R23 Model: A SAM Based Global Macro CGE Model: USER GUIDE[1]

Scott McDonald and Karen Thierfelder[2]

Addresses for correspondence:

Scott McDonald,
cgemod.org.uk,
14 Moncrieffe Road,
Sheffield,
S7 1HR, UK.
E: jrs.mcdonald@gmail.com

Karen Thierfelder
Department of Economics
US Naval Academy
Annapolis,
Maryland, USA
E: thier@usna.edu

Abstract

This paper is a User Guide for the R23 model. It is an evolving document that reflects the development of the R23 model and its associated support programmes.

Keywords: Computable General Equilibrium; R23; CGE; SAM.

[1] This User Guide is under continuous development. The authors welcome feedback on the Guide, suggestions for additions to the Guide and MOST of all drafts of suggested additions.

[2] Scott McDonald is a Visiting Professor at Humboldt University, Berlin, and Karen Thierfelder is Professor of Economics at the United States Naval Academy.

Table of Contents

1. Introduction

Version 2 of R23 requires the user to operate using GAMSIDE, Excel and GDXViewer. While none of the skills required are difficult the current arrangements do impose a requirement on the user to master certain skills. However, mastery of the required skills will have longer terms benefits for the user since these will allow the user to exercise more direct control over the operation of the model.

The released version of the R23 model, database and support programmes reduce the potential for errors when generating an aggregation from the database, simplify the coding of experiments and enhance access to the results. However, none of these components will make it possible for the user to generate more interesting results; rather they reduce the skills required to use the model and save time when analysing the results. To conduct insightful simulations/experiments and generate interesting results depend on the skills of the users; if users do not have these skills users will need to devote time to their development.

The setting up a variant of the R23 model is all done in the database aggregation phase. So this User Guide starts with database aggregation and then looks at the operation of the model and then at the accessing of results. The details provided are directed at an assumed 'archetypal' user; it is assumed that this user;

- has limited knowledge of GAMS/GAMSIDE and does not want to engage in programming experiments in GAMS;
- will only run one simulation/experiment at a time;
- has a moderate level of skill in Excel; and
- has a strong background in economics (at least taught masters level).

To accommodate the needs of more advance user the User guide also has a number of information boxes that provide guidance on options for more experienced users and/or those wishing to develop the range of simulations they can conduct.

1.2 Required Software

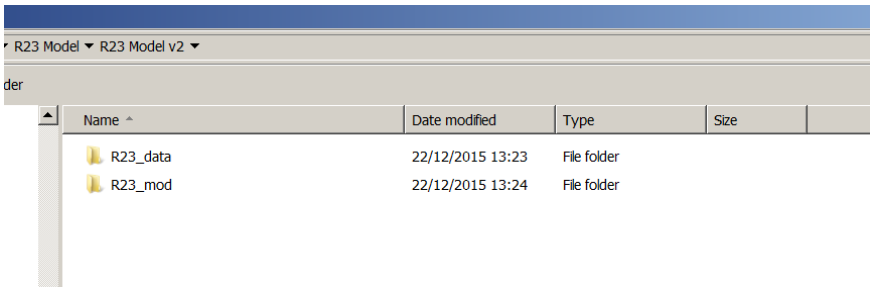
The user is assumed to be using a Windows PC with a recent (post 2015) and licenced[3] version of GAMS (testing was done on GAMS release 24.5.6; later versions should present no problems and many earlier versions, say post 2013, should also present no problems.). The PC also needs to have MS Excel installed with macros enabled; the models have been tested using Excel 2010, 2013 and 2016.

[3] It is possible to run very small aggregations, c 3/4 regions, using the demo version of GAMS.

2. Contents of the R23 Directories

The R23 model and database are supplied as a self-extracting WinZip archive; "R23 Model v2.exe". This will unpack into a single directory, 'R23 Model v2', that contains two sub-directories, 'R23_data' and 'R23_mod' (Figure 2.1).

Figure 2.1 Directory Structure for R23 Model

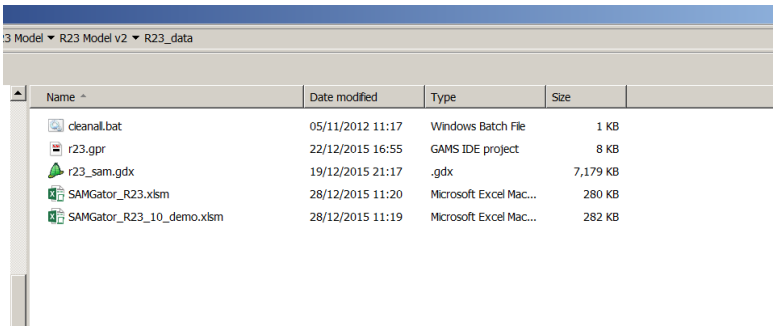


The database and the programmes for setting up the R23 model are in the directory 'R23_data', while the R23 model and associated files for running experiments and collating the results are in the directory 'R23_mod'.

2.1 Database Directory

The 'R23_data' directory contains 4 files: R23 database ('r3_sam.gdx'); 2 versions of the SAMGator programme for R23 (a demo version with a pre-set aggregation – 'SAMGator_R23_10_demo.xlsm' and 'blank' version – 'SAMGator_R23.xlsm') and a batch file to clean up the directory (in the distributed version this is supplied as 'cleanall.**d**at' and needs to be relabelled as 'cleanall.**b**at'). After a first aggregation has been generated a fifth file, which is a GAMS project file, will be added to the directory ('r23.gpr').

Figure 2.2 R23_data Directory



Every time an aggregation is generated additional files will be added to this directory. When the 'cleanall.bat' file is run – double click in windows explorer – surplus files will be deleted.

2.2 Model Directory

3. Preparing a Database for the R23 Model

The R23 database has, currently, 204 regions. While the R23 model will operate with a fully disaggregated version of the database, i.e., every region separately identified, there are several reasons why this may not be an appropriate way to use the model:

1. for any given study there will be a large proportion of the results that contain information not relevant to that study;
2. the model will take longer to solve;
3. some experiments may not solve due to extreme differences in scale – small vv very large economies, e.g., China or India vv Vanuatu; and
4. the functional forms used in all CGE models perform better when 'shares', e.g., value shares of imports to region *r* from region *s*, are not very 'small'.

The database aggregation is implement using (a variant of) the SAMGator programme.[4] The outputs from the aggregation process are recorded in the 'R23_mod' directory.

3.1. SAMGator Principles

An aggregation of the R23 SAM requires that the database is (simultaneously) aggregated in three dimensions, i.e., the intra-regional accounts for all regions are aggregated to reduce the number of trade partners and associated transactions, and inter regional transactions are aggregated so as to accord with the aggregation of the trade accounts for each region. GAMS is an extremely efficient medium for implementing the calculations required to aggregate the GLOBE SAM, but setting up the sets and (set) mappings to control the aggregation is potentially time consuming and error prone. One method for reducing the time costs involved in setting up a new aggregation, and simultaneously reducing the time required to configure the sets and other data needed to implement a version of R23 with a new aggregation, is to use the SAMgator software (PROVIDE, 2004) conditioned to operate with the R23 database.

SAMgator is a Visual Basic for Applications (VBA) programme implemented using Microsoft (MS) Excel. The user declares, describes and defines the new (aggregated) regions in an Excel worksheet, identifies the source and destination data files and then runs GAMS

[4] The SAMGator programme was developed as part of the PROVIDE project (PROVIDE, 2004).

remotely from within the Excel programme. The SAMgator programme generates the set and mapping files, checks to ensure that all the mappings are 'legal' and then generates the GAMS programme file.

SAMGator also aggregates and complies other data that are required to use the R23 model. These include *inter alia* all the sets (normal and mapping) and elasticities required to calibrate the model and a series of templates that facilitate the running of experiments. SAMGator allows the user to create a 1 to 1 mapping, i.e., a fully disaggregated database, so it can be used to generate all the information required to run a fully disaggregated version of the model.

3.1.1 The Mechanics of SAMgator

At the heart of SAMgator is a single (GAMS) equation that aggregates the GTAP database in three dimensions (see below).

```
SAMR(sacp,sacpp,r)      = SUM((ss,ssp,rr)
                              $(map_sac_ss(sacp,ss)
                                $map_sac_ss(sacpp,ssp)
                                $map_r_rr(r,rr)),
                                  SAMG(ss,ssp,rr)) ;
```

The parameter *SAMG*(*ss,ssp,rr*) contains the R23 database where *ss*[5] is the set that defines the row and columns labels for the SAM for each of the *rr* regions. The parameter *SAMR*(*sacp,sacpp,r*) contains the aggregated database where *sac* is the set that defines the row and columns labels for the aggregated (NEW)SAM for each of the *r* regions. The set *map_sac_ss*(*sacp,ss*) defines the members of *ss* that aggregated into *sacp* by the rows of *SAMG*(*ss,ssp,rr*), while the set *map_sac_ss* (*sacpp,ssp*) defines the members of *ssp* that aggregated into *sacpp* by the columns of *SAMG*(*ss,ssp,rr*). (Note that there is only one mapping set *map_sac_ss* with its implementation being solely driven by the sets *ss* and *sac* and their aliases.) The set *map_r_rr*(*r,rr*) defines the aggregation of regions *rr* into *r* for the aggregated (NEW)SAM for each of the *r* regions.[6]

[5] The character p in a set label always indicates an alias for the set identified by the previous characters, e.g., set *ssp* is an alias for *ss*.

[6] The use of two sets for regions ensures that the set ordering chosen by the user (see below) when generating an aggregation is maintained in the output even if the same (apparent) label is used for a region in the aggregation.

3.1 Using SAMgator

The user interface to SAMgator consists of three worksheets 'Control', 'MappingConfig' and 'Sets'. the user only needs to access these three worksheets and henced the discussion here is limited to these three worksheets.

> Other worksheets in the workbook allow the user to make changes to how SAMGator operates, the data it uses and the aggregation programmes code. These features and the VBA code are not locked.

3.1.1 Control

The 'Control' worksheet is used to identify the location of GAMS.EXE, input and output file names, check for errors, write output and to run the aggregation programme. A screen shot of the worksheet is given in Figure 3.1.1.1.

In order to run GAMS from within MS Excel it is necessary for Excel to know the location of the GAMS.EXE file; this information is provided in the 'GAMS.EXE location' box.[7] In order to test that Excel can implement a GAMS programme first click on the 'Pause after execution' box – so it has a tick in the box as in Figure 3.1.1.1 – and then click on the 'Test GAMS' button. A DOS window will appear (Figure 3.1.1.2); if this indicates a normal completion this is confirmation that Excel can trigger GAMS correctly and the user can press any key to continue.

The Data Exchange (GDX) input file needs must be identified; in this case the file is called `'R23_sam.gdx'`; the file name is **fixed**.[8] Then name the output file; in this case `'R23_data.gdx'`, which is the default file name, which does not require the user to edit the R23 model file. **If you change this name, you will need to edit the R23 model file and the basic user mode of R23 will not operate.** And finally the user should provide a name for the GAMS programme file; in this case `'R23aggr.gms'`.[9] SAMGator is not sensitive to

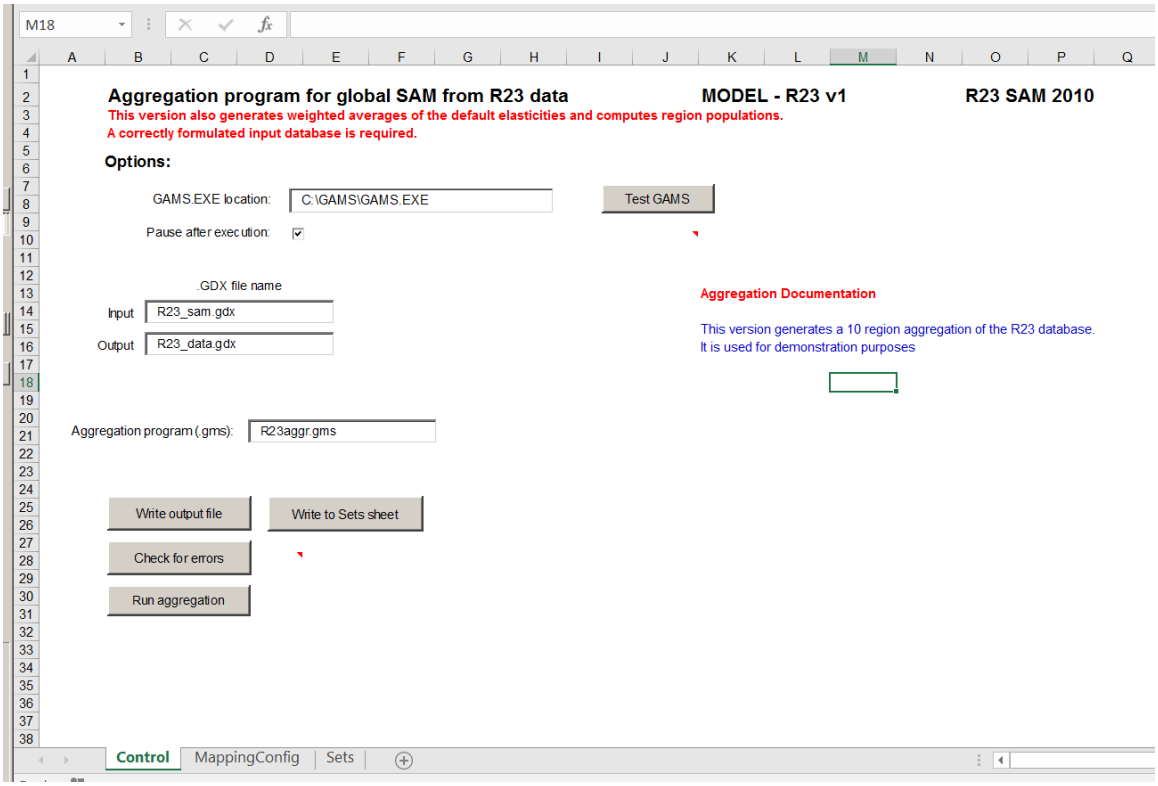
[7] With some versions of GAMS spaces in the paths can lead to problems. Note how the screen shot indicates that GAMS.exe is stored in a GAMS directory on the C drive and not in the 'Program Files' directory that is the default destination when installing GAMS; this is usually an advantage with the Excel based programmes included with this model.

[8] At present there is only one version of the R23 database so this name should not be changed.

[9] Giving the aggregation file a unique name is one way of preserving the details of the aggregation.

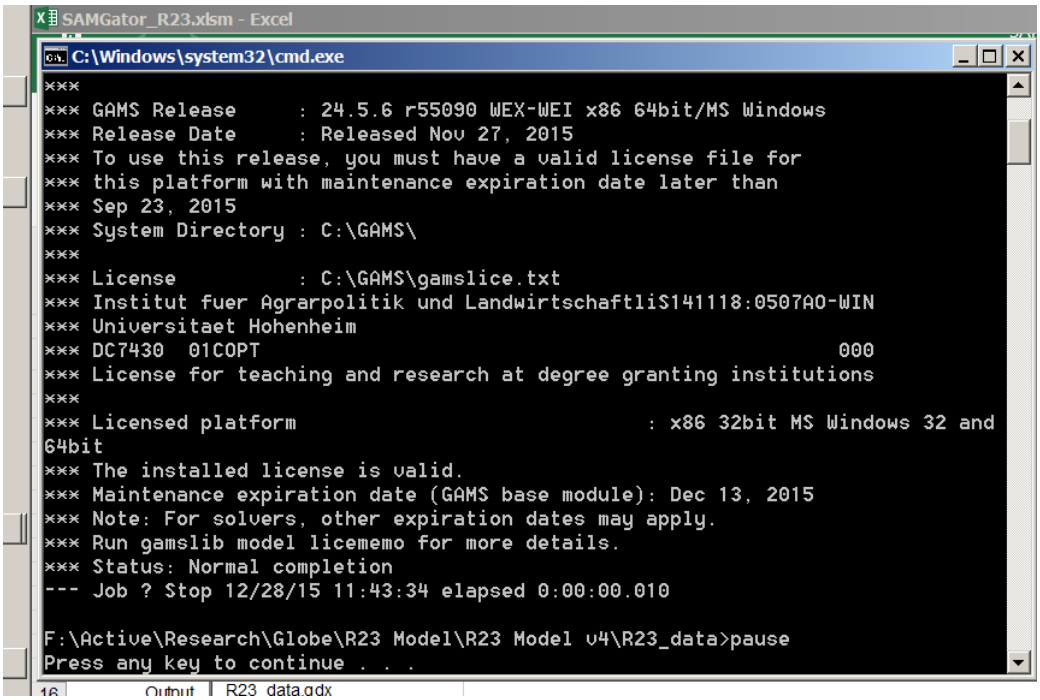
the choice of naming conventions, but the setup of the R23 model and database follow certain conventions so these names should not be changed unless the user has good reasons.

Figure 3.1.1.1 Control Worksheet for SAMgator



It may be useful to preserve the actual aggregation generated. This can be achieved by giving the 'Output' file a unique name, e.g., 'R23_data_****.gdx'. If this is option is chosen note that the companion Excel file generated will also need to be saved, outwith SAMGator, with a unique name, e.g., 'R23_data_****.xlsx'. It will also be necessary to edit the model file to read the data from these files (see box below).

