETER', 'T\_PARAMETER', 'T\_ROAD\_PARAMETER', 'T\_ROAD\_PERIOD\_AGGMODE\_PARAMETER', 'T\_VEHICLE\_PARAMETER', 'VEHICLE\_AGE\_PARAMETER', 'VEHICLE\_PARAMETER').

Figure 10: Groups help you navigate the Input database



### 4.3 Generating GAMS Include-files

Before an update of the database has effect in the GAMS model calculations, the data must be converted into a formatted file which can be “read” by the GAMS model. These are called “include-files” and have a `.inc` extension. The conversion is not included in the general `TREMOVE.bat` batch file because it is time consuming. Instead, one has to execute the conversion manually after each updated of the database.

The conversion is performed by a collection of files (DOS batch scripts), located in folder `\Input DB\Access - GAMS Conversion`:

Table 1 :Batch scripts for generation of `.inc` input files

Name of Batch script	Generates contents of
<code>Build_Sets_Time</code>	<code>Sets_Time.inc</code>
<code>Buid_Sets_Road</code>	<code>Sets_Road.inc</code>
<code>Build_Sets_Rail</code>	<code>Sets_Rail.inc</code>
<code>Build_Sets_IW</code>	<code>Sets_IW.inc</code>
<code>Build_Sets_Emission</code>	<code>Sets_Emission.inc</code>

Build_Sets_Air	Sets_Air.inc
Build_Sets_Life_Cycle	Sets_Life_Cycle.inc
Build_Sets_Airco	Sets_Airco.inc
Build_Sets_Road_Country	Sets_Road_Country
Build_Sets_Rail_Country	Sets_Rail_Country
Build_Parameters_IW	BC_IW.inc
Build_Parameters_Air	BC_Air.inc
Build_Parameters_Copert	Copert3.inc
Build_Parameters_Biofuel	BC_Biofuel.inc
Build_Sets_Road_Country	Sets_Road_Country.inc
Build_Sets_Rail_Country	Sets_Rail_Country.inc
Build_Parameters_Road_Country	BC_Road.inc
Build_Parameters_Logit_Country	BC_Logit.inc
Build_Parameters_Road_Emissions_Country	BC_Road_Emissions.inc
Build_Parameters_Rail_Country	BC_Rail.inc
Build_Parameters_Life_Cycle_Country	BC_Life_Cycle.inc
Build_Parameters_Welfare_Country	BC_Welfare.inc
Build_Parameters_Airco_Country	BC_Airco.inc

The scripts ending with `_Country` generate country-specific data and take a country code (AT, BE...) as a parameter.

Each script generates several related sets or parameters from the database into an include-file. The correct script (the one containing the parameter or set that has been updated in the database) can be located with a simple text-search.

Note that the file path to `TREMOVE_input.mdb` has to be specified in file `T:\Input DB\Access - GAMS Conversion\Mdb_name.txt` before the 'build' procedures can be executed.

The scripts write the resulting `.inc` files to folder `\Input DB\Access - GAMS Conversion`. These files should be copied to `\Model\Input` (a country subfolder for the country specific `.inc` files). This is the folder from which input data is read by the GAMS model.

This procedure is further automated via the scripts `Build.bat` and `Build_All`. With `Build`, one or more scripts (*all* Table 1 scripts by default) can be executed. `Build_All` gives the opportunity to run `Build` for a list of countries (*all* countries by default). `Build` will also copy the `.inc` files to the `\Model\Input` folder.

All Build-scripts make use of the `mdb2gms` tool.

See [http://www.gams.com/~erwin/interface/interface.html#1.4.4\\_Download MDB2GMS](http://www.gams.com/~erwin/interface/interface.html#1.4.4_Download MDB2GMS) for more information.

#### For expert users:

The Build-scripts first generate the `.inc` files in folder `\Input DB\Access - GAMS Conversion\Input`. Thereafter they are automatically copied to `\Model\Input`.

If ANY in a `Country` field is replaced by specific country codes, the related Build DOS batch script must be modified accordingly (add `'where country = %1'` to the query).

## 4.4 Input files for the demand module

The input for the demand module is stored in the database `\Model\Demand Module\Country Input\NEC_all.mdb`. The information stored in this database includes:

- Basecase transport quantities by transport mode
- Basecase travel speeds
- Basecase tariffs for public transport modes
- Duration of the peak and off-peak periods
- Values of time

All this data is derived from the SCENES model.

This database is not directly read by the GAMS code and it is not automatically converted to GAMS include files by the TREMOVE batch file.

The database has to be converted to formatted data files (again “include-files” with extension `.inc`) which can be “read” by the GAMS code. To perform this conversion you can use the batch file `extract.bat`, which is located in folder `\Model\Demand Module\Country Input`. This batch file calls `expMDB2INC.bat`, which does the actual conversion of (a part of) the database to an include-file for GAMS. This batch file does the conversion for one country. A list of country codes is passed as a command-line parameter by `extract.bat`. One can edit the list of countries in `extract.bat` to match only the countries for which data has been updated. Besides passing the country code, `extract.bat` also ensures that the output of the conversion is stored in the right country subfolder.

## 5. Output procedures

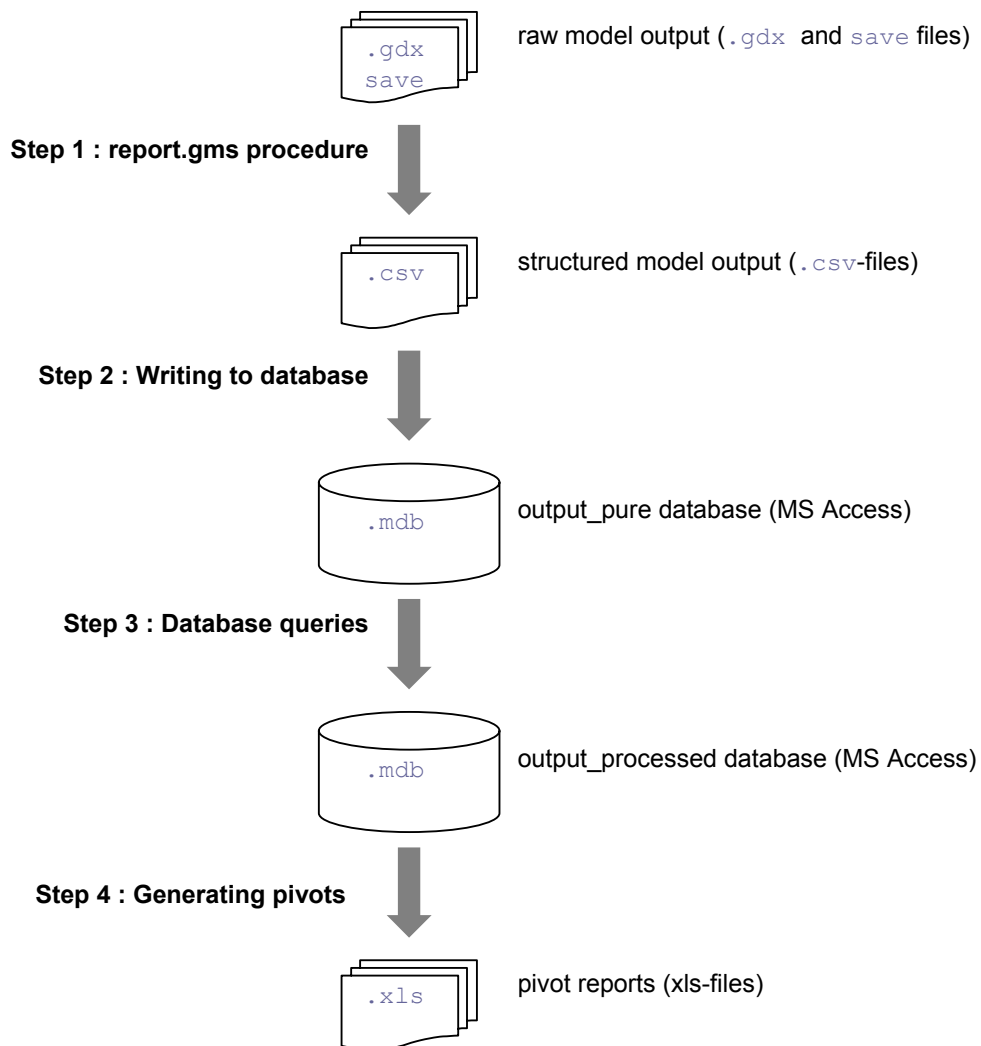
### 5.1 Overview

The TREMOVE output procedure converts “raw” GAMS model output into user reports in pivot tables. The procedure consists of following steps:

1. Convert raw GAMS model output to structured output data;
2. Import the structured output data in the output\_pure database;
3. Execute queries to generate output\_processed database tables;
4. Generate pivot tables based on the output\_processed database tables.

These steps are illustrated in , and further discussed in the following sections.

Figure 11: The different steps for producing TREMOVE output



## 5.2 Converting GAMS model output to structured output data

The first step in the output procedure is the conversion of “raw” model output to structured `.csv`-files that can be imported in the `output_pure` database.