Figure 3.1.1.2 Test GAMS

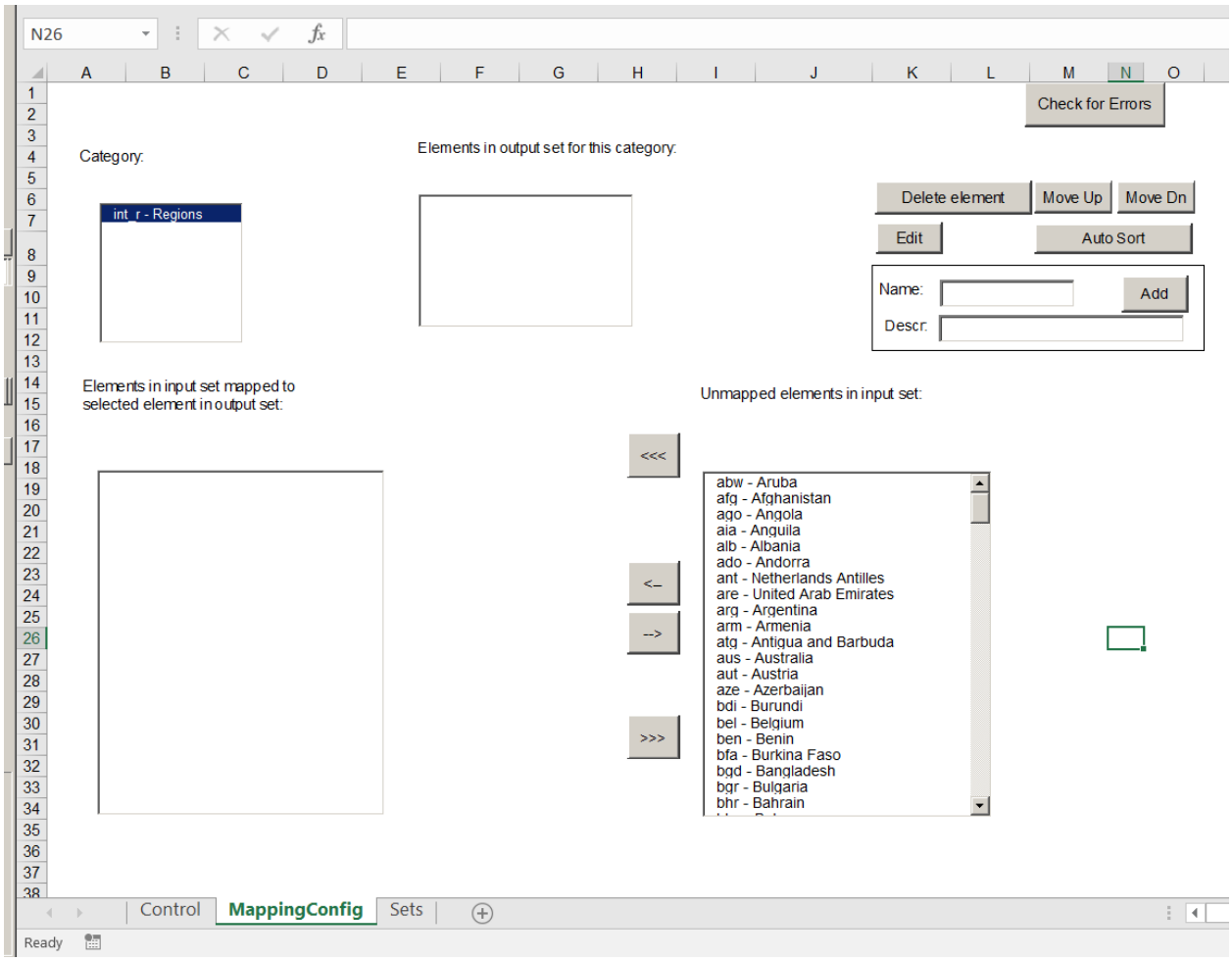


3.1.2 MappingConfig

The role of the 'MappingConfig' worksheet is to generate the sets for the aggregated SAM and set up the mappings, which will control the aggregation. The process has a number of safeguards built in to ensure the mappings are legitimate and that there are no errors. Although users may choose to adapt a previous aggregation the description given here assumes that the user starts with a version of SAMgator that contains no aggregation specific set or mapping data.

Figure 3.1.2.1 illustrates how the 'MappingConfig' worksheet should look when there is no aggregation specific set or mapping data; the only difference between this image and the one seen when opening SAMgator is that the 'int_r – Regions' line in the 'Category' box (top left) has been highlighted. The user needs to create sets and mappings for the regions.

Figure 3.1.2.1 Mapping and Configuration Worksheet 1



The steps undertaken to declare and define aggregate sets and to generate the mapping file are as follows.

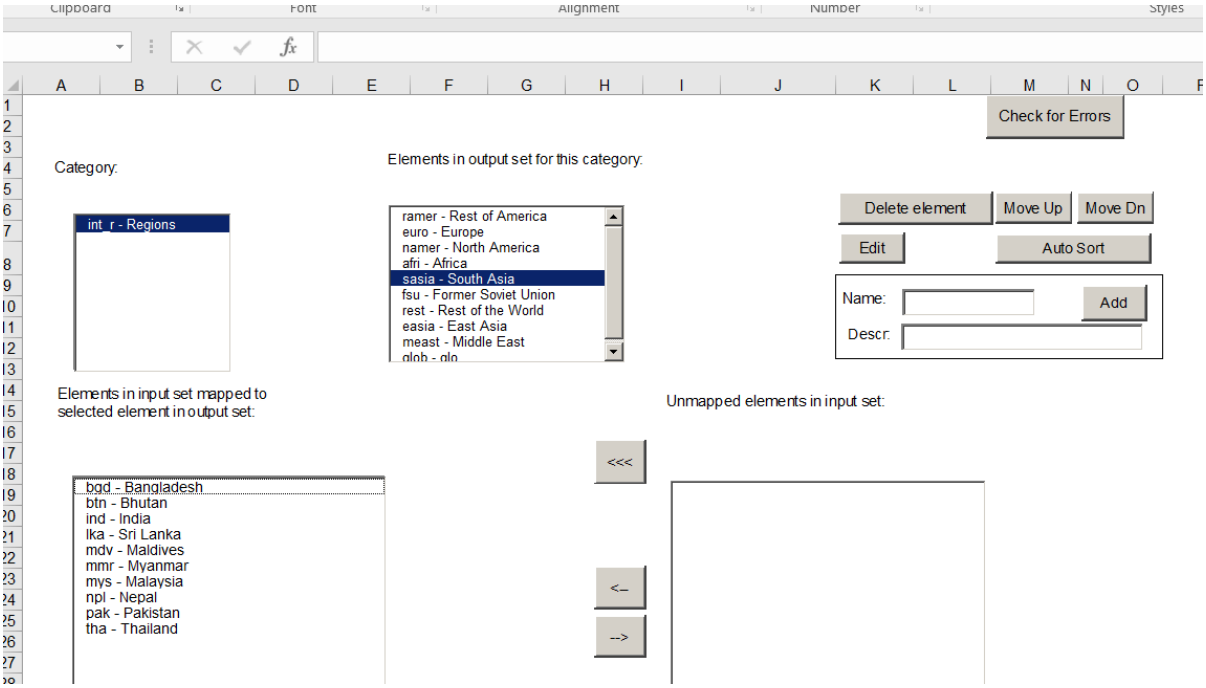
1. Select a category: choose regions in the ‘Category’ box (top left).
2. Declare (and describe) an aggregate set element: type the name in the ‘Name’ box and below it a description in the ‘Descr’ box and then click the ‘Add’ button. This element will be a member of an output set for which each element will appear with its description in the ‘Elements in output set for this category’ box.
    - Note that SAMgator will use the sets and mappings to generate automatically multiple sets by PREFIXING letters to each element of the output set.

3. Select elements in the input set that are to be mapped to the selected member of the output set: select an element in the 'Elements in output set for this category' box and then select those elements that are to be mapped to that output set element from the 'Unmapped elements of input set' box (bottom right) by clicking on each in turn (NB: the SHIFT and Ctrl click options do not operate). Click on the <- button to define the mapping. Note how the selected elements of the input set are moved to the 'Elements in input set mapped to selected element in output set' bow (bottom left).
    - The <- button moves selected elements in the unmapped element box to the mapped elements box; the -> button moves selected elements in the mapped element box to the unmapped elements box; the <<< button moves all elements in the unmapped element box to the mapped elements box; >>> button moves all elements in the mapped element box to the unmapped elements box.
    - Figure 3.1.2.2 illustrates that for the category 'int_r – Regions' the elements 'bgd', 'btn', 'ind', 'lka', 'mdv', 'mmr', 'mys', 'npl', 'pak' and 'tha' of the input set are mapped to the element 'sasia – South Aisa' of the output set.
4. Continue adding elements to the output set until all the elements in the input set are mapped to elements in the output set. Note how each element in the input set can only be mapped to one element in the output set – thereby avoiding 'double counting'. The error checking also checks to avoid duplicates in the output set.
5. Ordering of the output set: as each new element of an output set is declared it is added to the end of the list of members of the output set for that category. This ordering defines the set ordering that will be used by GAMS. The order of elements within an output set can be altered by selecting elements in the 'Elements in output set for this category' box and using the 'Move Up' and/or 'Move Down' buttons.[10]

[10] Note that GAMS lists set members in the order that they were declared in the programme. If the same name, say 'abc', is used in 2 different sets this can create ordering problems for the user: assume that the order wanted for reports is the order of members in the second set declared, but because 'abc' is in the first set declared it will be the first member of the second set.

6. Deleting an output set element: select the element to delete in 'Elements in output set for this category' box and click on the 'Delete element' button.
7. Select another category and complete steps 2 to 7 as appropriate until output sets and mapping sets for all three categories have been generated.
8. Verify that the 'Names' and 'descriptions' ('Descr') used for the elements in the output set are appropriate for the project. If any are not use the 'Edit' button to revise the details. (NB: the labels and descriptions used will be carried over to the model and used in the results; while it is easy to go back and re run the aggregation and simulations it is time consuming, so time spent making decisions when defining the output set is more efficient.)

Figure 3.1.2.2 Mapping and Configuration Worksheet 2



3.1.3 Sets

The worksheet sets provides a listing of the set names and descriptions required by the R23 model. This is only used for information since the information is saved to the 'R23_data.gdx' file.

> The information in the 'Sets' worksheet is saved in an Excel file ('R23_data.xlsm') generated when the aggregation is generated. However, if the user wishes to code simulations without using that Excel file this worksheet may be convenient.

3.2 Aggregation Guidelines

Some general guidelines for aggregations are given below. In the main these are common sense, but the note about the treatment of the Globe region (*glo/glob*) is critical.

3.2.1 Account Aggregation

The aggregation of accounts in the global SAM is overwhelmingly a matter of the context for which the aggregation is being compiled; the rationale should be based on economic reasoning and/or common sense. Aggregations within categories should typically avoid mappings that are difficult to rationalise, e.g., aggregating developing and developed regions when the focus is on a group of developing regions.

3.2.2 Treatment of the Globe region when aggregating

The Globe region serves a special purpose in the R23 model since it handles all transactions for which full bi-lateral data are absent. Consequently, Globe must **never** be part of an aggregate region and **MUST** be retained as a separate region in **ALL** implementations of the R23 model.

The element 'glo' in the R23 database must map to the element 'glob' in the aggregated database.

3.3 Output from SAMGator

The outputs from the SAMGator programme are written to the 'R23_mod' directory as two files: 'R23_data.gdx' and 'R23_data.xlsm'.

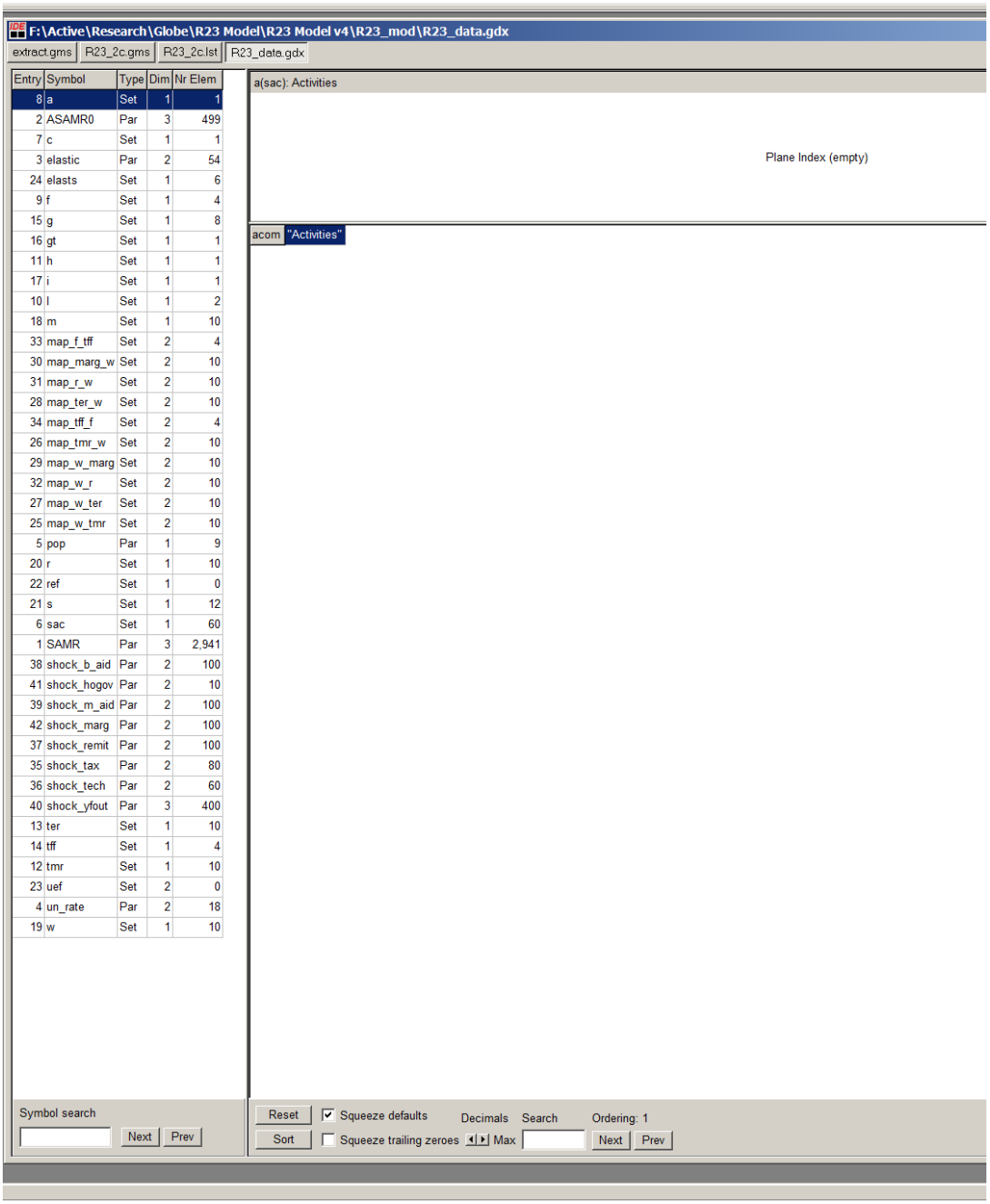
3.3.1 R23_data.gdx'

This is a GAMS data exchange file that contains the model database, default elasticities, sets and maps. The contents can be viewed using GAMSIDE or GDXViewer (see Figure 3.3.1 for the GAMSIDE view).

The contents of the 'R23_data.gdx' file include all the SAM (transactions) data, default values for elasticities, templates for shocks, sets and maps required to run the R23 model. It is a good idea to explore the information in the database before using the data to run any

simulations. Some simple questions are given below to serve as a (first) guide to exploring the model database.

Figure 3.3.1 R23_data.gdx: GAMSIDE view



Entry	Symbol	Type	Dim	Nr Elem
8	a	Set	1	1
2	ASAMR0	Par	3	499
7	c	Set	1	1
3	elastic	Par	2	54
24	elasts	Set	1	6
9	f	Set	1	4
15	g	Set	1	8
16	gt	Set	1	1
11	h	Set	1	1
17	i	Set	1	1
10	l	Set	1	2
18	m	Set	1	10
33	map_f_tff	Set	2	4
30	map_marg_w	Set	2	10
31	map_r_w	Set	2	10
28	map_ter_w	Set	2	10
34	map_tff_f	Set	2	4
26	map_tmr_w	Set	2	10
29	map_w_marg	Set	2	10
32	map_w_r	Set	2	10
27	map_w_ter	Set	2	10
25	map_w_tmr	Set	2	10
5	pop	Par	1	9
20	r	Set	1	10
22	ref	Set	1	0
21	s	Set	1	12
6	sac	Set	1	60
1	SAMR	Par	3	2,941
38	shock_b_aid	Par	2	100
41	shock_hogov	Par	2	10
39	shock_m_aid	Par	2	100
42	shock_marg	Par	2	100
37	shock_remit	Par	2	100
35	shock_tax	Par	2	80
36	shock_tech	Par	2	60
40	shock_yfout	Par	3	400
13	ter	Set	1	10
14	tff	Set	1	4
12	tmr	Set	1	10
23	uef	Set	2	0
4	un_rate	Par	2	18
19	w	Set	1	10

What information is recorded in the following symbols (note the symbol search box in the lower left corner):

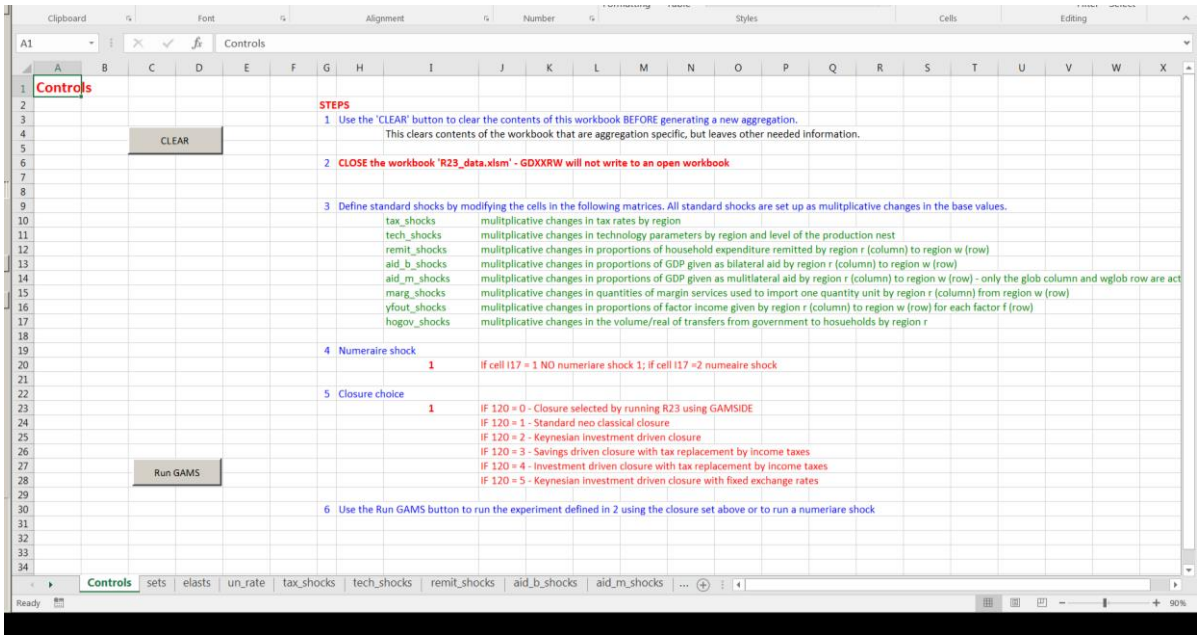
1. '*sac*';

2. '*r*' and '*w*';
3. '*map_r_w*';
4. '*SAMR*';
5. '*elastc*';
6. '*shock_tax*'; and
7. '*un_rate*';

3.3.2 R23_data.xlsm'

The file 'R23_data.xlsm' contains information that can be used to condition the model and run simple experiments without having to programme using GAMSIDE; however, running simulations using GAMSIDE greatly increases the options available to the analyst with, initially, very little additional expertise.

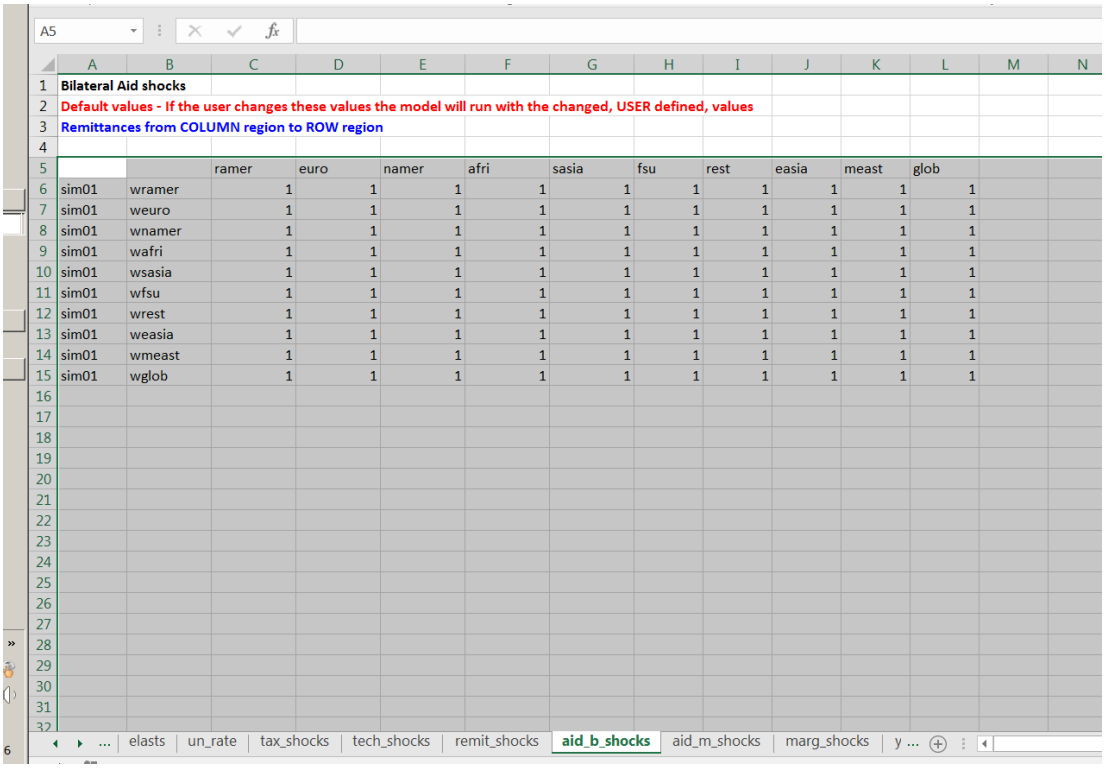
Figure 3.3.2 R23_data.xlsm File



The file 'R23_data.xlsm' contains 11 worksheets. These and their purposes are:

1. 'controls' – contains various controls that allow the user to condition the model (see below);
2. 'sets' – reports the sets used in the model (for information only);
3. 'elasts' – 'default' elasticities used by the model;
4. 'tax_shocks' – 'default' multiplicative changes in tax rates by region;

5. ‘tech_shocks’ - ‘default’ multiplicative changes in technology parameters by region and level of the production nest;
6. ‘remit_shocks’ - ‘default’ multiplicative changes in proportions of household expenditure remitted by region r (column) to region w (row);
7. ‘aid_b_shocks’ - ‘default’ multiplicative changes in proportions of GDP given as bilateral aid by region r (column) to region w (row);



	A	B	C	D	E	F	G	H	I	J	K	L
1	**Bilateral Aid shocks**											
2	**Default values - If the user changes these values the model will run with the changed, USER defined, values**											
3	**Remittances from COLUMN region to ROW region**											
4												
5			ramer	euro	namer	afri	sasia	fsu	rest	easia	meast	glob
6	sim01	wramer	1	1	1	1	1	1	1	1	1	1
7	sim01	weuro	1	1	1	1	1	1	1	1	1	1
8	sim01	wnamer	1	1	1	1	1	1	1	1	1	1
9	sim01	wafri	1	1	1	1	1	1	1	1	1	1
10	sim01	wsasia	1	1	1	1	1	1	1	1	1	1
11	sim01	wfsu	1	1	1	1	1	1	1	1	1	1
12	sim01	wrest	1	1	1	1	1	1	1	1	1	1
13	sim01	weasia	1	1	1	1	1	1	1	1	1	1
14	sim01	wmeast	1	1	1	1	1	1	1	1	1	1
15	sim01	wglob	1	1	1	1	1	1	1	1	1	1

8. ‘aid_m_shocks’ - ‘default’ multiplicative changes in proportions of GDP given as mulitlateral aid by region r (column) to region w (row) - only the glob column and wglob row are active;
9. ‘marg_shocks’ - ‘default’ multiplicative changes in quantities of margin services used to import one quantity unit by region r (column) from region w (row);
10. ‘yfout_shocks’ - ‘default’ multiplicative changes in proportions of factor income given by region r (column) to region w (row) for each factor f (row);
11. ‘hogov_shocks’ - ‘default’ multiplicative changes in the volume/real of transfers from government to households by region r.

The 'layout' worksheet is hidden. If the user wants to extend the information provided to the model and/or simulations from this Excel file, s/he needs to modify the 'layout' worksheet.

A review of the worksheets will demonstrate that all the worksheets with '-shocks' in their names, e.g., 'tax_shocks', only report the value of ONE, i.e., in the context of the model experiment file's code there are no shocks.

The worksheet 'elasts' reports the default elasticities used by the model. The user should review these elasticities and make ACTIVE decisions about the appropriateness of the elasticities. The model will operate even if these values are unchanged.

The worksheet 'un-rate' only reports the values 'EPS'. The effect these assignments is that the model assumes full employment for all factors reported in the columns. The user should make ACTIVE decisions about the appropriateness of the assumptions made by these settings. The model will operate even if these values are unchanged.

4 The R23 model

The R23 model is set up to run using one gms file, one inc(lude) file, one gdx file and one Excel file. The directory also contains another gms file; this is for more advancer uses/users. The files are in the `R23_mod` directory with the names `R23.gms`, `R23_expt_std.inc` `R23_data.gdx` and `R23_data.xlsm`.

> The file `R23_expt.gms` is intended for more advanced uses where the user wishes to use the `save` and `restart` facilities (see more on `Save` and `Restart` below). If the user wants to use this option, the experiment file in the file `R23.gms` (line 2888) needs to be commented out.

Only one file needs to be accessed and there is no need to change anything in that file to verify that the model calibrates and solves correctly.

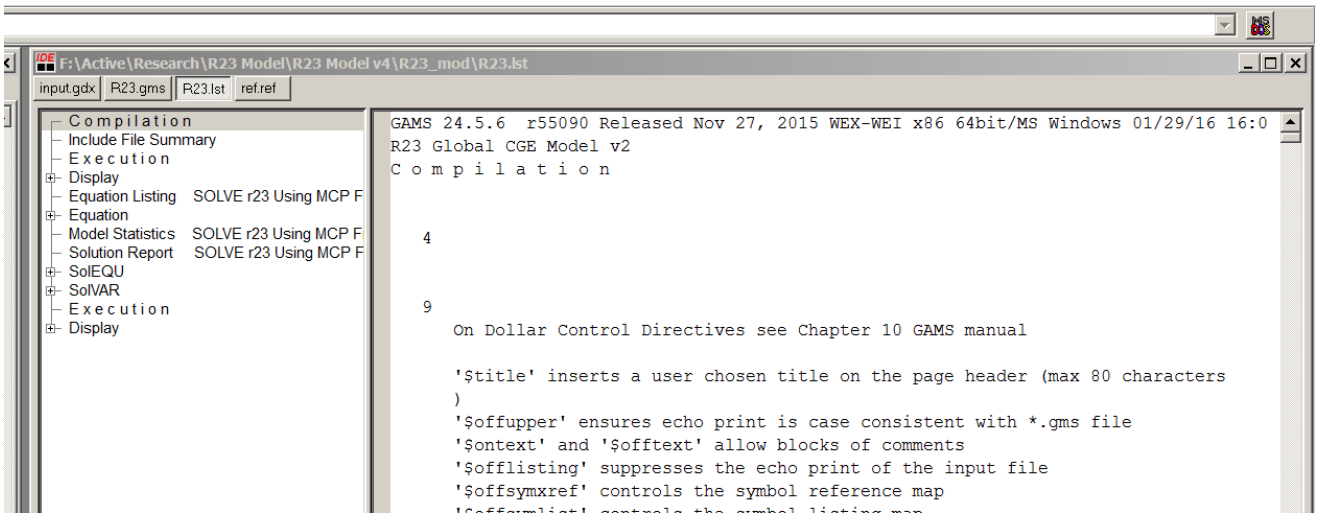
4.1 Model Calibration

In GAMSIDE change the project by going to the directory `R23_mod` and opening the project file (`File > Project > Open Project`) `r23.gpr`, and then open the file `R23.gms`. Run the programme and check that the output is correct (as described below).

> **NB: This process for calibration checking assumes that no changes have been made to the Excel file before first running the model.**

When a GAMS programme is run a file - **.lst (where ** is the name of the GAMS programme file) – is generated, together with an indexing file (**.lxi). The 'lst' file contains all the standard GAMS output and displays any information that it is instructed to do so; the 'lxi' file makes it easy to navigate (See Figure 4.1.1).

Figure 4.1.1 List/Output File in GAMSIDE



4.1.2 Model Checks

There are a standard set of model checks that should be followed in turn.

1. Slack variables: All the slack variables should equal zero, or very nearly zero. Search for '`var walras`', '`var kapworsys`', '`var globeslack`' and '`var globeslack3`' – all should be zero. (Note: if the version of GAMS used has indexing for the list file select SolVar and the slacks are reported at the end of the list of variables.)
2. Check the Left hand sides: Search for 'LHS', then after finding the first occurrence of 'LHS' search for '***'. If any equations are incorrectly specified they are identified. (Note: if the version of GAMS used has indexing for the list file select SolEQU and then the first named equation, this will move the cursor to the first equation.)
3. Check data replication: First check the Macro SAM: search for '`ASAMG2CHK`' – all the values should equal 1; then search for and check '`DIFFASAMG2`' and '`CNTASAMG2`' – these should be zeros or close to zero. Second check the Micro SAM: search for and check '`DIFFSAMG2`'and '`CNTSAMG2`'– these should be zeros or close to zero. (Note: if the version of GAMS used has indexing for the list file select the second DISPLAY index.)
4. Check the numéraire: Go to the 'Controls' worksheet in the Excel workbook 'R23_data.xlsm' and change the number in the cell I20. The recommendation

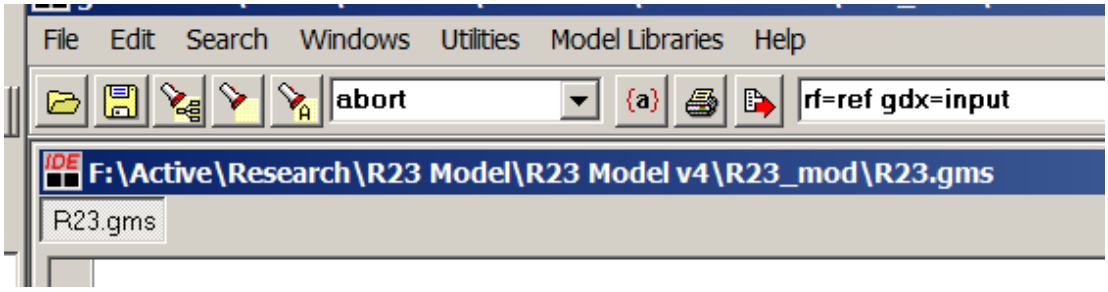
is to change I20 to 2, i.e., double all prices relative to the numéraire (the etxt below assumes 2 has been chosen) and rerun the model. Then, check the Macro SAM: search for '`ASAMG2CHK`' – all the values should equal 2; note that '`DIFFASAMG2`' and '`CNTASAMG2`' are no longer meaningful and therefore the micro SAM calculations have not been implemented.

If the model passes all these checks the model will (usually) be correct.

4.2 R23 Standard Outputs

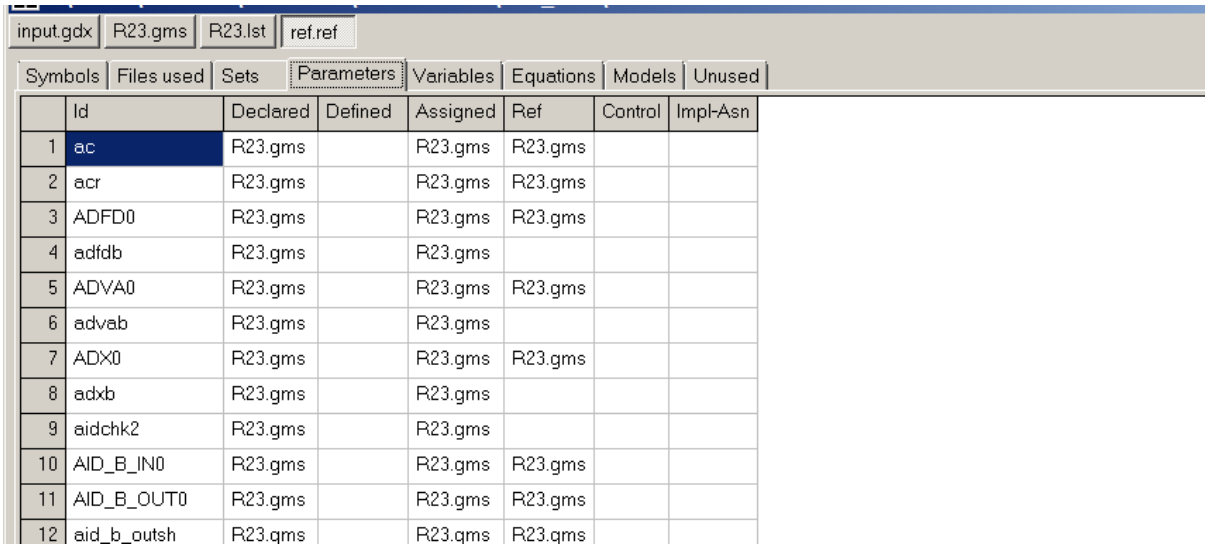
The programme file – 'R23.gms' – is supplied with two instructions in the command line – `rf=ref gdx=input` (see Figure 4.2.1) – **if these instructions are not included they should be added**. The first generates a 'reference file' (Figure 4.2.2) that reports a comprehensive reference map with all the files, sets, parameters, variables, equations, models and Unused symbols (the symbols sheet contains a full list of all symbols), generated when the current GAMS programme is run, and the second, produces a GDX file (Figure 4.2.3) that reports the values for all sets, parameters, variables and equations.