The raw model output consists of several files:

- `TREMOVE save-file`: In this file, all GAMS sets and parameter values are “saved” with the `save` (or `s`) GAMS command (see GAMS documentation for more information). The save file does not include information from the demand module (which is a GAMS program on its own, see 3.3.1).
- `demandsimdump.gdx`: This file is produced by the demand module after its execution in the last year of the model run. This file contains a dump of the relevant demand module output parameters for all years.
- `Active_Country.inc`: This file contains the name code of the country for which the output is being processed (AT, BE...). It is automatically generated by the `tremove.bat` batch procedure, while `TREMOVE` is executed for a list of countries.
- `Variant.inc`: This file contains the name of the scenario variant that has been run. This name is used in the database and pivot tables to identify the scenario variant. This file is automatically generated by the `tremove.bat` file.

These raw output files are converted to structured tab-separated text files (`.csv`) by the GAMS program `report.gms`. This program is located and executed in folder `\Model\Report`. It must be executed using the `TREMOVE save-file` with the `restore` (or `r`) GAMS command. The `.gdx` file and the `.inc` are automatically read by the `report.gms` program. The `tremove.bat` procedure will automatically copy the raw output files to the folder `\Model\Report` and execute `Report.gms`.

`report.gms` produces an extensive set of `.csv` files that will serve as input to the `output_pure` database. These `.csv` files no longer contain specific references to GAMS set element names. The set element names have been converted into integer “id” numbers. Thus, the `.csv` files are composed of structured rows and columns of numbers, with almost no text characters. As normal users will not use these files to consult `TREMOVE` results, these files are automatically deleted by the `tremove.bat` procedure, after they have been imported in the `output_pure` database.

## 5.3 Importing structured output data in the output-pure database

The MS Access output databases `output_pure.mdb` and `output_processed.mdb` are located in folder `\Pivots`. The `.csv` files that are generated by `Report.gms` are automatically copied to this folder and are imported to tables in `output_pure.mdb` by the Access macro `StartImport`.

The `report.gms` code and the Access `StartImport` macro are executed for every individual country listed in `tremove.bat`.

### Expert Tips:

Actually, for the first country listed in `tremove.bat` the batch procedure calls macro `ImportIDs` before macro `StartImport` is called. The macro `ImportIDs` will clear the `output_pure` database and import the “id” numbers for the set elements. Thereafter, macro

`StartImport` is called for each listed country to import its output data in the database. Alternatively, one can start with an `output_pure` database that already contains output data for one or more countries, and add other countries. In that case one can run the new countries by using `tremove.bat` and adapt the batch procedure such that the macro `ImportIDs` is not executed. Macro `ImportIDs` is called for the first listed country by the following line in the batch file `Run_And_Import.bat` :

```
if exist firstpass.txt msaccess.exe "%DBDIR%\output_pure.mdb" /X ImportIDs
```

To avoid that macro `ImportIDs` is executed, one can write “rem” before this line.

Note that `Run_And_Import.bat` is a batch procedure that is executed for each country listed in `tremove.bat`.

After every country listed in `tremove.bat` has been imported, another macro is called (once): `UpdateTempTables`. This macro executes queries in the `output_pure` database to generate some extra tables in the database.

## 5.4 Queries to generate output\_processed database tables

The `output_processed.mdb` links to the `output_pure.mdb` and includes queries to generate the final TREMOVE output tables.

### Expert Tips:

The link is not automatically updated when the database files are moved from one folder to another (the `output_processed.mdb` will continue to refer to the information in an `output_pure.mdb` in the old folder).

So if one moves the database files to another folder, one has to update the link. This can be done as follows :

- A.1. Open the `output_processed.mdb`
- A.2. Go to `tools > Database utilities > Linked table manager` (if this is not installed on your PC install it or go to B.1)
- A.3. Make sure `Always prompt for a new location` is ticked
- A.4. Click `Select All`
- A.5. Click `OK`
- A.6. Browse to the right location of the `output_pure.mdb` and select it (click open)

Part B: only if you do not have the linked table manager.

- B.1. Go to the table view in the `output_processed.mdb`
- B.2. Delete all the linked tables (you recognise them because there is an arrow in front of their name). E.g. `Air emission` is a linked table, `hulp_EMI_All` is not. Make sure not to delete any of the other tables!
- B.3. Go to `File > Get external data > Link tables`
- B.4. Browse to `output_pure.mdb` in the right folder and select it (click open)
- B.5. Click all the tables you want to link to and click `OK`. The fastest way to do this is by first clicking `select all` and then unselected the fixed tables (`ctrl+click`). The fixed tables are `Welfare2`, `transposeEmiSel` and the tables starting with `hulp_`

The `tremove.bat` procedure will call in `output_processed.mdb` the macro `UpdateTempTables`. Once this is done, the final TREMOVE output tables can be generated by using pre-defined queries. To

do this, click in the `output_processed` database the `Queries` button in the left pane. Then, one can run the following queries (by double-clicking them) that generate the final output tables. Note that these tables contain all data that is reported in the eventual Excel pivot tables.

- Demand : generates the table with the data for the `demand.xls` pivot;
- Stock : generates the table with the data for the `stock.xls` pivot;
- Emissions\_All : generates the table with the data for the `emissions.xls` pivot;
- Welfare1 : generates the table with the data for the first four sheets in the `welfare.xls` pivot;
- Welfare2\_MT : generates the table with the data for the last sheets of the `welfare.xls` pivot.

The database can hold about 17 countries at once. Since Microsoft limits Access to handle files of 2 GB, it is not possible to put all the countries in one database.

## 5.5 Generating pivot tables

The final output of the TREMOVE model is reported in four Excel pivots in `\Pivots`:

1. `Stock.xls`
2. `Demand.xls`
3. `Emissions.xls`
4. `Welfare.xls`

The pivots are linked to the aforementioned queries in `output_processed.mdb`. Each pivot Excel sheet has a macro called `RefreshPivot` which refreshes the pivot data by linking to `output_processed.mdb`. It requires the database to be in the same directory as the Excel sheet.

The `tremove.bat` procedure makes use of an additional Excel sheet, named `Refresh.xls`. This latter Excel file does not contain any data, but contains only a special macro, which is automatically started when the file is opened. This macro starts each of the `RefreshPivot` macros in the other Excel sheets. Hence, the four pivot Excel files are automatically updated with the latest data from the `output_processed` database.

By default, the pivots only include data for five-year intervals (1995, 2000, 2005, 2010, ...). I.e. only for these selected years, the data in `output_processed.mdb` is transferred to the pivots. In order select other years, select a data cell in the pivot, right-click it and choose `Wizard... > Back... > Get Data...` This opens Microsoft Query. Click `View > SQL...` to adapt the selected years in this SQL code. Note that, given file size restrictions of Excel, the pivots cannot include data for all years and countries together, thus one has to make a selection.

### Expert user tip:

If the pivots and output databases are copied to another folder than `T:\model\pivots`:

- Using the `RefreshPivot` macro will refresh the pivot figures using the `output_processed.mdb` that is in the same folder as the pivots
- Just refreshing the pivot without using the macro will refresh the pivot figures using the `output_processed.mdb` in the location where you last applied the refresh macro (typically `T:\model\pivots`). This is also the case if the pivot queries are altered in the copied pivot file (e.g. if years are added to the pivots).

**Trouble-shooting :**

In case you get the error message [Microsoft][ODBC driver] not found :

This can be caused because your ODBC driver, which takes care of the communication between Excel and Access, is not recognized. The macro in excel that refreshes your pivot table expects the driver to be called 'MS Access Database', which is the default name in English PC's. To check and modify the name of your ODBC driver, please do the following:

- Go to your 'Administrative tools', either from our start menu or in your control panel;
- Select 'Data sources (ODBC)';
- You will see a screen, similar to Figure 12;
- Select the Microsoft Access Driver and click 'configure';
- You see Figure 13 - Adapt the name of the data source to 'MS Access Database';
- Click 'OK' twice.