Figure 4.2.1 Command Line in GAMSIDE



The reference file is an invaluable guide to the GAMS programme. It provides information on where each symbol in the model is declared, defined, assigned, ref(erenced), used as a control (sets) and Impl(emented)-Asn (equations). If the user clicks on, say, the column 'assigned' for the row 'AID_B_IN0' the file R23.gms will be opened at the line in the programme where 'AID_B_IN0' is assigned (line 1189). Thus the reference file allows the user to find information about the programme rapidly and without extensive searches.

Figure 4.2.2 Reference File in GAMSIDE

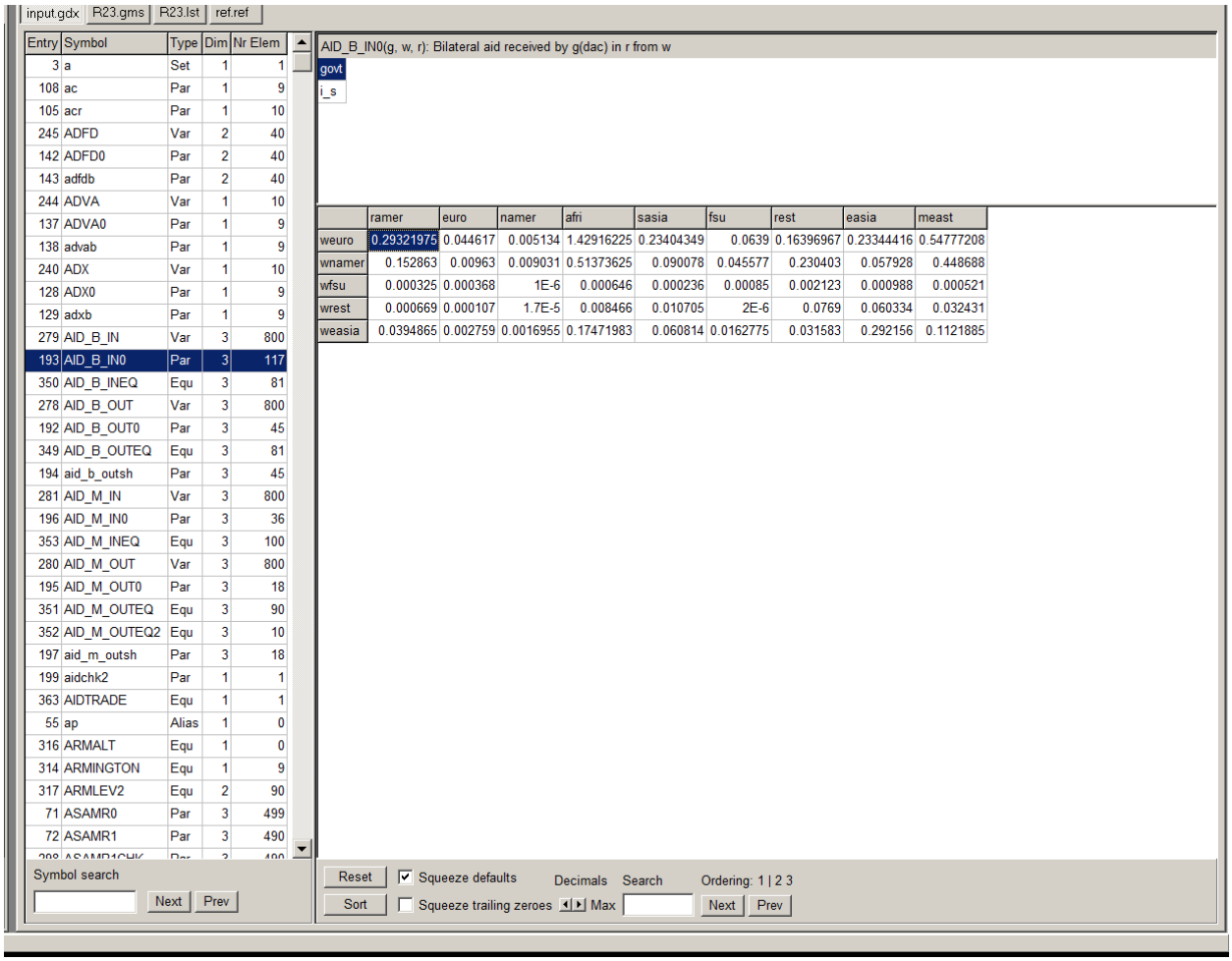


	Id	Declared	Defined	Assigned	Ref	Control	Impl-Asn
1	ac	R23.gms		R23.gms	R23.gms		
2	acr	R23.gms		R23.gms	R23.gms		
3	ADFD0	R23.gms		R23.gms	R23.gms		
4	adfdb	R23.gms		R23.gms			
5	ADVA0	R23.gms		R23.gms	R23.gms		
6	advab	R23.gms		R23.gms			
7	ADX0	R23.gms		R23.gms	R23.gms		
8	adxb	R23.gms		R23.gms			
9	aidchk2	R23.gms		R23.gms			
10	AID_B_IN0	R23.gms		R23.gms	R23.gms		
11	AID_B_OUT0	R23.gms		R23.gms	R23.gms		
12	aid_b_outsh	R23.gms		R23.gms	R23.gms		

The second file produced, the `input.gdx` file provides the user with an easily search compendium of ALL the symbols generated by the programme. This has been used with the R23 model to replace the large number of display statements traditionally found in many GAMS programme; not only is it easy to search and extract data from, it also has the advantage of not requiring the coder to include explicit display instructions in the code.

The user can also use the `input.gdx` file to verify the items (1) and (3) and to carry out the checks for a numeraire shock – item (4) of the calibration checks.

Figure 4.2.3 INPUT.GDX File in GAMSIDE



Entry	Symbol	Type	Dim	Nr Elem
3	a	Set	1	1
108	ac	Par	1	9
105	acr	Par	1	10
245	ADFD	Var	2	40
142	ADFD0	Par	2	40
143	adfdb	Par	2	40
244	ADVA	Var	1	10
137	ADVA0	Par	1	9
138	advab	Par	1	9
240	ADX	Var	1	10
128	ADX0	Par	1	9
129	adxb	Par	1	9
279	AID_B_IN	Var	3	800
193	AID_B_IN0	Par	3	117
350	AID_B_INEQ	Equ	3	81
278	AID_B_OUT	Var	3	800
192	AID_B_OUT0	Par	3	45
349	AID_B_OUTEQ	Equ	3	81
194	aid_b_outsh	Par	3	45
281	AID_M_IN	Var	3	800
196	AID_M_IN0	Par	3	36
353	AID_M_INEQ	Equ	3	100
280	AID_M_OUT	Var	3	800
195	AID_M_OUT0	Par	3	18
351	AID_M_OUTEQ	Equ	3	90
352	AID_M_OUTEQ2	Equ	3	10
197	aid_m_outsh	Par	3	18
199	aidchk2	Par	1	1
363	AIDTRADE	Equ	1	1
55	ap	Alias	1	0
316	ARMALT	Equ	1	0
314	ARMINGTON	Equ	1	9
317	ARMLEV2	Equ	2	90
71	ASAMR0	Par	3	499
72	ASAMR1	Par	3	490

	ramer	euro	namer	afri	sasia	fsu	rest	easia	meast
weuro	0.29321975	0.044617	0.005134	1.42916225	0.23404349	0.0639	0.16396967	0.23344416	0.54777208
wnamer	0.152863	0.00963	0.009031	0.51373625	0.090078	0.045577	0.230403	0.057928	0.448688
wfsu	0.000325	0.000368	1E-6	0.000646	0.000236	0.00085	0.002123	0.000988	0.000521
wrest	0.000669	0.000107	1.7E-5	0.008466	0.010705	2E-6	0.0769	0.060334	0.032431
weasia	0.0394865	0.002759	0.0016955	0.17471983	0.060814	0.0162775	0.031583	0.292156	0.1121885

4.3 Running a Simple Experiment

The Excel file R23_data.xlsm contains a number of worksheets that allow the user to run simple experiments without the user needing to change the GAMS code. The model has pre-coded experiments for changes in

1. tax rates by region;
2. technology parameters by region and level of the production nest;
3. proportions of household expenditure remitted by region *r* to region *w*;
4. proportions of GDP given as bilateral aid by region *r* to region *w*;
5. proportions of GDP given as multilateral aid by region *r* to region *w*;
6. quantities of margin services used to import one quantity unit by region *r* from region *w*;
7. proportions of factor income given by region *r* to region *w* for each factor *f*; and

8. the volume/real of transfers from government to households by region *r*.

BEFORE RUNNING ANY EXPERIMENTS, THE USER SHOULD REVIEW THE BASE VALUES OF ALL THE INSTRUMENTS IN THE DATABASE. THIS CAN BE READILY DONE USING THE FILE `INPUT.GDX`.

The simple experiments are all coded using multiplicative changes to the base values of the parameters that will be shocked. The version of R23_data.xlsm generated by the SAMGator progamme sets the default values for the multiplicative changes **ALL** equal to **ONE**, i.e., all the parameters will be subjected to shocks that leave their values unchanged. In order to implement shocks the user needs to change the default values. Brief details of how to do this are given below.

Note that if there are no tax rates, remittances and aid and domestic transfers for a region in the base case this method does not allow the user to simulate changes in those instruments.

Tax rates

If the user wants to simulate a reduction/increase in an existing tax rate, then the value for that tax instrument in the selected region needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a tax, the value should be zero. To change a tax into a subsidy the value should be set to less than ZERO. The layout of worksheet for tax shocks is illustrated below.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Tax Rate shocks											
2	Default values - If the user changes these values the model will run with the changed, USER defined, values											
3												
4			stmSIM	steSIM	stsSIM	stvSIM	stxSIM	styhSIM	styfSIM	stfSIM		
5	sim01	ramer	1	1	1	1	1	1	1	1		
6	sim01	euro	1	1	1	1	1	1	1	1		
7	sim01	namer	1	1	1	1	1	1	1	1		
8	sim01	afri	1	1	1	1	1	1	1	1		
9	sim01	sasia	1	1	1	1	1	1	1	1		
10	sim01	fsu	1	1	1	1	1	1	1	1		
11	sim01	rest	1	1	1	1	1	1	1	1		
12	sim01	easia	1	1	1	1	1	1	1	1		
13	sim01	meast	1	1	1	1	1	1	1	1		
14	sim01	glob	1	1	1	1	1	1	1	1		
15												
16												

The tax instruments are identified by the columns where the labels are interpreted as

- `stmSIM` – import duties on commodities;
- `steSIM` – export taxes on commodities;
- `stsSIM` – sales taxes (GST) on all domestically consumed commodities;
- `stvSIM` – value added taxes (VAT) on commodities consumed by households only;
- `stxSIM` – production taxes on activities;
- `styhSIM` – income (direct) taxes on households;
- `styfSIM` – income (direct) taxes on factors; and
- `stfSIM` – factor use taxes on factors (f) used by activities (a).

Note how the first column of the table identifies the label for the simulation – '`sim01`'. This should not be changed for the basic standard single simulation setup; more advanced users may wish to make changes (see below).

Technology

If the user wants to simulate a reduction/increase in an existing level of technology, then the value for that technology instrument in the selected region needs to be set to a value MORE THAN ONE/LESS THAN ONE. Technologies CANNOT be eliminated; hence ZERO and NEGATIVE vales are not valid. The layout of worksheet for technology shocks is illustrated below.

	A	B	C	D	E	F	G	H	I	J
1	**Technology shocks**									
2	**Default values - If the user changes these values the model will run with the changed, USER defined, values**									
3										
4			sadxSIM	sadvaSIM	sadflndSIM	sadfsklSIM	sadfuskSIM	sadfcapSIM		
5	sim01	ramer	1	1	1	1	1	1		
6	sim01	euro	1	1	1	1	1	1		
7	sim01	namer	1	1	1	1	1	1		
8	sim01	afri	1	1	1	1	1	1		
9	sim01	sasia	1	1	1	1	1	1		
10	sim01	fsu	1	1	1	1	1	1		
11	sim01	rest	1	1	1	1	1	1		
12	sim01	easia	1	1	1	1	1	1		
13	sim01	meast	1	1	1	1	1	1		
14	sim01	glob	1	1	1	1	1	1		
15										
16										
17										

The technology instruments are identified by the columns where the labels are interpreted as

- `sadxSIM` – total factor productivity with respect to (wrt) all inputs;
- `sadvaSIM` – total factor productivity wrt all factor inputs;
- `sadflndSIM` –factor specific productivity wrt land;
- `sadfsklSIM` –factor specific productivity wrt skilled labour;
- `sadfuskSIM` –factor specific productivity wrt unskilled labour; and
- `sadfcapSIM` –factor specific productivity wrt capital.

Remittances

If the user wants to simulate a reduction/increase in an existing share of household income remitted from one region to another, then the value for that remittance instrument from the selected **source** region (*r* in the column) to the selected **destination** region (*w* in the row) needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a remittance, the value should be zero. Values less than ZERO are not valid. The layout of worksheet for remittance shocks is illustrated below.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	**Remittance shocks**												
2	**Default values - If the user changes these values the model will run with the changed, USER defined, values**												
3	**Remittances from COLUMN region to ROW region**												
4													
5			ramer	euro	namer	afri	sasia	fsu	rest	easia	meast	glob	
6	sim01	wramer	1	1	1	1	1	1	1	1	1	1	
7	sim01	weuro	1	1	1	1	1	1	1	1	1	1	
8	sim01	wnamer	1	1	1	1	1	1	1	1	1	1	
9	sim01	wafri	1	1	1	1	1	1	1	1	1	1	
10	sim01	wsasia	1	1	1	1	1	1	1	1	1	1	
11	sim01	wfsu	1	1	1	1	1	1	1	1	1	1	
12	sim01	wrest	1	1	1	1	1	1	1	1	1	1	
13	sim01	weasia	1	1	1	1	1	1	1	1	1	1	
14	sim01	wmeast	1	1	1	1	1	1	1	1	1	1	
15	sim01	wglob	1	1	1	1	1	1	1	1	1	1	
16													
17													
18													
19													

The **source** regions are identified in the columns and the **destination** regions are identified in the rows. Those an entry in the cell `wafri:euro` refers to remittances from the region Europe (`euro`) to the regions Africa (`afri`). NB: remittances can flow both ways.

Bilateral aid

If the user wants to simulate a reduction/increase in an existing share of bilateral aid from GDP transferred from one region to another, then the value for that bilateral aid instrument from the selected **source** region (*r* in the column) to the selected **destination** region (*w* in the row) needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a bilateral aid, the value should be zero. Values less than ZERO are not valid. The layout of worksheet for bilateral aid (`aid_b_shocks`) is the same as for remittance shocks.

The **source** regions are identified in the columns and the **destination** regions are identified in the rows. Thus an entry in the cell `wafri:euro` refers to bilateral aid from the region Europe (`euro`) to the region Africa (`afri`).

Multilateral aid

If the user wants to simulate a reduction/increase in an existing share of multilateral aid from GDP transferred from one region to another, then the value for that multilateral aid instrument from the selected **source** region (*r* in the column) to the selected **destination** region (*w* in the row) needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a multilateral aid, the value should be zero. Values less than ZERO are not valid. The layout of worksheet for multilateral aid (`aid_m_shocks`) is the same as for remittance shocks.

The **source** regions are identified in the columns and the **destination** regions are identified in the rows. Thus an entry in the cell `wafri:euro` refers to multilateral aid from the region Europe (`euro`) to the region Africa (`afri`).

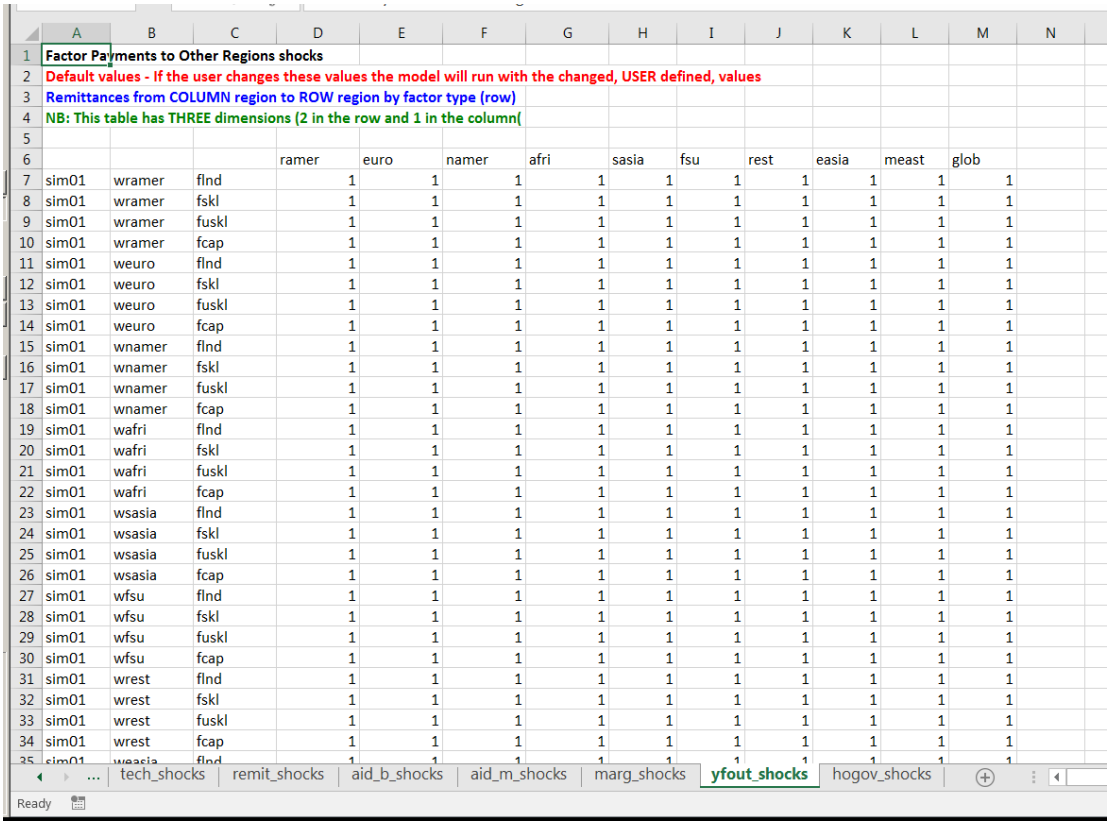
Margins

If the user wants to simulate a reduction/increase in the margin coefficients on trade between regions, then the value for that coefficient for imports from the selected **source** region (*w* in the row) to the selected **destination** region (*r* in the column) needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a margin coefficient, the value should be zero; but this implies that trade is costless, which is not logical. Values less than ZERO are not valid. The layout of worksheet for margin shocks (`marg_shocks`) is the same as for remittance shocks.

The **source** regions are identified in the row and the **destination** regions are identified in the column. Thus an entry in the cell `wafri:euro` refers to margins associated with trade from the region Africa (`afri`) to the region Europe (`euro`).

Factor payments

The layout for the factor payments worksheet (`yfout_shocks`) differs from that in previous worksheets since there are now three dimensions in the row and one in the column. The columns identify the **source** regions, the first dimension in the row identifies the simulations, the second dimension in the row the **destination** regions and the third dimension in the row the factor paid for. If the user wants to simulate a reduction/increase in the payments of factor incomes from the selected **source** region (*r* in the column) to the selected **destination** region (*w* in the row) with respect to a factor (*f* in the row) then the entry needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a factor payment, the value should be zero. Values less than ZERO are not valid.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Factor Payments to Other Regions shocks													
2	Default values - If the user changes these values the model will run with the changed, USER defined, values													
3	Remittances from COLUMN region to ROW region by factor type (row)													
4	NB: This table has THREE dimensions (2 in the row and 1 in the column(													
5														
6				ramer	euro	namer	afri	sasia	fsu	rest	easia	meast	glob	
7	sim01	wramer	flnd	1	1	1	1	1	1	1	1	1	1	
8	sim01	wramer	fskl	1	1	1	1	1	1	1	1	1	1	
9	sim01	wramer	fuskl	1	1	1	1	1	1	1	1	1	1	
10	sim01	wramer	fcap	1	1	1	1	1	1	1	1	1	1	
11	sim01	weuro	flnd	1	1	1	1	1	1	1	1	1	1	
12	sim01	weuro	fskl	1	1	1	1	1	1	1	1	1	1	
13	sim01	weuro	fuskl	1	1	1	1	1	1	1	1	1	1	
14	sim01	weuro	fcap	1	1	1	1	1	1	1	1	1	1	
15	sim01	wnamer	flnd	1	1	1	1	1	1	1	1	1	1	
16	sim01	wnamer	fskl	1	1	1	1	1	1	1	1	1	1	
17	sim01	wnamer	fuskl	1	1	1	1	1	1	1	1	1	1	
18	sim01	wnamer	fcap	1	1	1	1	1	1	1	1	1	1	
19	sim01	wafri	flnd	1	1	1	1	1	1	1	1	1	1	
20	sim01	wafri	fskl	1	1	1	1	1	1	1	1	1	1	
21	sim01	wafri	fuskl	1	1	1	1	1	1	1	1	1	1	
22	sim01	wafri	fcap	1	1	1	1	1	1	1	1	1	1	
23	sim01	wsasia	flnd	1	1	1	1	1	1	1	1	1	1	
24	sim01	wsasia	fskl	1	1	1	1	1	1	1	1	1	1	
25	sim01	wsasia	fuskl	1	1	1	1	1	1	1	1	1	1	
26	sim01	wsasia	fcap	1	1	1	1	1	1	1	1	1	1	
27	sim01	wfsu	flnd	1	1	1	1	1	1	1	1	1	1	
28	sim01	wfsu	fskl	1	1	1	1	1	1	1	1	1	1	
29	sim01	wfsu	fuskl	1	1	1	1	1	1	1	1	1	1	
30	sim01	wfsu	fcap	1	1	1	1	1	1	1	1	1	1	
31	sim01	wrest	flnd	1	1	1	1	1	1	1	1	1	1	
32	sim01	wrest	fskl	1	1	1	1	1	1	1	1	1	1	
33	sim01	wrest	fuskl	1	1	1	1	1	1	1	1	1	1	
34	sim01	wrest	fcap	1	1	1	1	1	1	1	1	1	1	
35	sim01	weasia	flnd	1	1	1	1	1	1	1	1	1	1	

... | tech_shocks | remit_shocks | aid_b_shocks | aid_m_shocks | marg_shocks | yfout_shocks | hogov_shocks

Ready

Thus an entry in the cell `wafri:facp:euro` refers to payments from the **source** region Europe (`euro`) to the **destination** region Africa (`afri`) for the **factor** capital (fcap) owned in Africa but used in Europe.

Domestic transfers

If the user wants to simulate a reduction/increase in the real value of domestic transfers from the government in a region (*r* in the row) to the household (*hous* in the column) the entry needs to be set to a value LESS THAN ONE/MORE THAN ONE. To eliminate a margin coefficient, the value should be zero. Values less than ZERO are not valid. The layout worksheet for domestic transfer shocks (hogov_shocks) is illustrated below.

	A	B	C	D	E	F	G	H	I	J	K
1	**Domestic transfer**										
2	**Default values - If the user changes these values the model will run with the changed, USER defined, values**										
3											
4											
5			hous								
6	sim01	ramer	1								
7	sim01	euro	1								
8	sim01	namer	1								
9	sim01	afri	1								
10	sim01	sasia	1								
11	sim01	fsu	1								
12	sim01	rest	1								
13	sim01	easia	1								
14	sim01	meast	1								
15	sim01	glob	1								
16											

4.4 Simulation Results

All the results from the simulations are written out to ten GDX files that are saved into a directory called `Results`, which is a sub directory of the directory `R23_mod`. The ten GDX files with brief descriptions of their contents are:

1. resLEV(r,res,clos,sim) LEVELS results for scalar variables from simulations
2. resPER(r,res,clos,sim) PERCENTAGE change results for scalar variables from sims
3. resWLEV(w,r,resw,clos,sim) LEVELS results for scalar variables from simulations

4. resWPER(w,r,resw,clos,sim) PERCENTAGE change results for scalar variables from sims
5. resFLEV(f,r,resf,clos,sim) LEVELS results for factor variables from simulations
6. resFPER(f,r,resf,clos,sim) PERCENTAGE change results for factor vars from sims
7. resALEV(w,g,r,resa,clos,sim) LEVELS results for factor variables from simulations
8. resAPER(w,g,r,resa,clos,sim) PERCENTAGE change results for factor vars from sims
9. resHLEV(w,h,r,resh,clos,sim) LEVELS results for factor variables from simulations
10. resHPER(w,h,r,resh,clos,sim) PERCENTAGE change results for factor vars from sims

These results could be assessed by opening the respective GDX files using GAMSIDE. However, it is easier if it is done using the SeeResults software that is described in Section 5 (below).

4.5 More Advanced Simulations

Advanced users are likely to want to programme more complicated simulations. This can be readily done by increasing the number of simulations – the set `sim` that is defined in the worksheet `exp_sets` (column A) of the workbook `R23_data.xlsm`. The standard experiment file is already set up to allow the user to run LOOPS over the simulations (`sim`) the closures (`clos`) and the elasticities (`elst`), so the user can readily extend the experiments run. This will of course require the user to modify the standard experiment file – `R23_expt_std.inc`.

Similarly, the standard shocks that are assigned from the Excel workbook `R23_data.xlsm`, can be readily extended by extending each worksheet to include information for additional simulations – members of the set `sim`.

It is anticipated that some users will choose to use the save and restart option provided by GAMS. Thus an additional file – `R23_expt.gms` – is provided for the user to adapt.

It is assumed that advanced users will not need additional information.

5 Accessing Model Results using SeeResults

Accessing model simulation results through `GAMSIDE` or `GDXViewer` is very convenient – multi dimension results parameters can be filtered easily and the products of the filtering process can be easily exported. But there are limitations to `GAMSIDE` and `GDXViewer`; the `SeeResults` programme was developed to address some of these 'limitations'.

`SeeResults` is an Excel based VBA programme, which suitably configured can be used to access data from any GDX file. `SeeResults` exploits the pivot table features of Excel and uses the graphing and other features of Excel. There are different versions of `SeeResults` depending on the version of Excel being used: '`SeeResults_07.xlsm`', '`SeeResults_10.xlsm`', '`SeeResults_13.xlsm`' and `SeeResults_07.xlsm`' or '`SeeResults_16.xlsm`', where the number identifies the version of Excel.[11]

5.1 Setting Up SeeResults

When first starting `SeeResults` it is necessary to ensure that Excel is configured so that it will accept the operation of macros. Typically Excel is configured to reject macros. When opening SeeResults using Excel 2007\2010\2013\2016 'some active content has disabled' appears as a Security Warning, click on the (Excel) `options` button and select `enable this content` or `enable content`

[11] The different versions of SeeResults are broadly identical; the different versions exist to allow for changes in Excel over the years. SeeResults was first developed c2000, so there are versions before Excel 2007 although these versions have not been checked for robustness in recent years.

Figure 5.1.1a Opening SeeResults_07\10

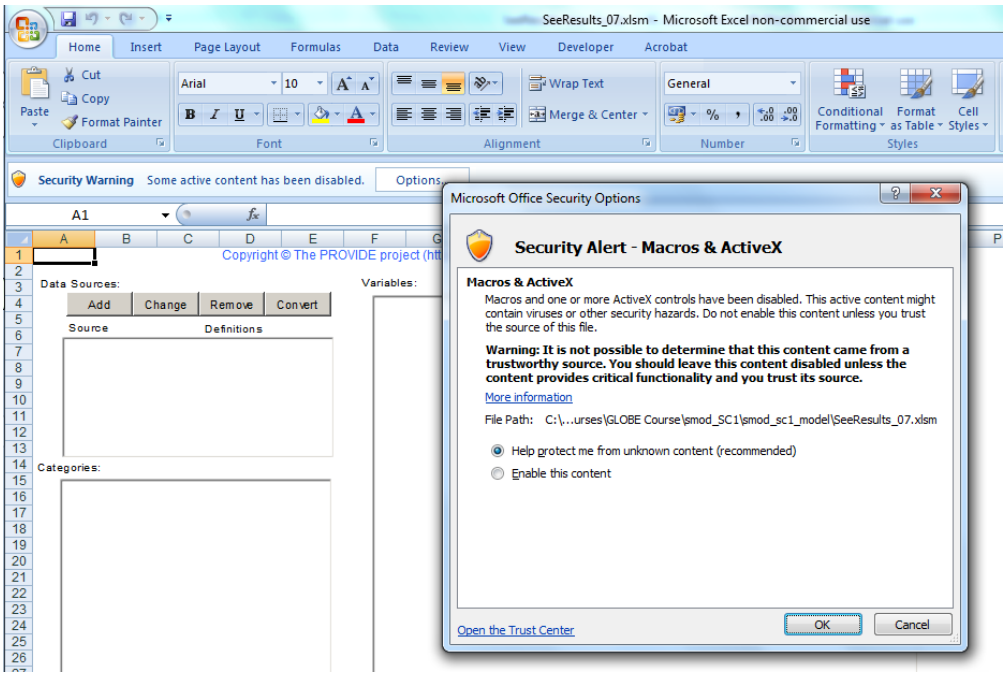
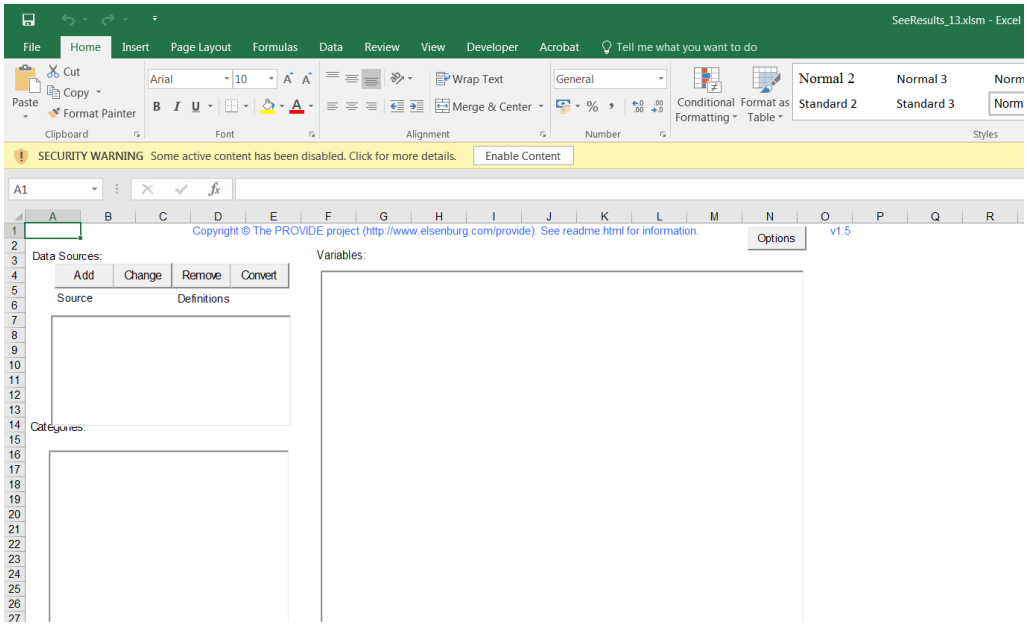
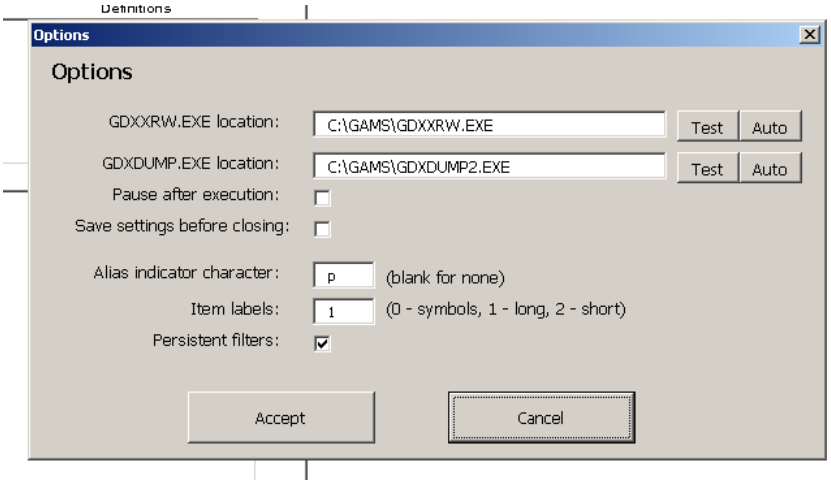


Figure 5.1.1b Opening SeeResults_13\16



The operation of `SeeResults` is described in detail in PROVIDE (2004b) so the comments here are deliberately limited. The typical user of `SeeResults` only has to concern themselves with two types of settings; the options and data definitions.