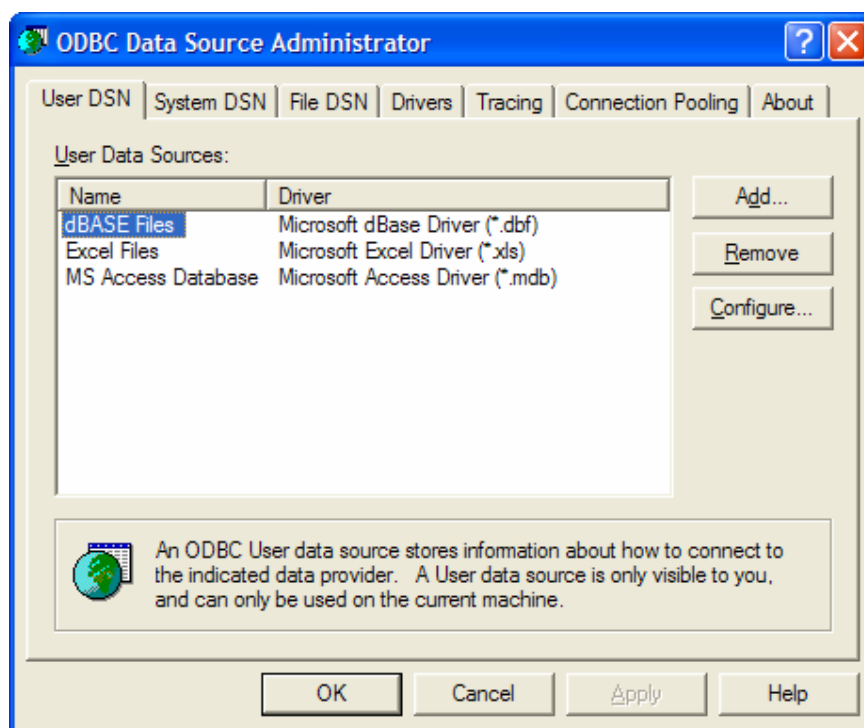


Figure 12: Select the Access Driver

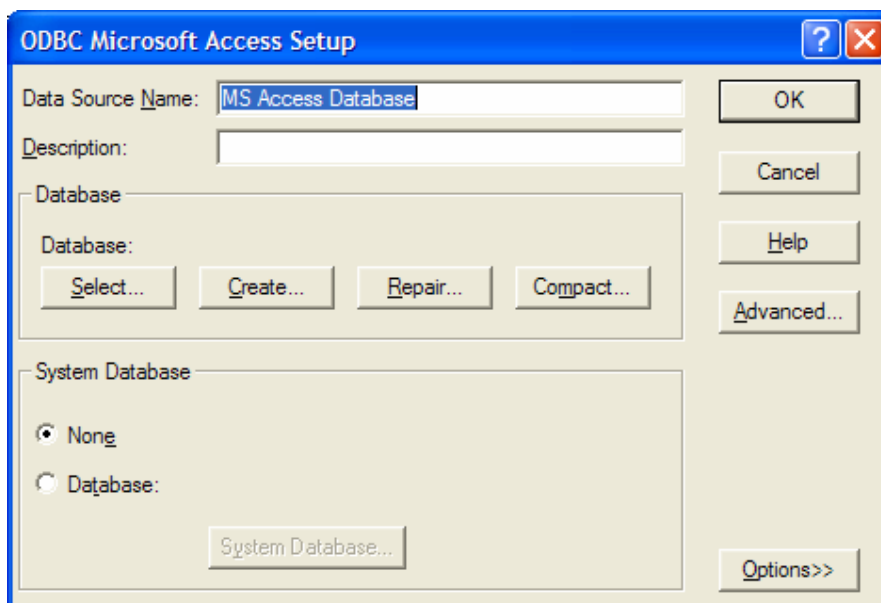


Figure 13: Adapt the Data Source Name

## 6. Adding simulation scenarios

Simulation scenarios are defined in the `\Model\Scenarios` folder. As explained in section 3.4.1, they make use of the override mechanism.

To be compatible with the `TREMOVE.bat` script (see section 2.2), following procedure has to be followed. Suppose one wants to add a scenario named “CAR\_POLICY”, these are the steps that have to be taken:

1. Create the folder `\Model\Scenarios\CAR_POLICY`. Here you will put the overridden files which are common to all CAR\_POLICY variations.
2. Choose a name for the first variation, e.g. CP\_1 and create a subfolder with the same name: `\Model\Scenarios\CAR_POLICY\CP_1`. Overridden files which are specific to the variation will be stored here.
3. Other variation folders can be created alongside the `\CP_1` folder.

When the scenarios are set up following this structure, `TREMOVE` can be run by editing the first lines in the `TREMOVE.bat` script and then executing it.

These folder conventions are the reason for the empty `\Model\Scenarios\NULL` and `\Model\Scenarios\NULL\NULL` folders. Using those in the `TREMOVE.bat` script results in a simulation scenario that is equal to the basecase, because no files are overridden.

Folder `Model\Vehicle Stock Module` contains two empty files: `Scenario_Main.gms` and `Scenario_Variation.gms`, which are called from `Init_Simulation.gms`. In the latter file simulation (“SIM”) input values are set equal to basecase (“BC”) input values, for a long list of input parameters. The `Scenario_Main.gms` and `Scenario_Variation.gms` then are designed to be overridden in the scenario subfolders (e.g. respectively `\Model\Scenarios\CAR_POLICY` and `\Model\Scenarios\CAR_POLICY\CP_1`). Typically they will specify input values for the simulation (“SIM”) input which differ from the basecase (“BC”) input values.

**Example:**

Folder `Model\Scenarios\FUELTAX` includes an empty `Scenario_Main.gms` file. Folder `Model\Scenarios\FUELTAX\HARMONISATION` includes a `Scenario_Variation.gms` file in which (from year 2010 onwards) diesel fuel excise taxes are set equal to petrol fuel excise taxes. Running `tremove.bat` for scenario FUELTAX and variant HARMONISATION will thus generate a scenario simulation in which (from 2010 onwards) diesel fuel excise taxes are set equal to petrol fuel excise taxes.

## 7. File structure

As the code is designed in a modular way, TREMOVE is made up of a lot of (small) files, each modelling one specific aspect. The following pages guide you through the structure of the TREMOVE code, to indicate where each file plays its role in the overall model structure or to help you find the right file you should modify to adapt TREMOVE to your specific needs.

Every grey rectangle stands for a module that is expanded on another page. The white rectangles stand for a file of GAMS code, the tubes stand for data that is imported.

The structure of the files has been designed in SmartDraw and then imported in this Word document. The original SmartDraw objects contain hyperlinks that allow you to navigate through the TREMOVE structure (every grey box is clickable and will lead you directly to the right – expanded – page). SmartDraw has been used because the viewer is free of any charge. The original SmartDraw objects and the viewer can be obtained from Transport & Mobility Leuven.

*The following pages are based on TREMOVE version 2.44, thus have not yet been updated to v2.7. The v2.44 and v2.7 model structure are very similar however.*

Figure 14: Main

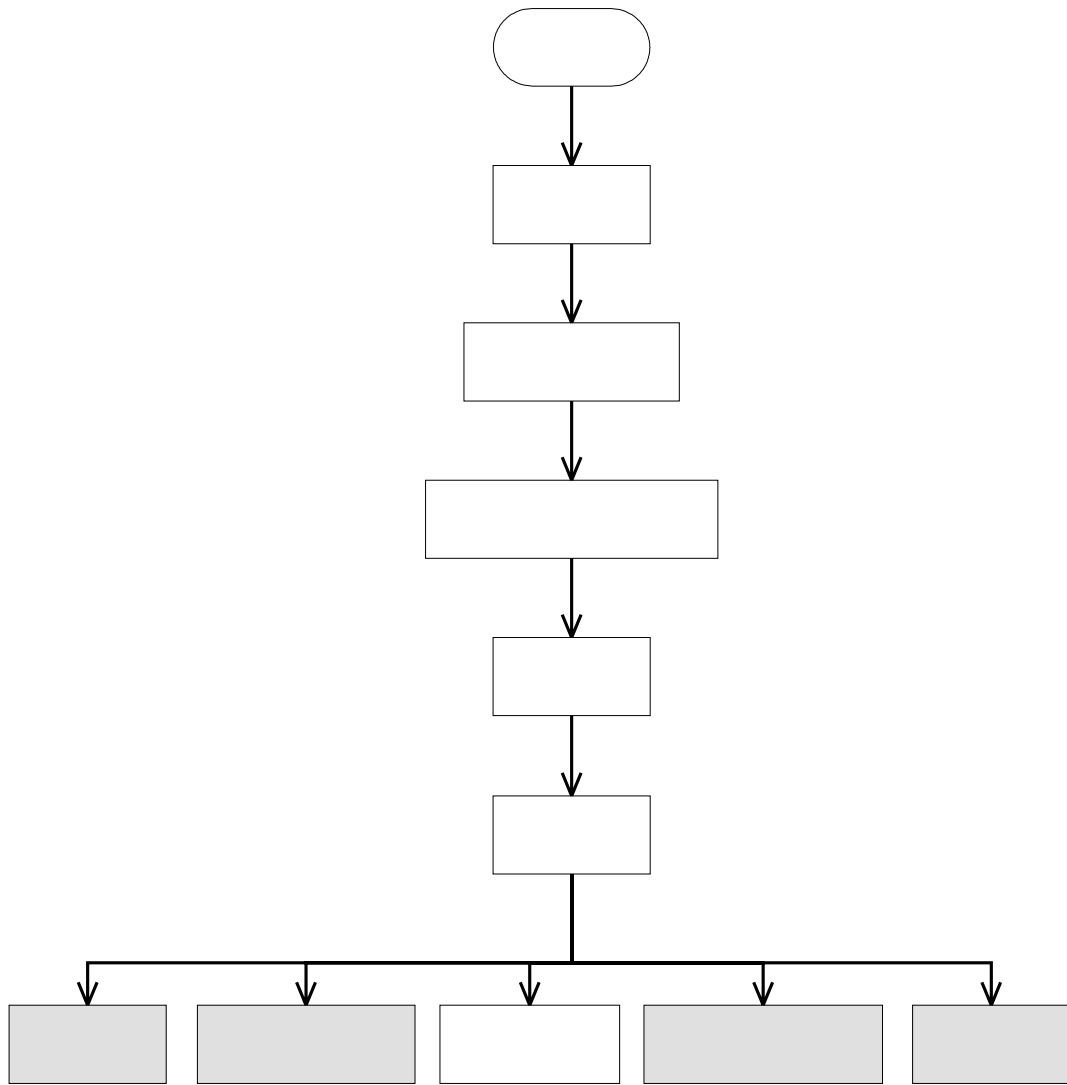
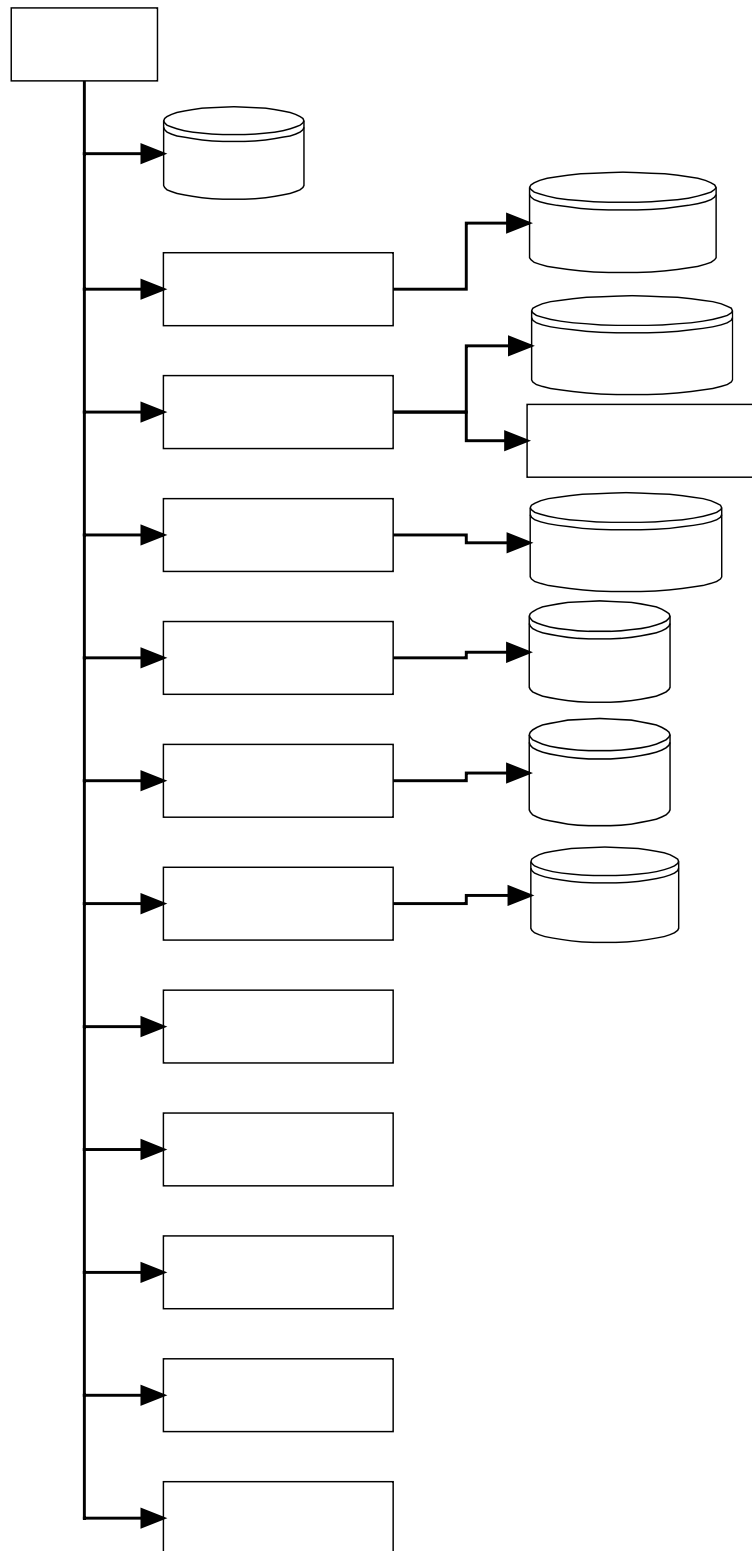