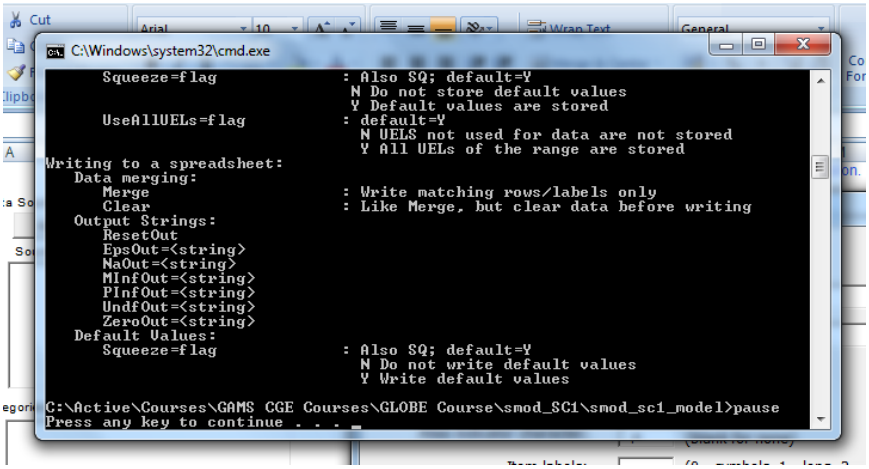
Figure 5.1.2 SeeResults Options Settings



The options settings page – see Figure 5.1.2 – has four options that usually get changed (the remainder are part of the configuration of `SeeResults` and are covered in detail in the manual); first the locations of `GDXXRW.EXE` and `GDXDUMP.EXE`, second, the 'item labels' that control the substitution of the GAMS labels with more user friendly labels and third persistent filters.

To identify the locations of `GDXXRW.EXE` and `GDXDUMP.EXE`, click on `AUTO` for each `EXE` file. Tick the 'Pause after Execution' box and click `TEST` for the `GDXXRW.EXE` entry. A dos style window will appear; if it is in the form found in Figure 5.1.3 then `GDXXRW.EXE` has been correctly located. Repeat the process for the `GDXDUMP.EXE` entry. This simply ensures that `SeeResults` knows where to find `GDXXRW.EXE` and `GDXDUMP.EXE`; it will use these programmes when accessing the results.

Figure 5.1.3 GDXXRW location



The 'item labels' box controls the symbols[12] used when SeeResults converts a symbol in a GDX files into a PivotTable. It operates by substituting the set labels used in GAMS for descriptions provided by the user. This allows the user to produce tables and graphs in Excel for reports etc., using set labels that can have more meaning to the reader of the reports. There are three option: 0 uses the labels as used in GAMS; 1 uses a 'long' description (commonly the detailed description used in the GAMS programme; 2 uses a 'short' description. Typically option 2 is the default since it should be customised to the GDX file and report. The information use to implement the chosen option is provided in the model and experiment specific 'data definitions' file; this is detailed below in section 5.2.

If there are any errors in the 'data definitions' `SeeResults` is designed to resort to the labels used in GAMS; this minimise the probability of errors in translation between GDX and Excel.

The 'Alias indicator character' box allows for the fact that GAMS/GDX may have used the same set to define more than one dimension of the results parameters contained in the GDX file. To do so GAMS will have used aliases; i.e., the sets of the same elements with different names. One approach, and that used in the GLOBE, STAGE and R23 models is to identify aliases by attaching one or more '*p*' characters to the base sets label, e.g., *w*/*wp*/*wpp*. Excel does not 'understand' that two or more sets, e.g., *w* and *wp*, can have the same members and therefore would require definitions for all sets and all their aliases to be provided in the

[12] In GAMS symbols encompass sets, parameters, variables etc.

'data definitions' file. If a common final character is used for aliases then `SeeResults,` can remove these characters when moving symbols from GDX to Excel, thereby saving effort.

If there is a tick in the box for 'Persistent filters' then the filtering used for on symbol carries over the next symbol moved from GDX to `SeeResults`. This can save time when accessing results.

It is recommended that the 'Pause after Execution' and 'Save settings before closing' boxes are left unticked in normally use. The former means the dos window will close on completion. The latter means that if `SeeResults` is closed without saving, the settings from the current session are lost; experience indicates that starting afresh with each session is less likely to produce errors.

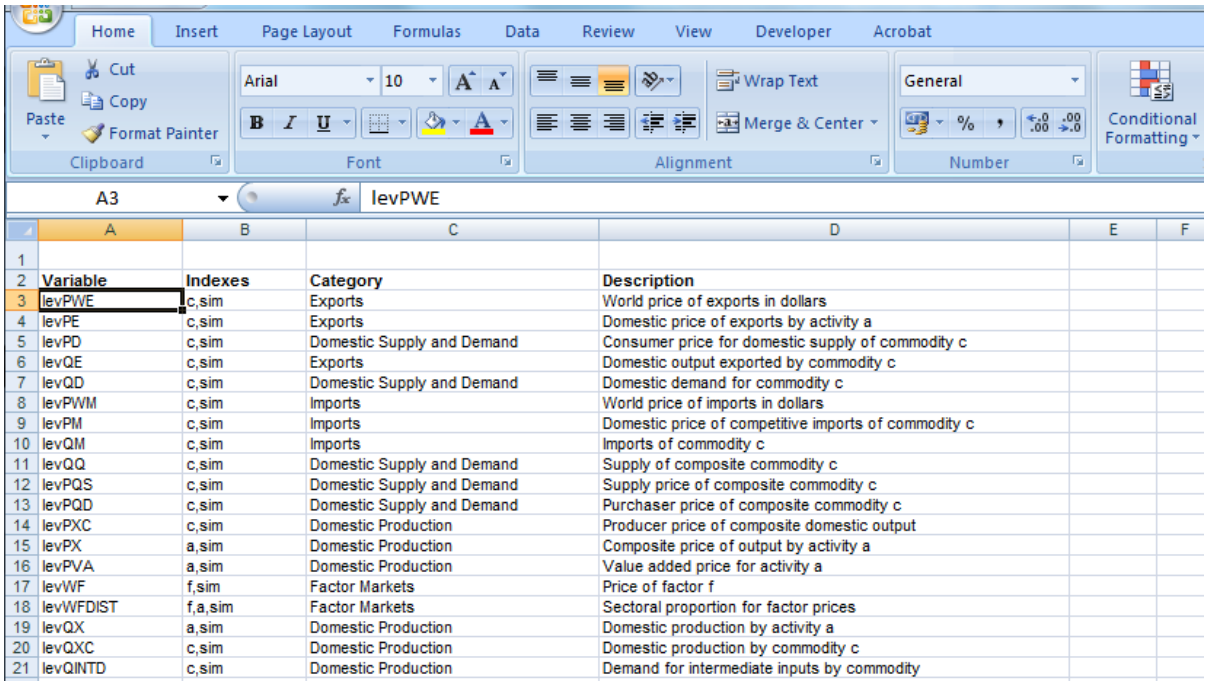
5.2 Setting up the Data Definitions File

The user also needs to configure the data definitions that are recorded in an Excel workbook called `r23_DataDef.xls`. The data definitions files for `SeeResults` have three worksheets; 'Information', 'Sets' and 'Variables' and are always saved a *.xls files. The 'Information' worksheet simply records information about SeeResults and the downloadable example; it never needs to be changed.

The 'Variables'[13] worksheet is specific to the model and results files; unless the model and/or results files are changed this worksheet should not be changed. If a user modifies the model and/or the results files, then the user needs to update the 'Variables' worksheet. There are four columns in the worksheet. Three of the columns, 'variables', indices' and 'description', should be copied (identically) from the GAMS programme to aid interpretation; `SeeResults` uses the information in these columns when presenting the data. The fourth column is user specific and is used to group 'variables' in the GDX database; this pre filtering makes it easier to identify results. The number of rows, their order and labels should note be changed, otherwise this worksheet can be adjusted to meet the users needs.

[13] Strictly the term 'variables' is not correct in terms of the data stored in the GDX file. The data in the GDX file are parameters that are used to store the results for variables and summary variables derived from the models variables and parameters.

Figure 5.2.1 Data Definitions: Variable Worksheet



	A	B	C	D
2	**Variable**	**Indexes**	**Category**	**Description**
3	levPWE	c,sim	Exports	World price of exports in dollars
4	levPE	c,sim	Exports	Domestic price of exports by activity a
5	levPD	c,sim	Domestic Supply and Demand	Consumer price for domestic supply of commodity c
6	levQE	c,sim	Exports	Domestic output exported by commodity c
7	levQD	c,sim	Domestic Supply and Demand	Domestic demand for commodity c
8	levPWM	c,sim	Imports	World price of imports in dollars
9	levPM	c,sim	Imports	Domestic price of competitive imports of commodity c
10	levQM	c,sim	Imports	Imports of commodity c
11	levQQ	c,sim	Domestic Supply and Demand	Supply of composite commodity c
12	levPQS	c,sim	Domestic Supply and Demand	Supply price of composite commodity c
13	levPQD	c,sim	Domestic Supply and Demand	Purchaser price of composite commodity c
14	levPXC	c,sim	Domestic Production	Producer price of composite domestic output
15	levPX	a,sim	Domestic Production	Composite price of output by activity a
16	levPVA	a,sim	Domestic Production	Value added price for activity a
17	levWF	f,sim	Factor Markets	Price of factor f
18	levWFDIST	f,a,sim	Factor Markets	Sectoral proportion for factor prices
19	levQX	a,sim	Domestic Production	Domestic production by activity a
20	levQXC	c,sim	Domestic Production	Domestic production by commodity c
21	levQINTD	c,sim	Domestic Production	Demand for intermediate inputs by commodity

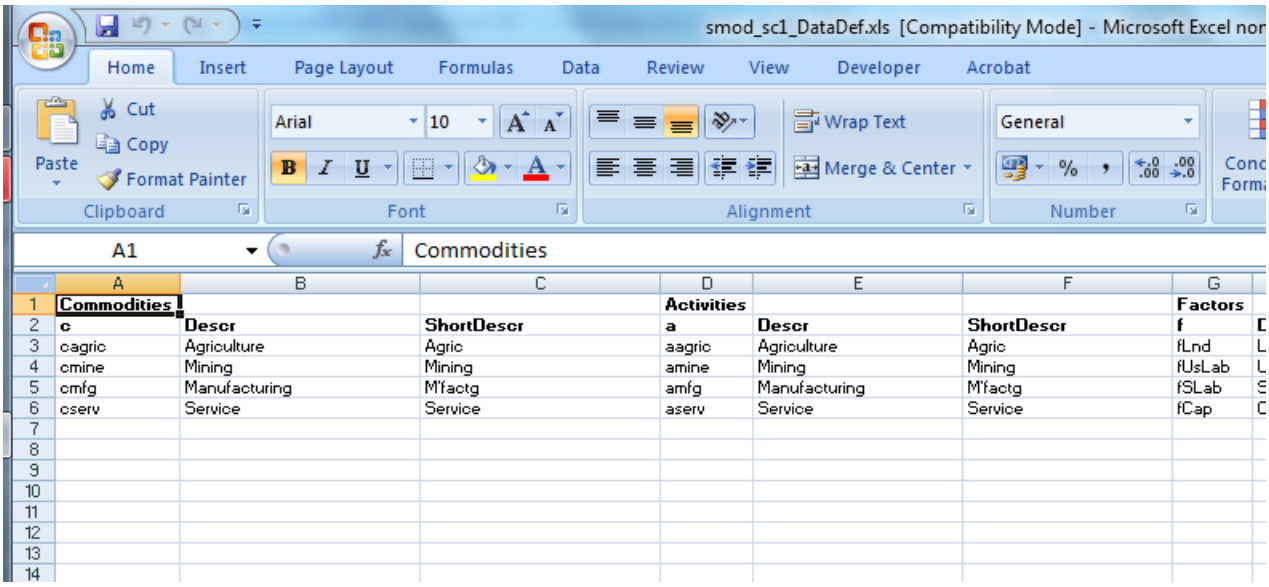
The '`sets`' worksheet (Figure 5.2.2) contains details for ALL the sets used in the results files and needs revision whenever a need set of experiments are implemented. Each `set` has **three** columns; the first gives the GAMS label, the second the 'long' label and the third the 'short' label – the choice of which is used in the Pivot tables generated by `SeeResults` is selected in the options page. The requirement for three columns for each set is absolutely binding. When setting up a new version of a model the '`sets`' need updating to include all those sets used in the new version. `SeeResults` is however 'smart': if the set labels in the GDX file\files do not match those in the data definition table the `SeeResults` uses the GAMS labels. In addition, the user should include labels and descriptions for the simulations conducted in the experiment file.

For the R23 model the process has largely been automated. The details sets in an aggregation of the database used in the model are written to the default data definitions file – '`r23_DataDef.xls`' – when generating an aggregation using SAMGator. SAMGator also writes the default sets for simulations (`sim` and `simc`), closure (`clos`) and (elasticity) sensitivity analyses (`elst`) in the basic user version of the R23 model to the data definitions file. For advanced users there is code in the template experiment file that will update the sets

defined by the user for simulations (`sim` and `simc`), closure (`clos`) and (elasticity) sensitivity analyses (`elst`).

NB: Information can only be written to and Excel workbook by GAMS if the workbook is closed.

Figure 5.2.2 Data Definitions: Sets Worksheet



	A	B	C	D	E	F	G
1	**Commodities**			**Activities**			**Factors**
2	**c**	**Descr**	**ShortDescr**	**a**	**Descr**	**ShortDescr**	**f**
3	cagric	Agriculture	Agric	aagric	Agriculture	Agric	fLnd
4	cmine	Mining	Mining	amine	Mining	Mining	fUsLab
5	cmfg	Manufacturing	M'factg	amfg	Manufacturing	M'factg	fSLab
6	cserv	Service	Service	aserv	Service	Service	fCap
7							
8							
9							
10							
11							
12							
13							
14							

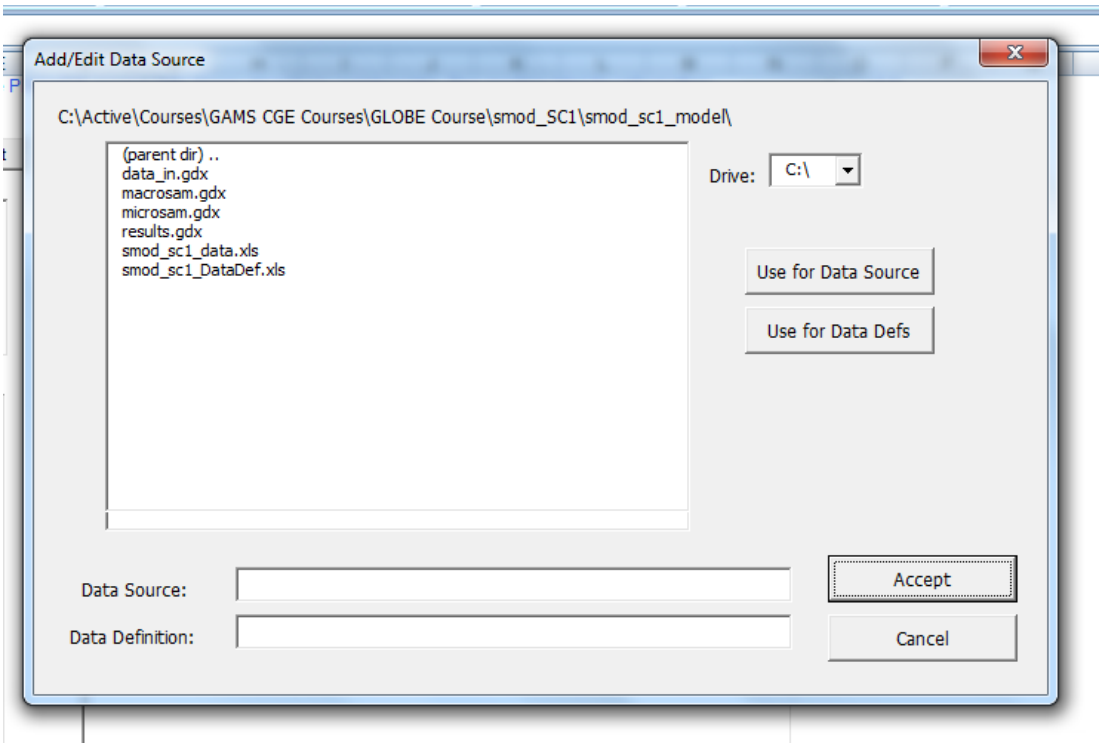
The '`variables`' worksheet – see Figure 5 - only needs changing if the variables in the model have been changed and\or additional result parameters have been created. The order of the variables is not important, except for readability, so it is simple to add the additional information at the bottom of the worksheet.

Accessing the Parameters in Results.gdx

1. Save and close down `SeeResults` and Excel. Now restart `SeeResults` (this just ensures everything is starting from a 'clean' point).

2. Select Add and the following dialogue box opens (Figure 6)

Figure 6 Add\Edit Data Source



3. Select `results.gdx` for the data Source and `smod_sc1_DataDef.xls` for the Data Defs.

4. Note how the results can be filtered using the categories box, which groups related variables. The user controls these groupings. (see Figure 5, column C)

5. Now select – `pcPM(c,sim,clos)` – by double clicking.

6. This opens a new worksheet – '`PTableView`' – which can now be filtered by sets. (see Figure 8)

7. Experiment with various filtering.

8. Using `pcPM(c,sim,clos)` filter as follows: *`c`* – all in the row, *`sim`* – all in the columns; *`clos`* – all in the columns. Click on `Table` to refresh.

9. Click on `Export` to send the resultant filtering to Excel. Note how useful information has also been exported to the new Excel workbook.