Figure 15: Main >> Define Sets



**define\_**  
**ba**

Figure 16: Main>>Define Parameters

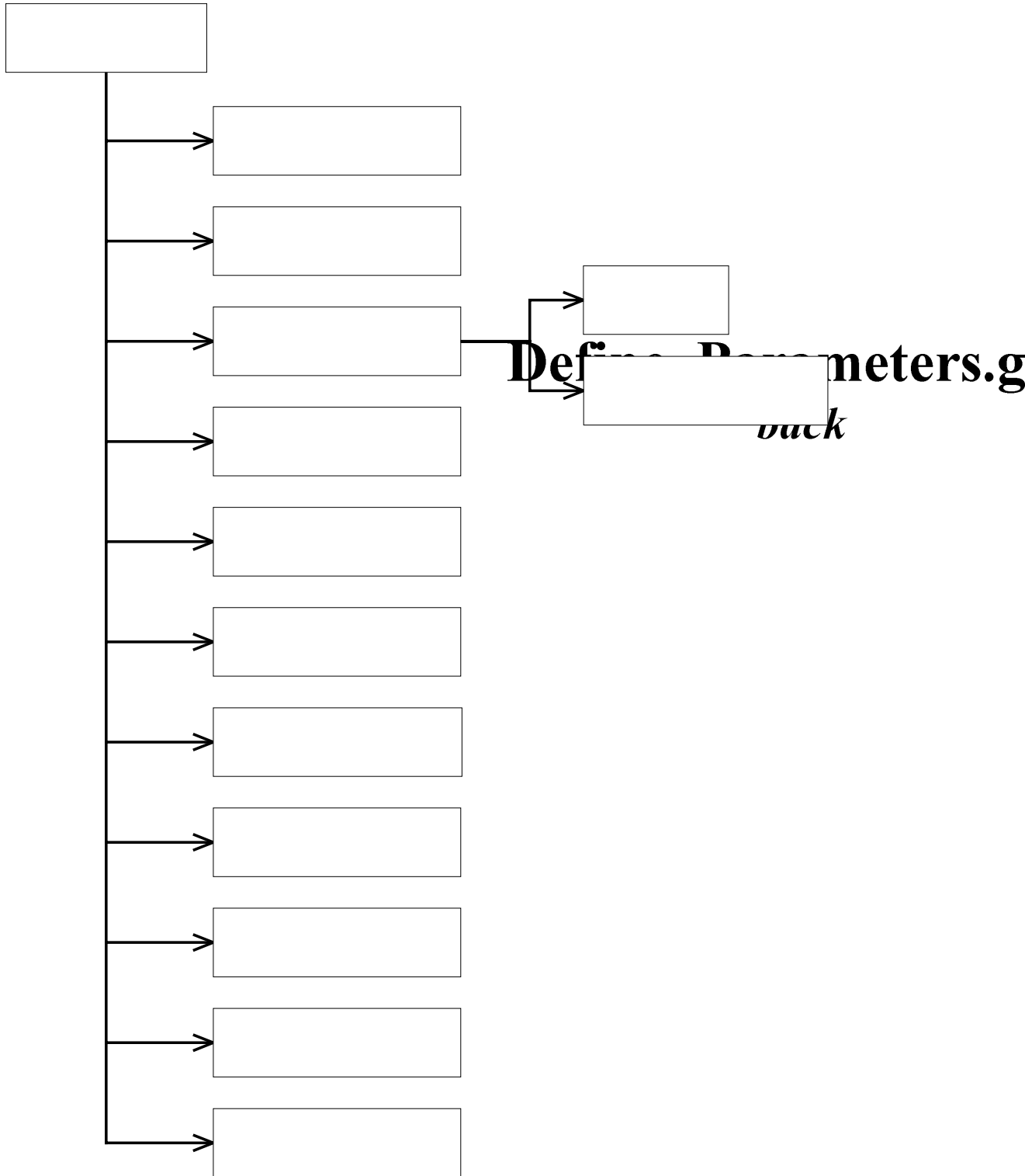


Figure 17: Main>>Calibrate Base\_Case

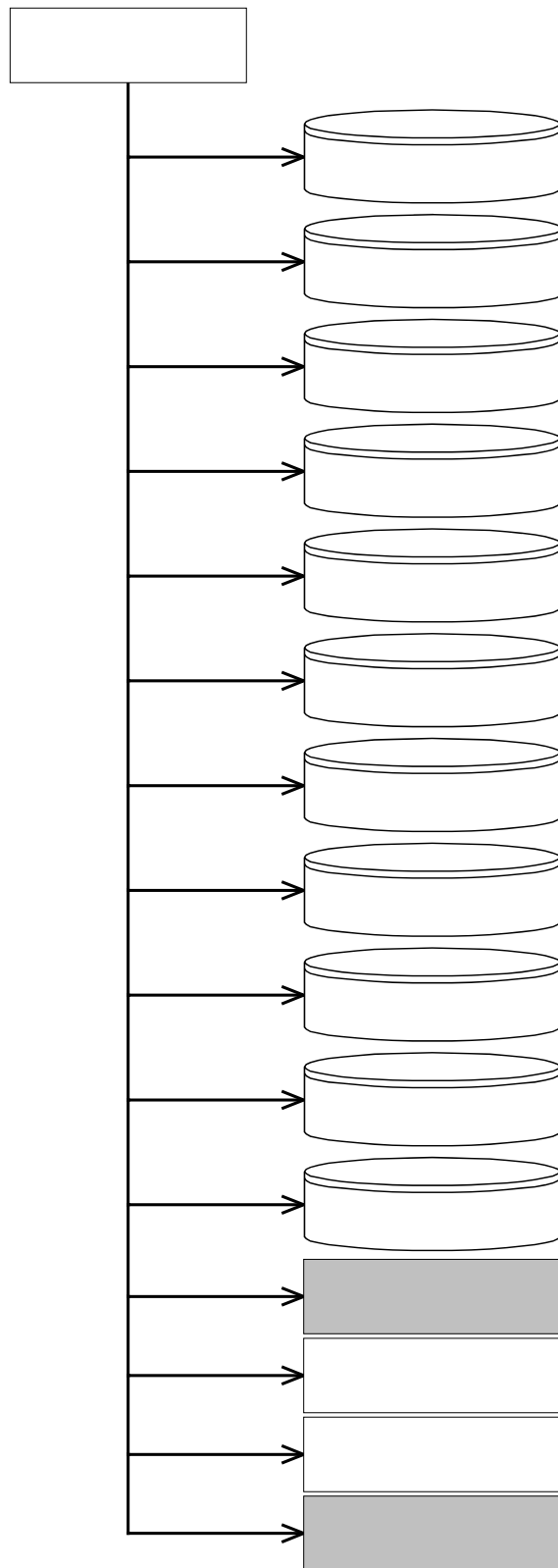
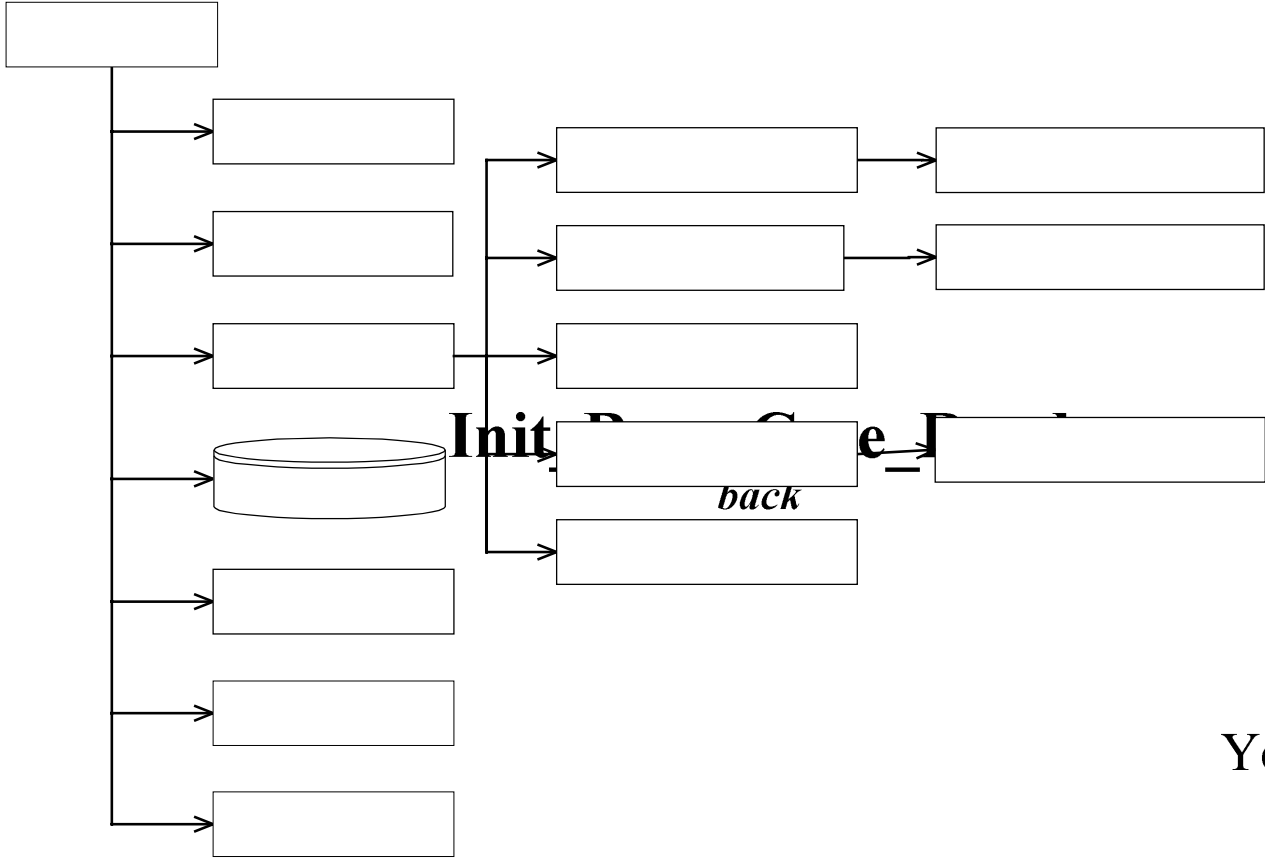


Figure 18: Main>>Calibrate Base\_Case >>Init\_Base\_Case\_Road



Years\_Be

Init\_Missing\_Ro

Init\_Logit\_Li

Figure 19: Main>>Calibrate Base\_Case>>Run TREMOVE and Main>>Run Simulation>>Run TREMOVE