10. Use `SeeResults` to generate an Excel workbook of results comparable to those you generated during Lab 5.

Figure 7 Parameter Categories

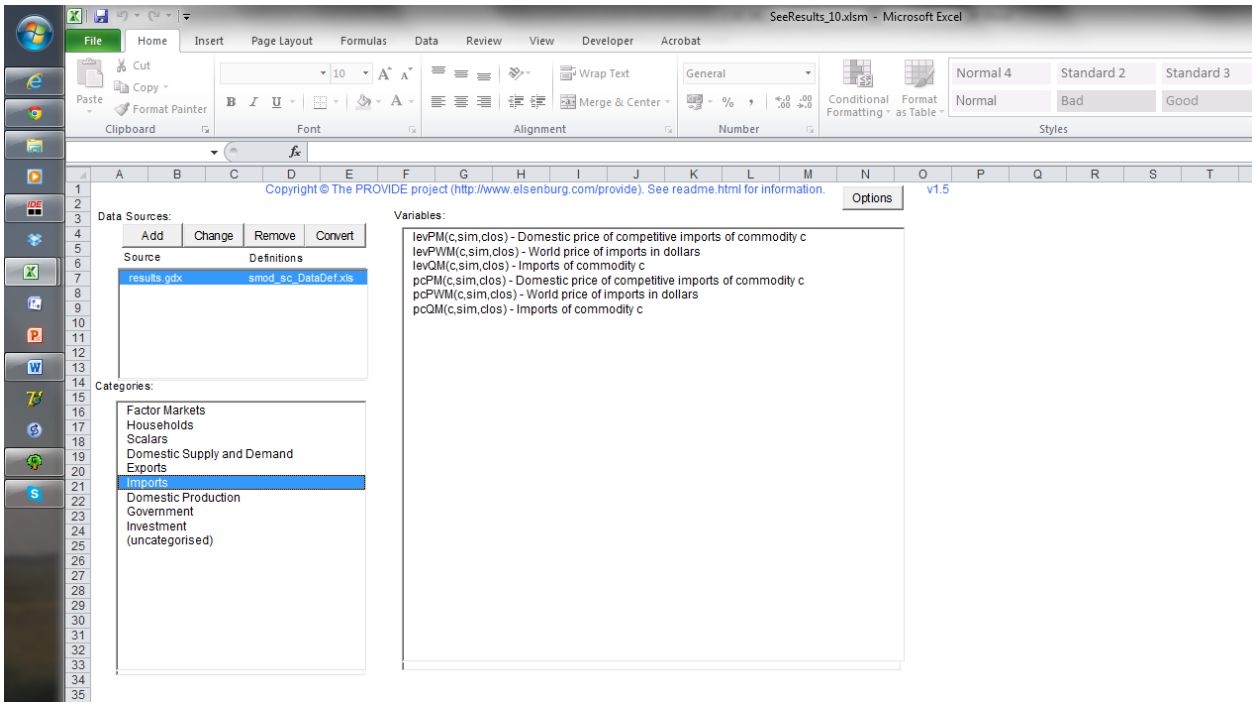
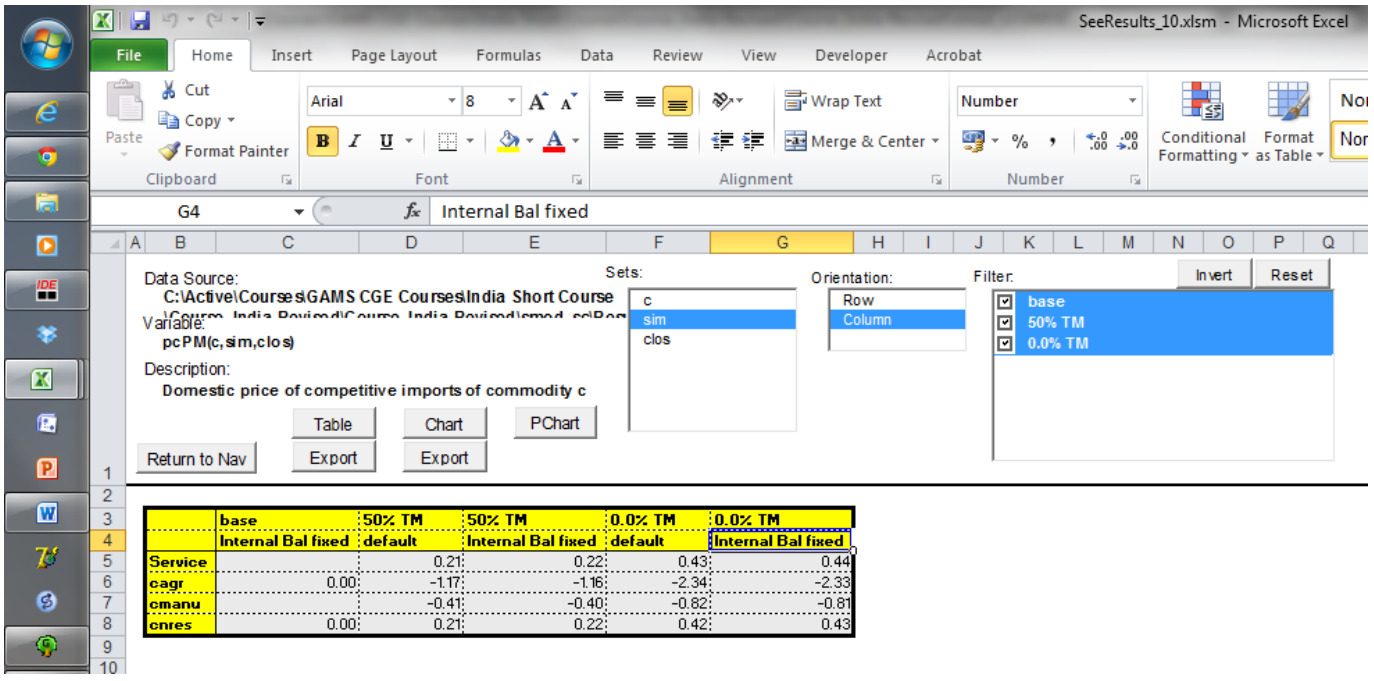


Figure 8 PTableView



	base	50% TM	50% TM	0.0% TM	0.0% TM
	Internal Bal fixed	default	Internal Bal fixed	default	Internal Bal fixed
Service		0.21	0.22	0.43	0.44
cagr	0.00	-1.17	-1.16	-2.34	-2.33
cmanu		-0.41	-0.40	-0.82	-0.81
cnres	0.00	0.21	0.22	0.42	0.43

Appendix 1

An Introduction to (Student) GAMS and GAMSIDE

(General Algebraic Modelling System)

Introduction

The GAMS (General Algebraic Modelling System) (version 2.50) suite consists of the base GAMS module and a collection of solvers. An interface, GAMSIDE, is also available. GAMS is a high level programming language amongst whose objectives is to allow programmers to prepare programmes that are transparent. When a programme is run GAMS converts it into code compatible with a specified solver, executes the programme by calling the solver and writes a report file back. GAMS is attractive because

- it can use a range of specialised solvers without requiring the user to know their specific syntax;
- the separation of data and the logic of a problem allows the size of the problem to be increased without increasing the complexity of the representation;
- the programme is its own documentation; and
- it looks after a number of common programming problems, like other high level languages e.g., dimensionality.

"GAMS was developed to [overcome a series of mathematical programming problems] by

- Providing a high-level language for the compact representation of large and complex models
- Allowing changes to be made in model specifications simply and safely
- Allowing unambiguous statements of algebraic relationships
- Permitting model descriptions that are independent of solution algorithms" (Brooke *et al*., 1998, p1).

A major, if not the major, use of GAMS is for optimisation models. Simple examples of such models are linear programming (LP) models, CGE models are another example.

Among the solvers available are

Solvers

BDMLP	LP
MINOS/5	NLP
ZOOM	Zero/One Optimisation Method
CONOPT/3	NLP optimizer
CPLEX	LP and Mixed Integer Programme solver
LAMPS	linear and mixed integer solver
OSL	primal and dual Simples, interior point methods, MIP solver
LOQO	Interior point solver
XA	linear and mixed integer solver
DICOPT	Mixed integer Nonlinear Programming
MILES	Mixed complementarity problem solver
PATH	Mixed complementarity problem solver
PATHNLP	NLP

To use GAMS you need a programme file. All GAMS programme files are prepared as text files and saved as `[filename]`.gms. (Note: it is no longer necessary to limit the filename to 8 characters or to avoid spaces). The easiest way to prepare a GAMS programme is by using a modern Windows based text file editor. You can also use the editor to read GAMS output files. GAMS output files are written (automatically) to disk as `[filename]`.lst files where the filename is the same filename for the `[filename]`.gms file.

You can write/read GAMS files in any text editor (any DOS or Windows editor, even Word in txt mode).

There are various ways to run a GAMS programme file. One of the easier ways is to use GAMS's own text editor GAMSIDE, although other users have preferences for other editors/EMACS.

Installing GAMS 2.50 with GAMSIDE

The Student or demo edition of GAMS 2.50 is available as a single exe file, that can be downloaded freely from the GAMS website.[14] Installation instructions from GAMS and some additional instructions are available from the web site (http://www.gams.com). To read the installation instructions a copy of Adobe Acrobat Reader is required (available free from http://www.adobe.com). Acrobat Reader is also required to access the electronic versions of the GAMS Manual and Tutorial. It is also very useful to have access to file compression software, e.g., WinZip (http://www.winzip.com), RAR (http://www.??????), and Windows 7, to transfer files efficiently.

The installation process is standard for Win95/98/NT/XP/VISTA/7 programmes. In brief the installation process for Win95/98/XP/Vista/7 is as follows

i) The latest version of GAMS is available as a student/demo version from GAMS at http://www.gams.com/download/. It is necessary to select the version of GAMS consistent with your operating system: in this document it is assumed that you are using a version of Windows.

ii) Copy the file '`windows_x86_32.exe`' or '`windows_x86_64.exe`'to a directory/folder[15], e.g., `TEMP`, on your C drive.

iii) Run the programme '`windows_x86_32.exe`'/' `windows_x86_64.exe`'. It is WISE for you to install GAMS in a directory (say GAMS) on the top level of the C Drive; this is **not** the default chosen by the installation software, but it does have some advantages. (It is assumed for all the exercises etc., detailed in this and related documents that GAMS has been installed into the directory C:\GAMS).

iv) Follow the installation instructions as they appear on the screen.

v) You will be asked for a license file at which point you have two options.

   a. Either point the installation programme to the license file (`gamslice.txt`) if you have a full version;

[14] The student version of GAMS/GAMSIDE is identical to the full version except for the fact that the absence of a licence file restricts the size of model that can be implemented. GAMS also make available licence files for courses that have a limited life span to allow users to explore the properties of GAMS before committing to the purchase of the full licence.

[15] A 'directory' and a 'folder' are the same; directory is the 'old' DOS name while 'folder' was the name used by Apple in the 1980s. Directory is the name used in this document.

b. OR choose the option that allows you to continue without a license file, in which case only the student/demo version will be installed.

vi) The installation of a Windows version of GAMS will conclude by offering the option to launch GAMS IDE; select this option.

vii) The final step will be to run a few models to check the installation was successful, this is explained below.

This brief description is NOT a substitute for reading the installation guide or using the Help facility in GAMSIDE.

GAMSIDE

(General Algebraic Modelling System Integrated Development Environment)

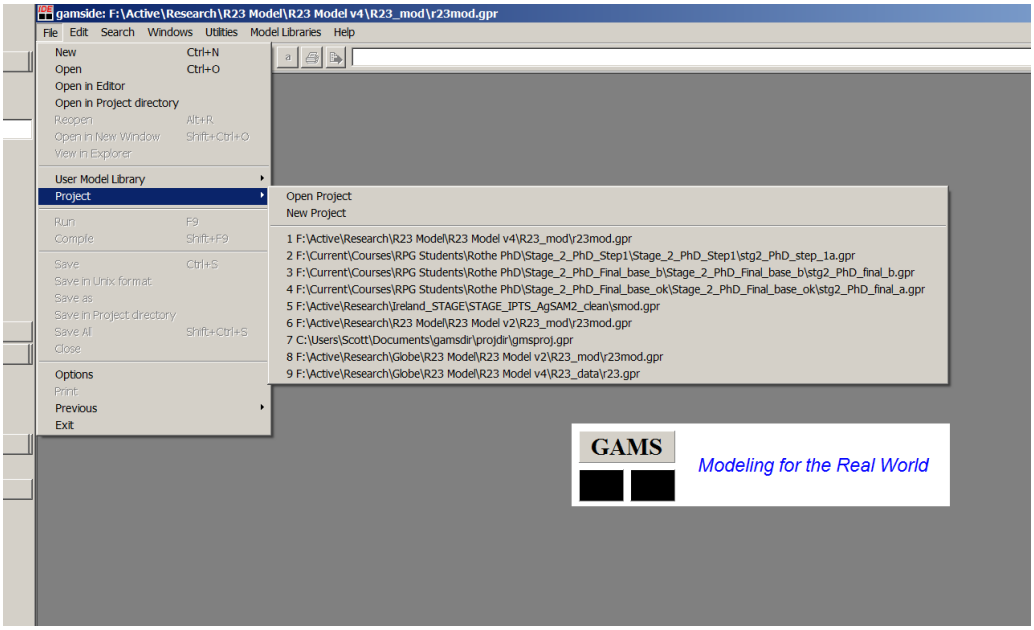
Using GAMS 2.50 with GAMS IDE

To use GAMS you need a programme file. All GAMS programme files are prepared as text files and saved as `[filename].gms`.[16] To prepare a GAMS programme it is necessary to use a text file editor. In the past it was necessary to use a text file editor to write the programme file and read the resulting (list) file(`[filename].lst`) and to run GAMS from DOS. GAMS**I**(ntegrated)**D**(evelopment)**E**(nvironment) allows you to do all this from a single Windows based interface. This makes things very much easier, but there are one or two features that are initially a little tricky to grasp.

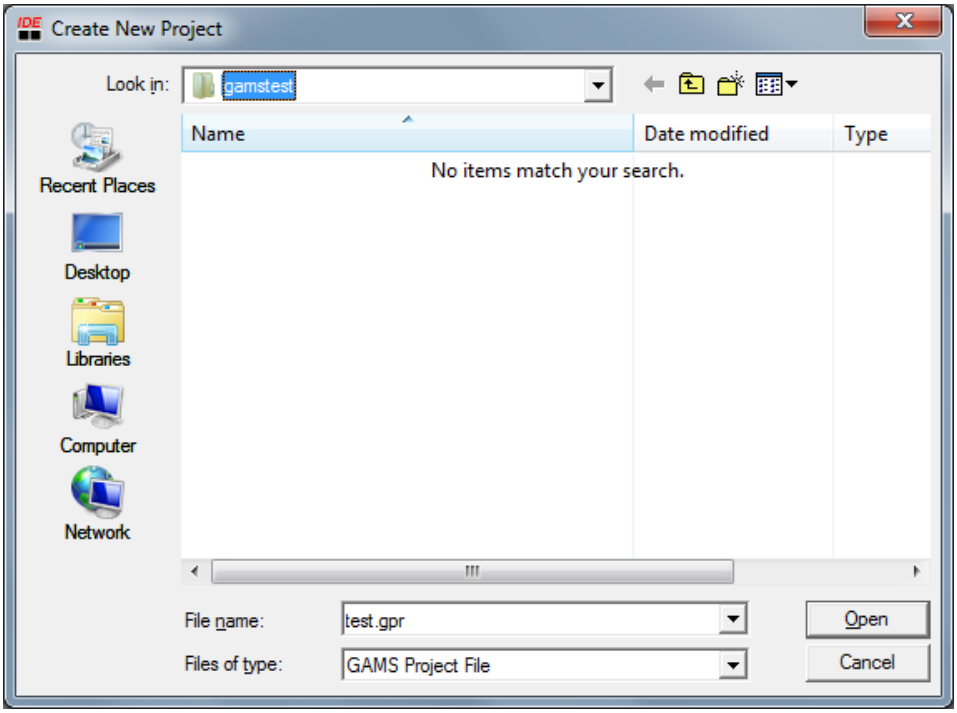
Creating a Project File

GAMSIDE uses a project file (`[project_name].grp`) to manage paths and keep track of files; the project file tells a GAMS programme the default location of files needed to run the programme and where GAMS should send the files generated by the programme. It cannot be recommended strongly enough that for each 'project' you develop that you keep all the related files in a single directory/folder with its own project file. Hence, the first step is to create a directory in which to keep all the related files; this can be done from within GAMSIDE.

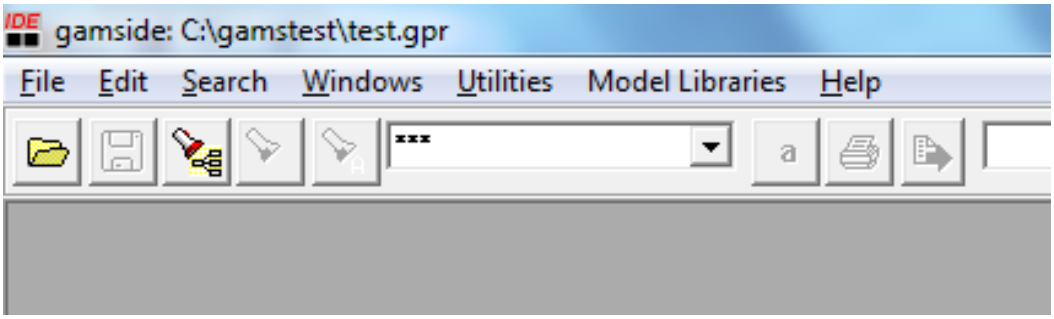
[16] Notes: (i) It is no longer necessary to limit filenames to 8 characters and to avoid spaces – but it may be good practice to avoid over long filenames and directory names so as to avoid long paths. (ii) It is also good practice to use multiple directories to avoid difficulties identifying related files.