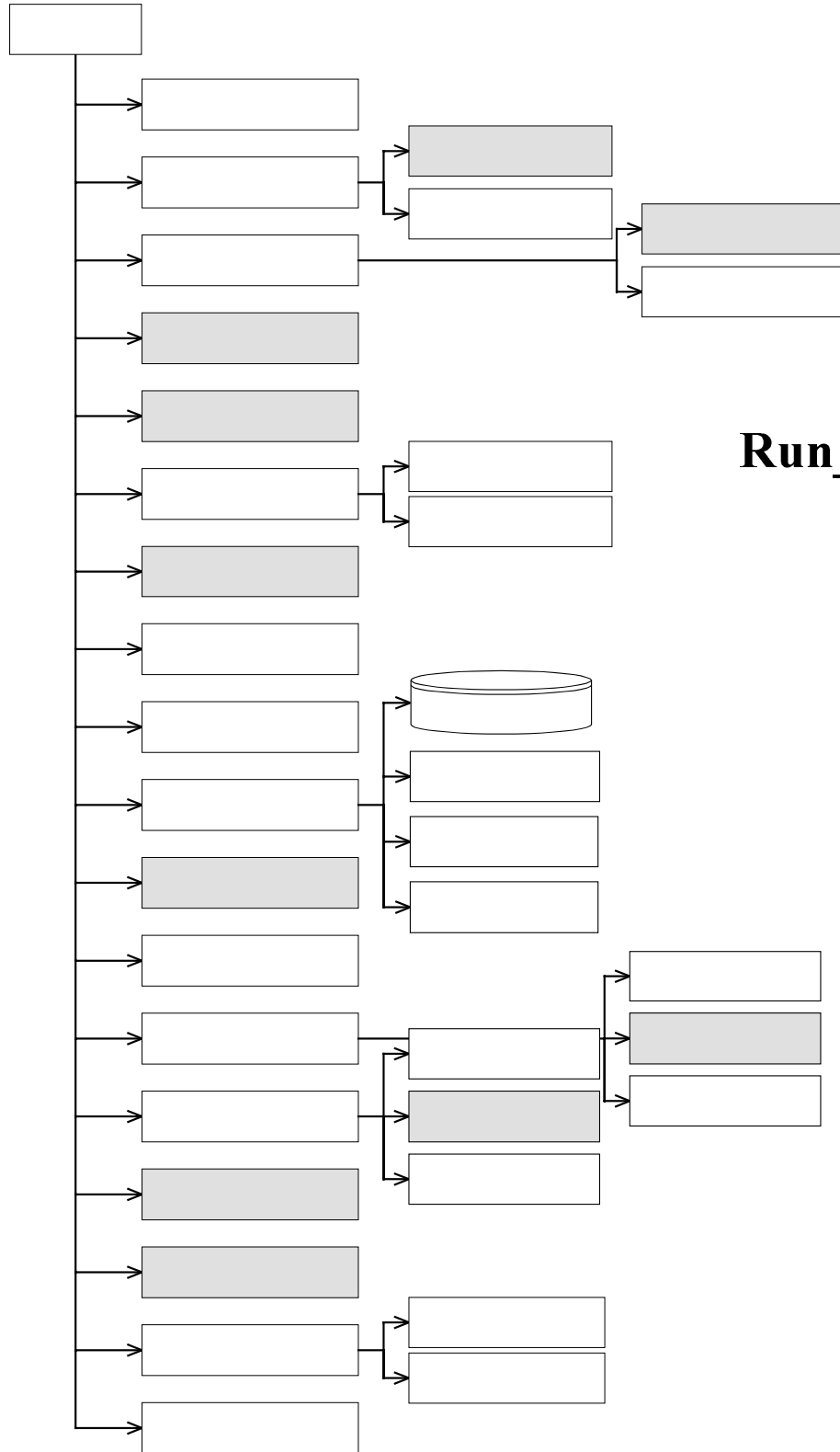


Figure 20: ...Run REMOVE>>Demandcalibinit

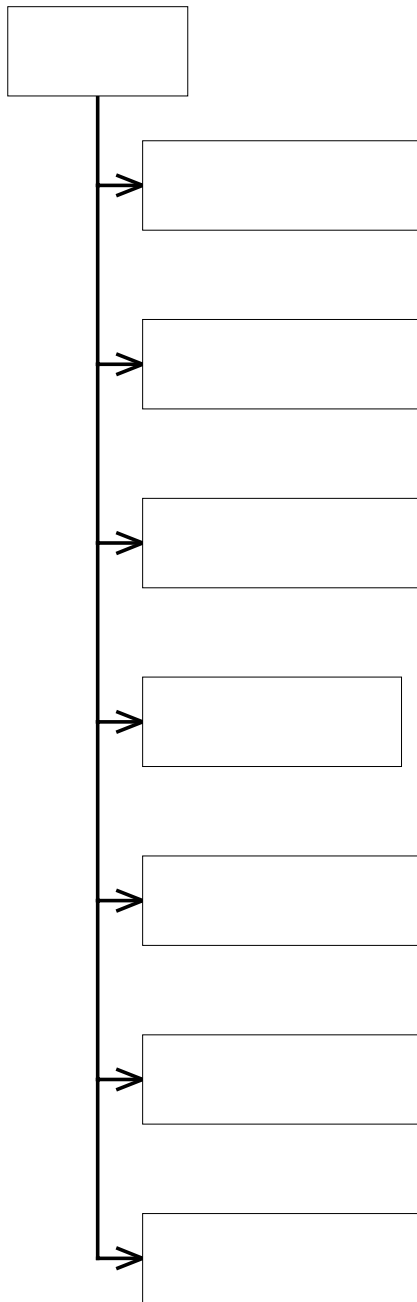


Figure 21: ...Run TREMOVE>>Demandsiminit

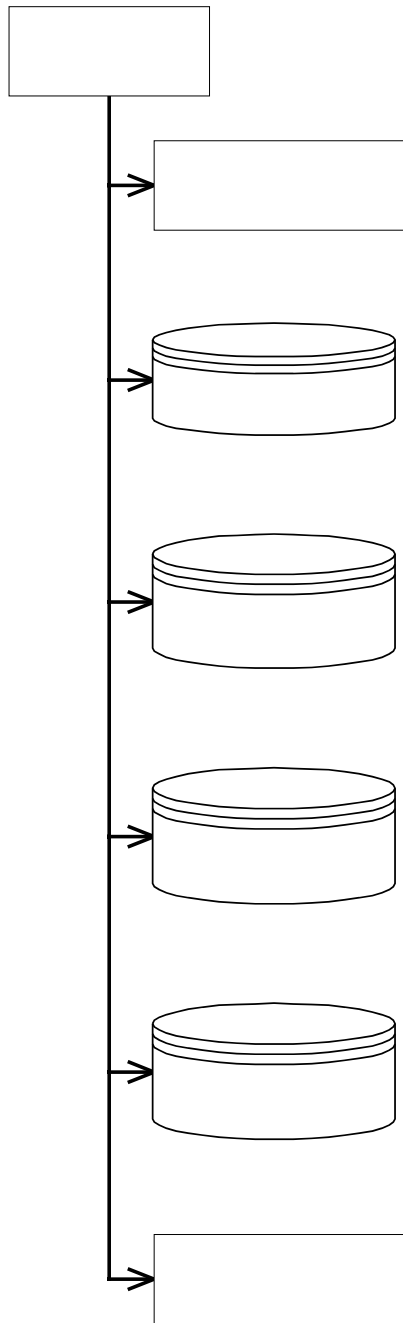


Figure 22: ...Run TREMOVE>>Stock\_Road\_BY

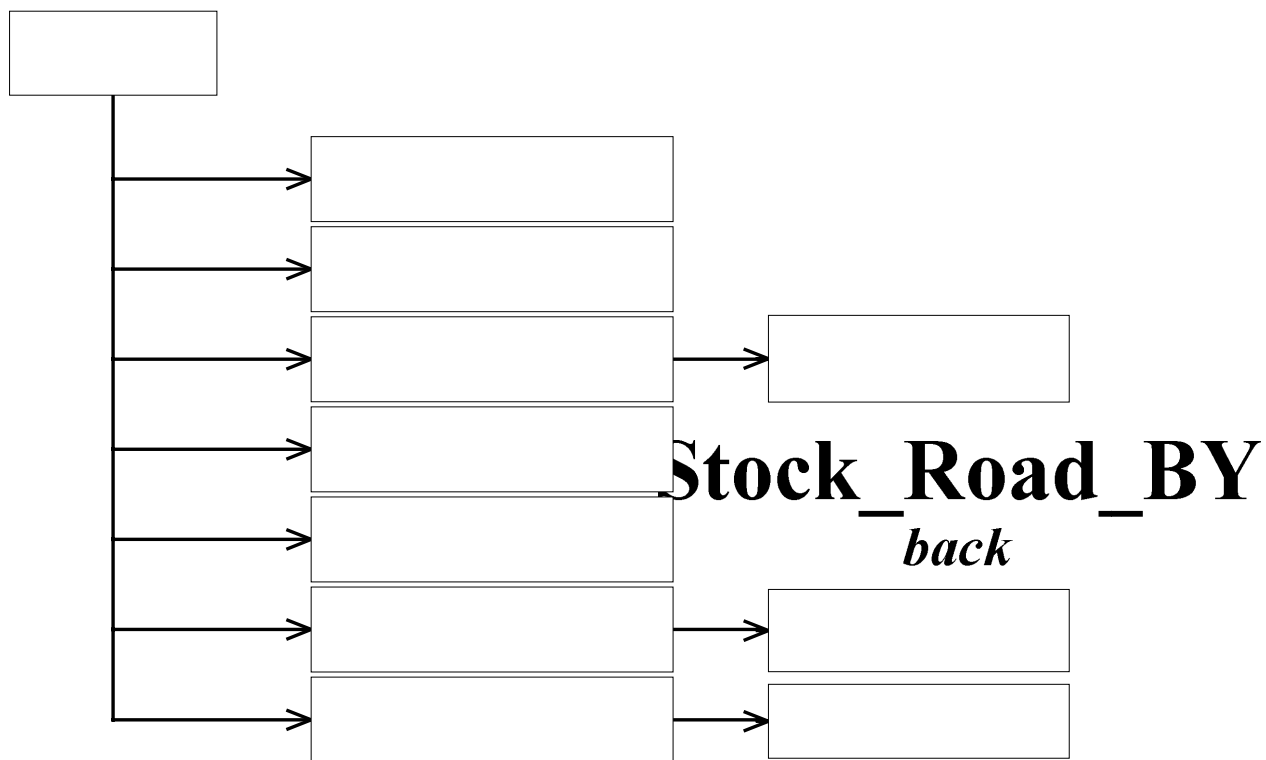


Figure 23: ...Run TREMOVE>>Stock\_Rail\_BY

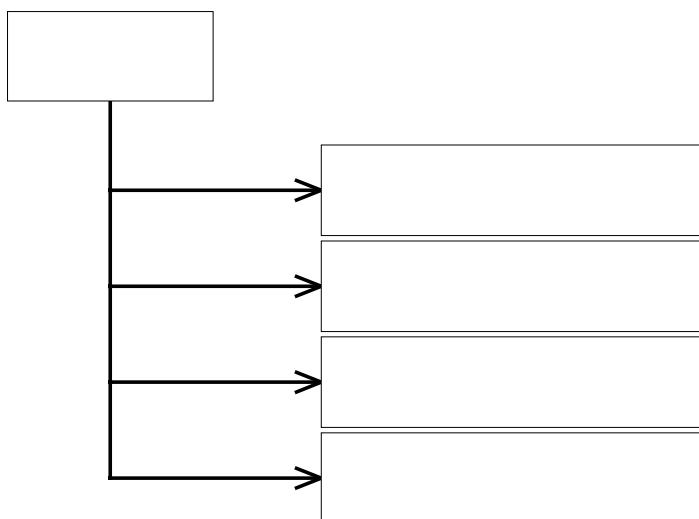


Figure 24: ...Run TREMOVE>>Fuel\_Consumption\_And\_Emissions

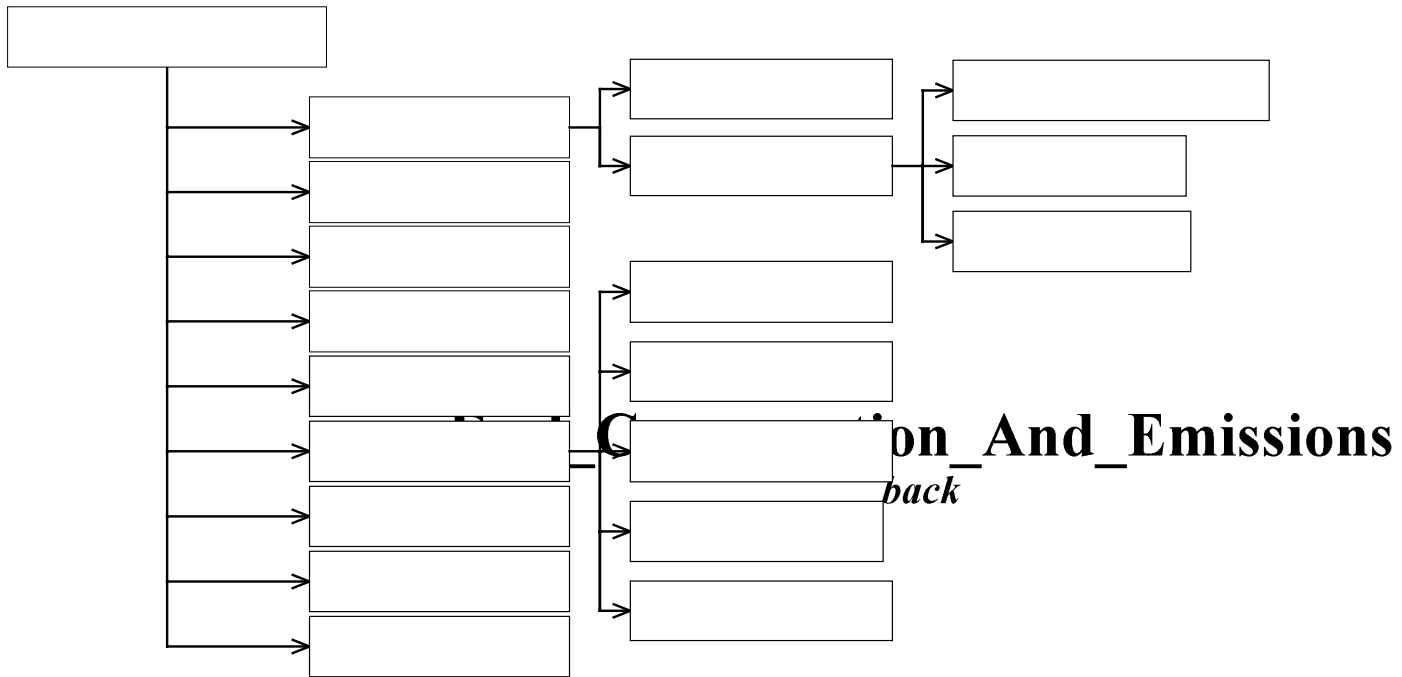
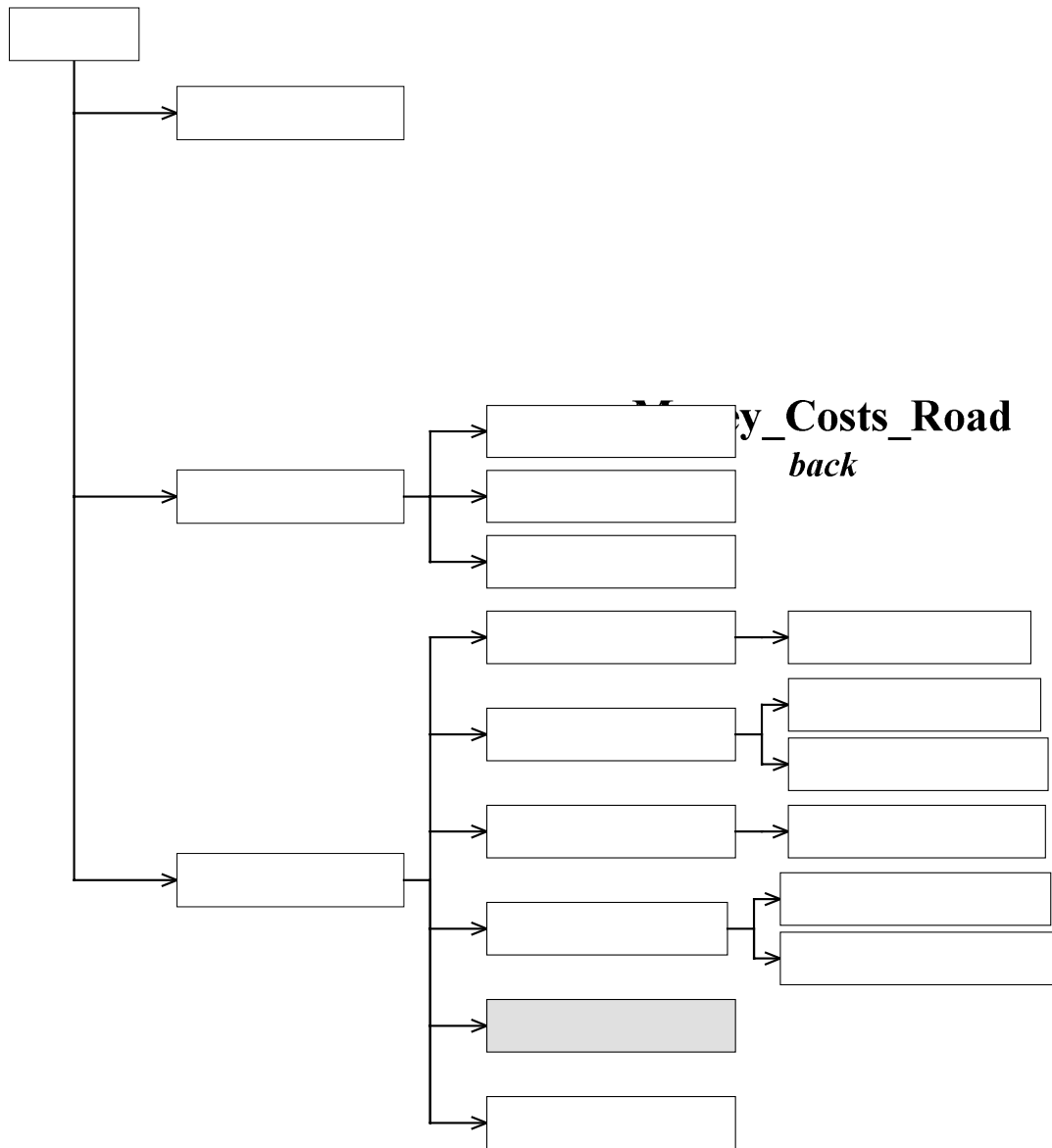


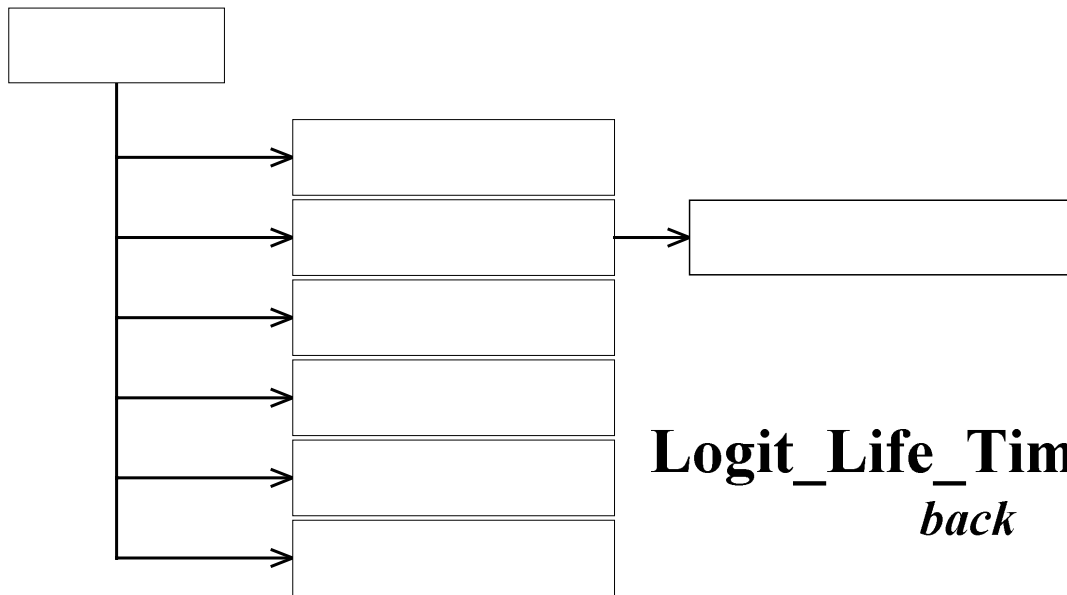
Figure 25: ... Run TREMOVE>>Money\_Cost\_Road



Mon

Mon

Figure 26: ...Run TREMOVE>>Money\_Cost\_Road>>Logit\_Life\_Time\_Costs



**Logit\_Life\_Time\_Costs**  
*back*

Figure 27: ...Run REMOVE>>demandcalibnext

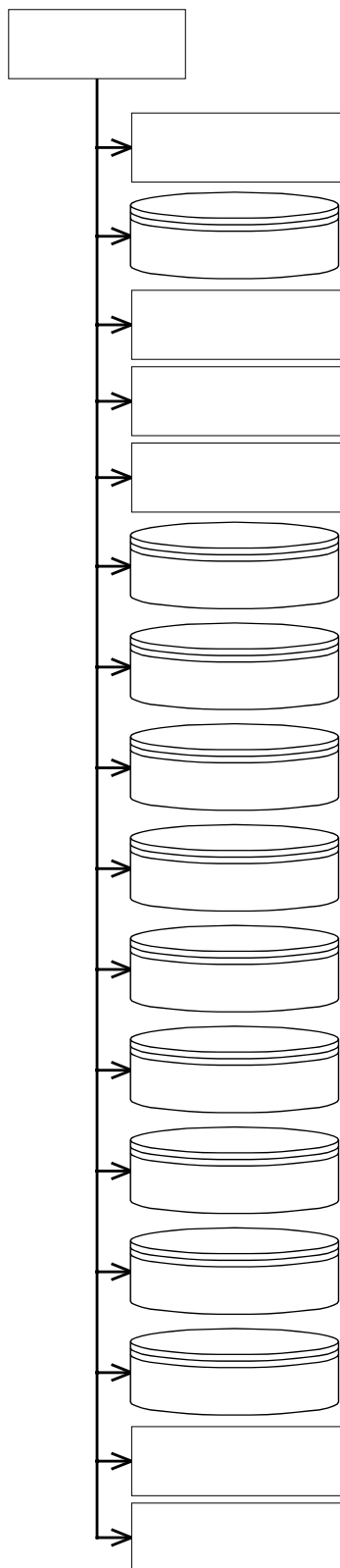


Figure 28: ...Run **TREMOVE**>>demandsimnext

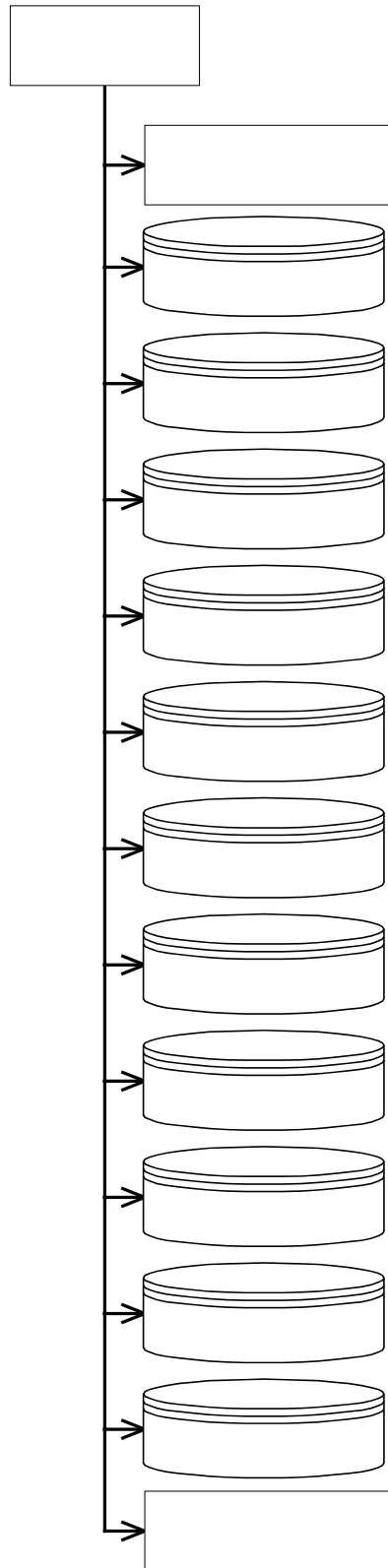
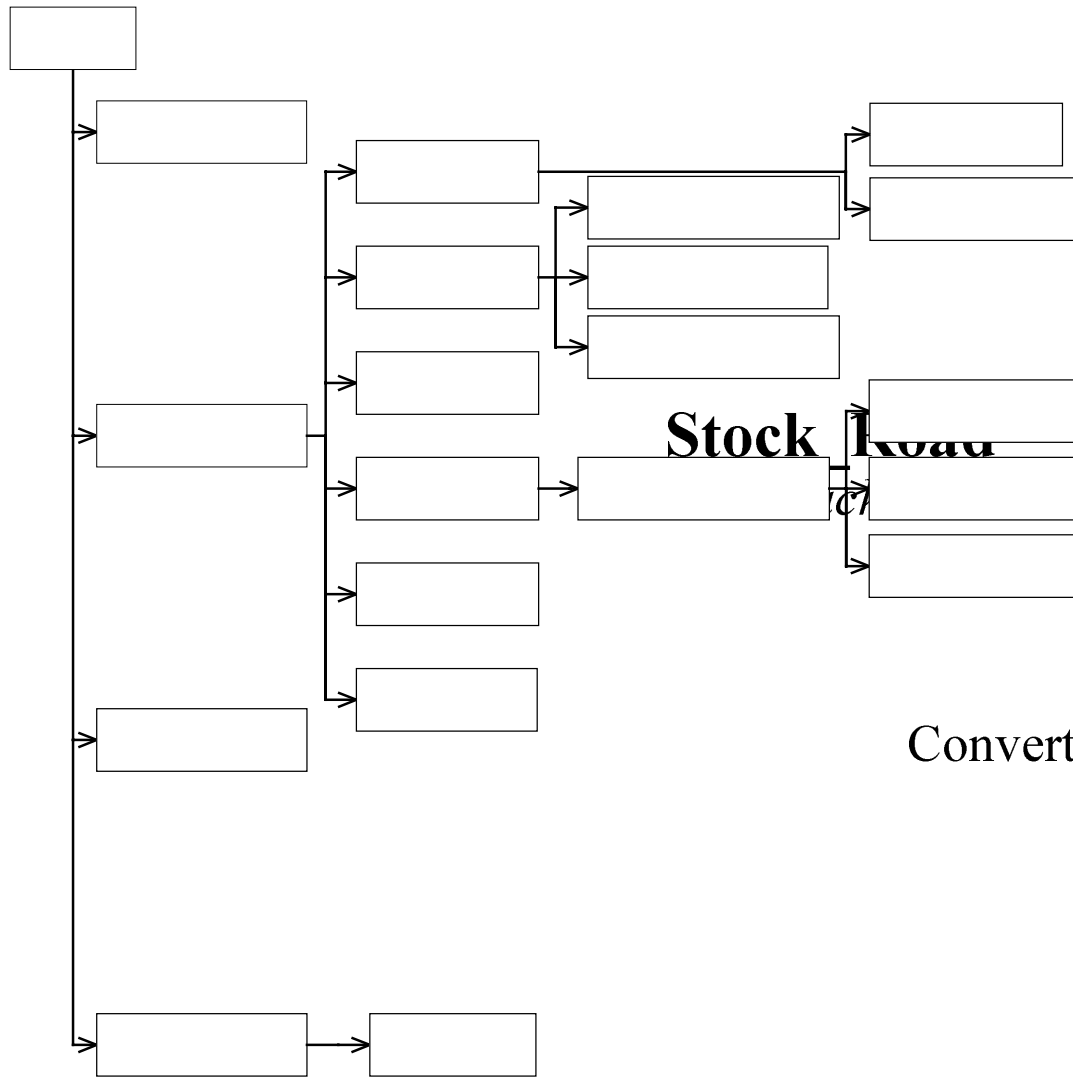


Figure 29: ...Run TREMOVE>>Stock\_Road



Convert\_Technolog

Sales\_Road

Figure 30: ...Run TREMOVE>>Stock\_Rail

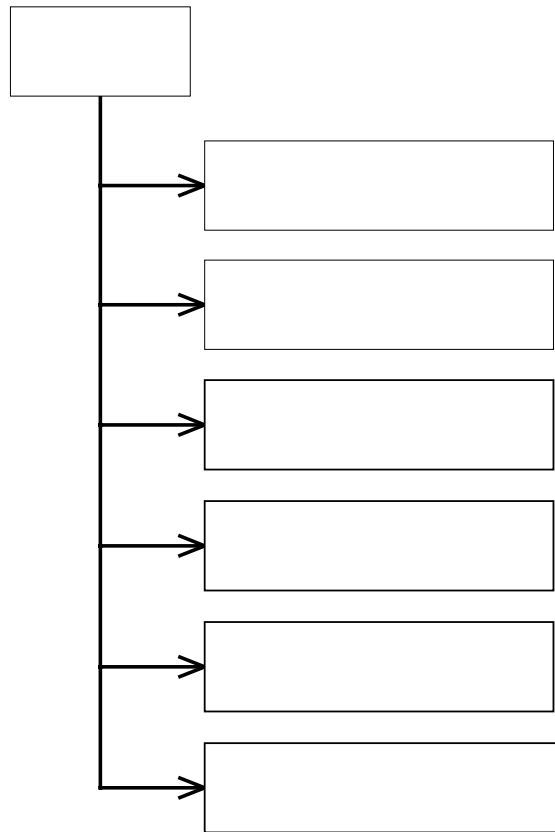
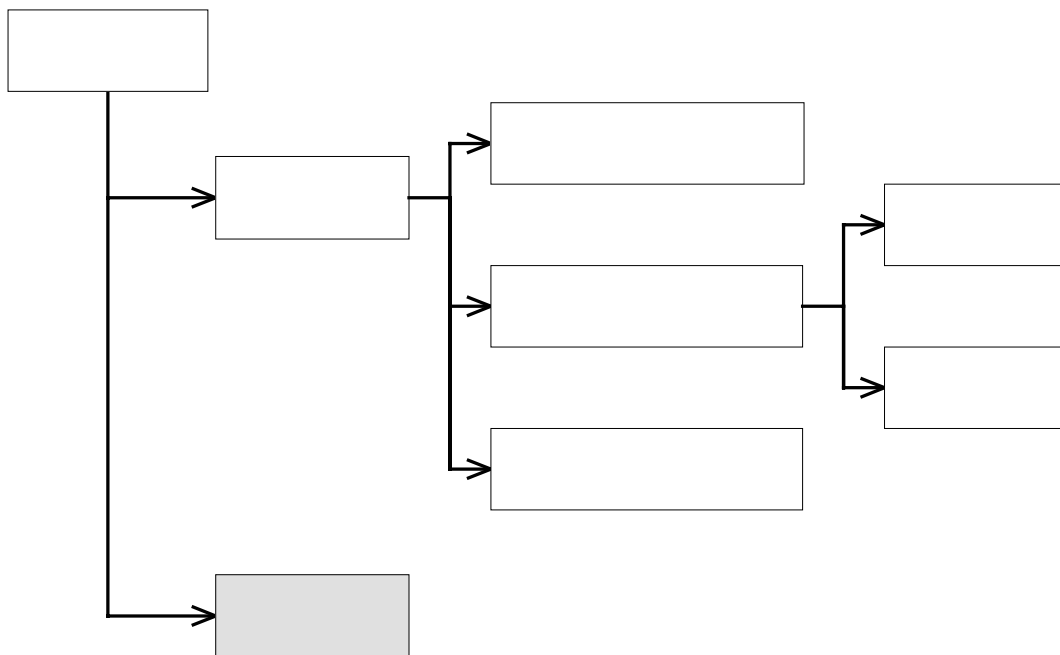


Figure 31: Main>>Run\_Simulation



---

<sup>iii</sup> This kind of modularisation is normally provided in programming languages through functions or procedures. Unfortunately, GAMS does not support this mechanism. The much more primitive `$include` and `$batinclude` constructs offer the only alternative.

<sup>iii</sup> The working directory is the directory in which the `gams` command is invoked