From `File, Project` choose `New Project` and a 'standard' window (see below) will appear. Navigate to the top level of your C drive – or an alternative used for storing data on your PC – and create a directory called `'gamstest'`, in which a project file can be created by typing `'test.gpr'` in the File name box (see the screenshot below).



The title bar of the main window will show the project file name and the path for the working directory (see the cropped screenshot below).



Once a project file has been created the management of paths while working in GAMSIDE is automatic until a new project is created/selected. You will encounter problems with file management, if you try to short cut the above procedure.

It is not good practice to scroll through directories to find a file and then open it from within GAMSIDE. If you do, your programme may not run – because it cannot find necessary files – and if it does run you are likely to end up with `[filename].gms` files in different directories to the associated `[filename].lst` files. Rather if you need a file from another directory, use Windows Explorer to make a copy of the file (using the right button of the mouse) in the current project directory – you can always delete this file later and moreover this method makes sure you do not unintentionally corrupt a valuable file.

Discipline in file management is very important when programming. Good housekeeping requires both that you are tidy, e.g., keep all files relating to a specific project in one directory, use a system for file names that assists in identifying the relationship between files, etc., and that you document your work, e.g., keep a file that records the stages of a project in the directory and/or use the documentation facility in GAMS to record what each file does, etc. You will rapidly generate a relatively large number of files; without good housekeeping you will lose files regularly, with good housekeeping you will lose many fewer files and save yourself the effort of rewriting programme files.

In addition to the `[filename].lst` files GAMS will also automatically produce a number of files for backup and reporting purposes. Typically it is not necessary to keep these files after the session of work has been completed; deleting them 'manually', i.e., one by one, is slow and the possibility of making errors is high so it is necessary to be careful.[17]

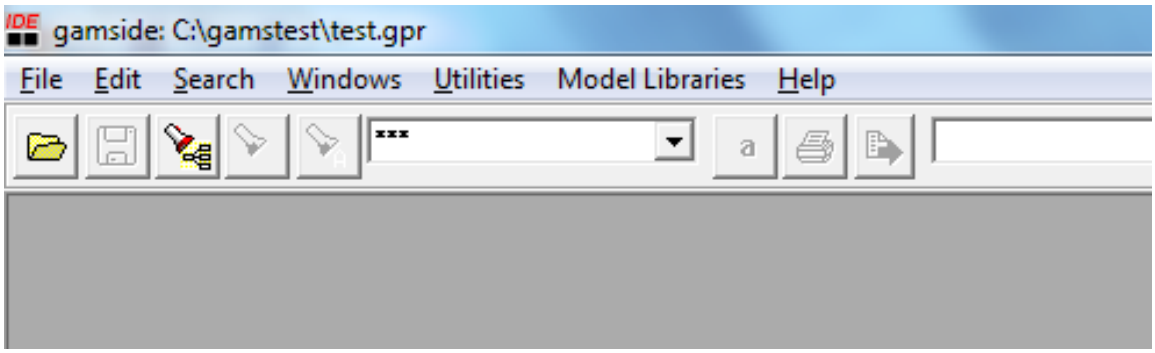
[17] A simple batch file to delete files that satisfy defined criteria is one way to avoid simple errors.

Preparing a GAMS Programme File

GAMSIDE is GAMS's own proprietary editor. You can therefore write your programme in GAMSIDE and save it to the project directory.

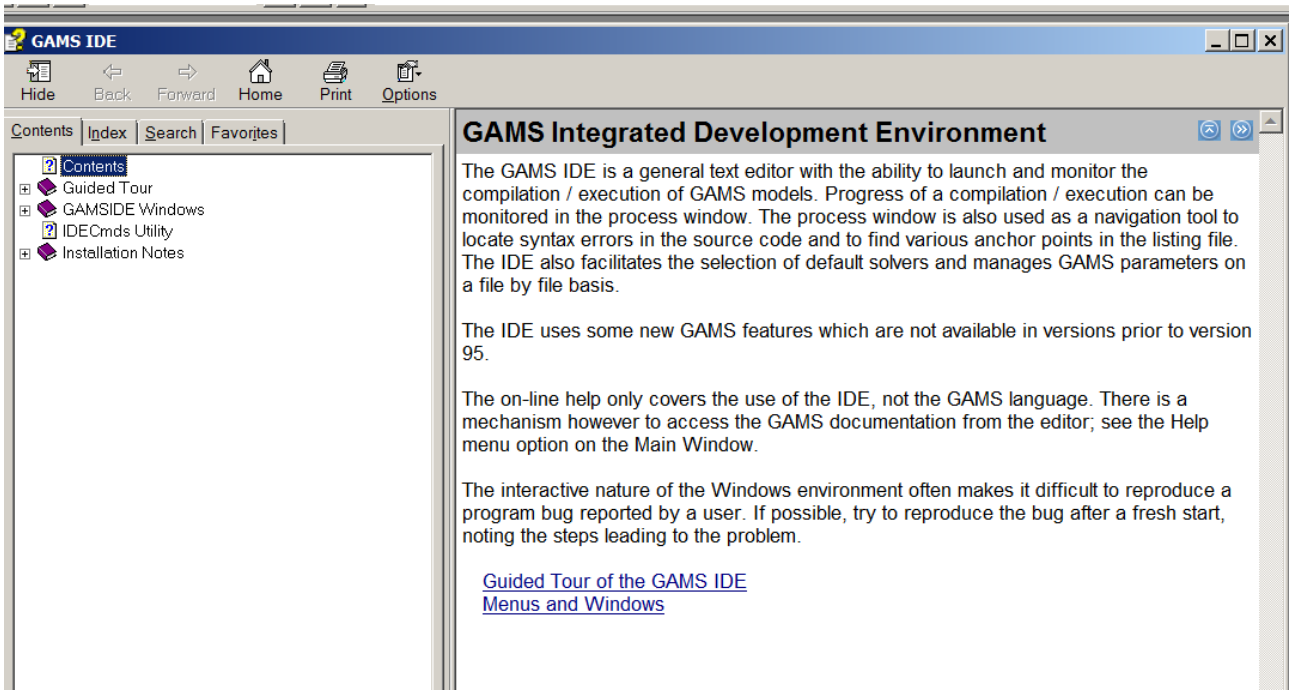
Most of the menu choices are similar to those found in other Windows based programmes. You should also familiarise yourself with the various buttons on the toolbars – these have pop-up descriptions.

- The `File` menu has standard `open/close, save, print` etc., functions. It also has functions to allow the running of programmes, run and compile.
- The `Edit` menu has standard `copy, paste, replace` etc., functions.
- The `Search` menu has standard `find, replace and GoTo` functions. It also has a function to `match parenthesis` – this is very useful.
- The `Windows` menu is about customising the layout of windows in the interface.
- The `Model Libraries` menu is one method for accessing the libraries of programmes provided with the GAMS software.
- The `Utilities` menu allows access to a couple of specialized utilities
- The `Help` menu leads to a standard hypertext based help system. You should use it.



GAMSIDE is relatively well documented in the associated Help file that can be accessed from the Help menu. Choose `HELP > Help Topics,` which provides a standard Windows HTML help system. New users should make use of this facility since it provides

both an introduction to GAMSIDE and documents numerous useful facilities provided by GAMSIDE.



Configuring GAMSIDE

While GAMSIDE looks and behaves like a standard Windows programme, and therefore it is not hard to make adjustments, there are four ‘commands’ that it is helpful to know about from the beginning since they make it much easier to use GAMS.

Option Settings

Among other things the option settings control the display. It is useful to make some adjustments to the defaults immediately.

Choose `Options` from the `File` menu. There are some settings you should make here

- In the `Editor` page
  - set the `font` and `fontsize` – use a fixed pitch font, e.g., `Courier New`, and choose a fontsize that suits you (the choice is inevitably a compromise between having enough information visible on screen and a fontsize that is easy for you to read).

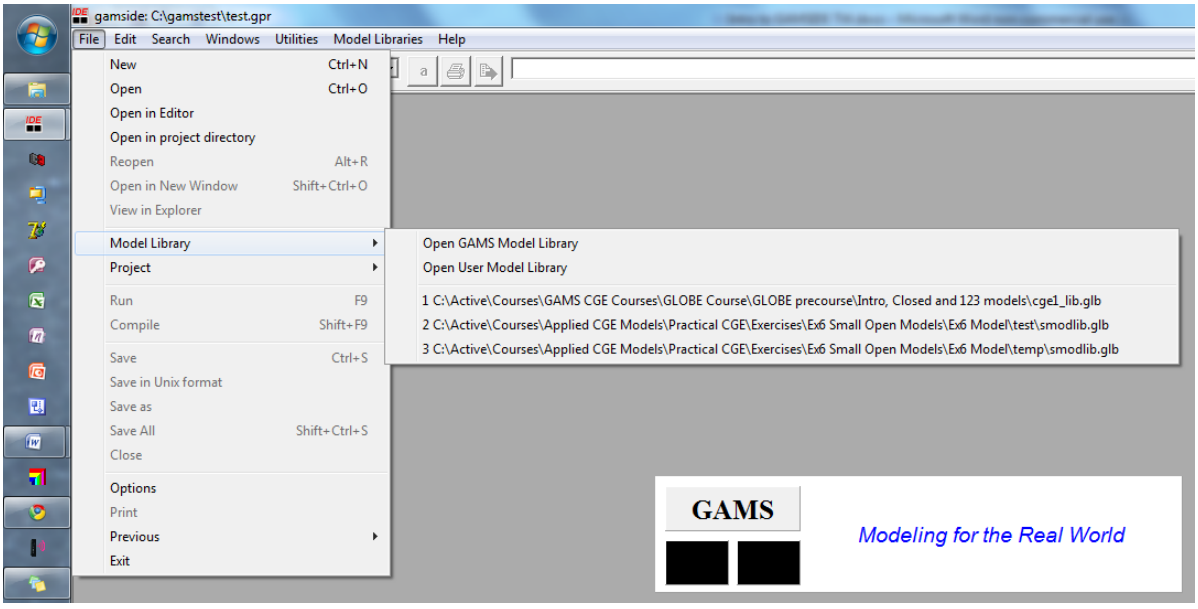
- In the `GAMS file extensions box` add "`inc,dat`" – this ensures that all listed file types use syntax colouring and other standard layout features.

- In the `Output` page set the `Page Height` to (say) 99999 – this reduces the amount of unnecessary output in the list file.

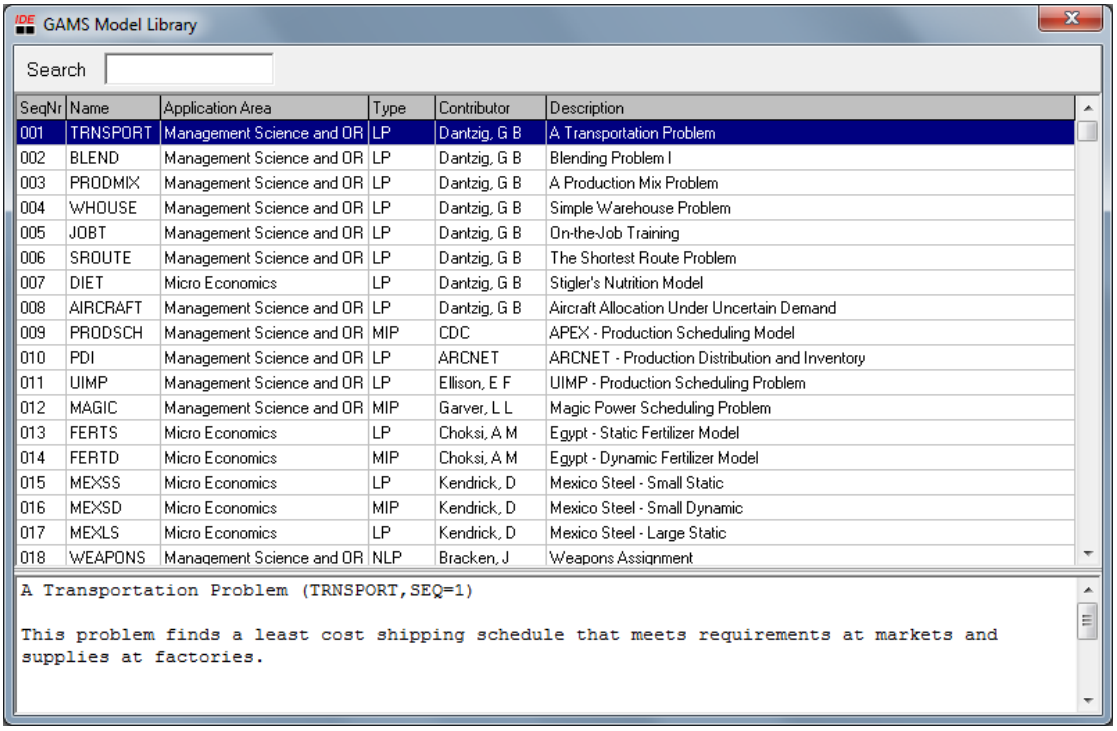
Leave the other settings for now, although you can adjust them later as you become more familiar with GAMSIDE.

Model Library

GAMS comes with a large model library. Go to the `File` menu and select `Model Library`, and then from the sub menu select `Open GAMS Model Library`. A listing of sample GAMS files will appear, and when each is selected a brief description will come up at the bottom of the window.

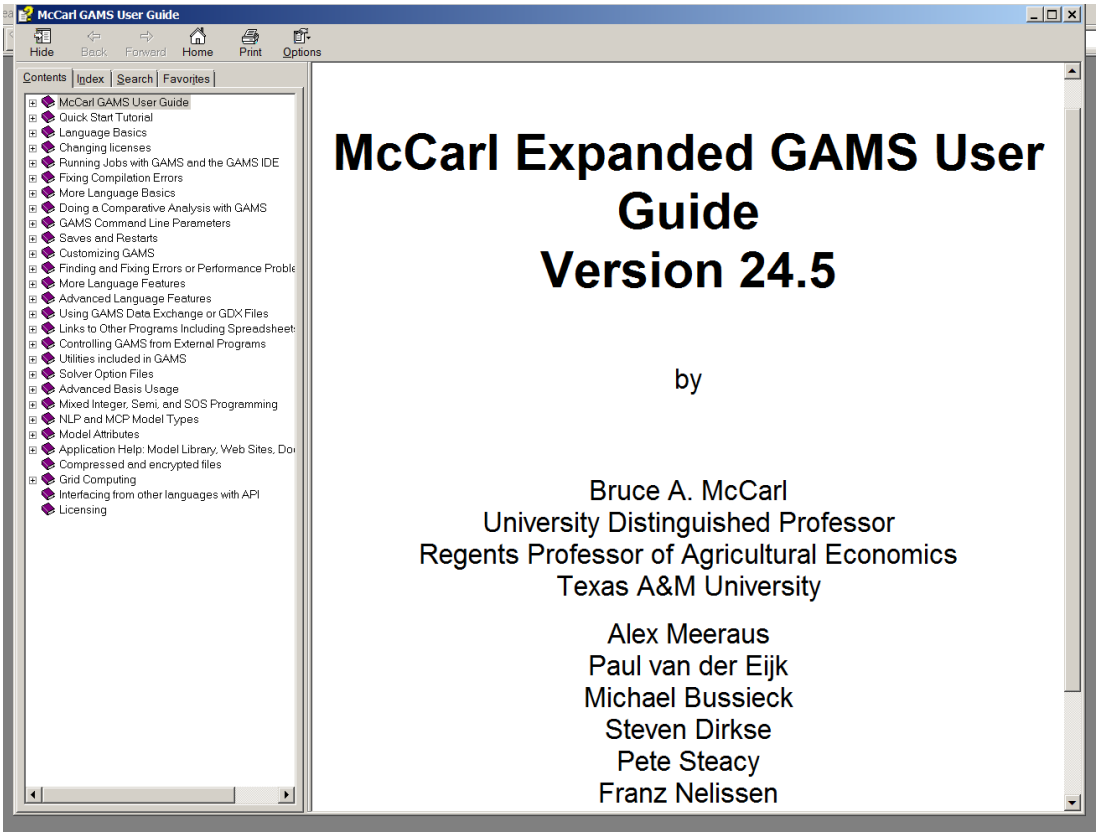


If you click on the column title the library will be sorted in the alphanumeric order of the entries in that column. Click on the column `'SeqNr'`, the first entry in that column will now be `'001'` with the name `'TRNSPORT'` (see the screenshot below). Double click somewhere on the first row and the file `'TRNSPORT.gms'` will open in the current GAMS window.



GAMS Documents

GAMSIDE comes with a help file, and a full set of GAMS manuals in electronic (Adobe Acrobat) format. You may find it useful to print a copy of the manuals for your own use, at the least you should have a copy of Chapter 2 – The GAMS Tutorial. GAMS now also includes a copy of the McCarl Guide in HTML Help format – this is very useful.



Printing

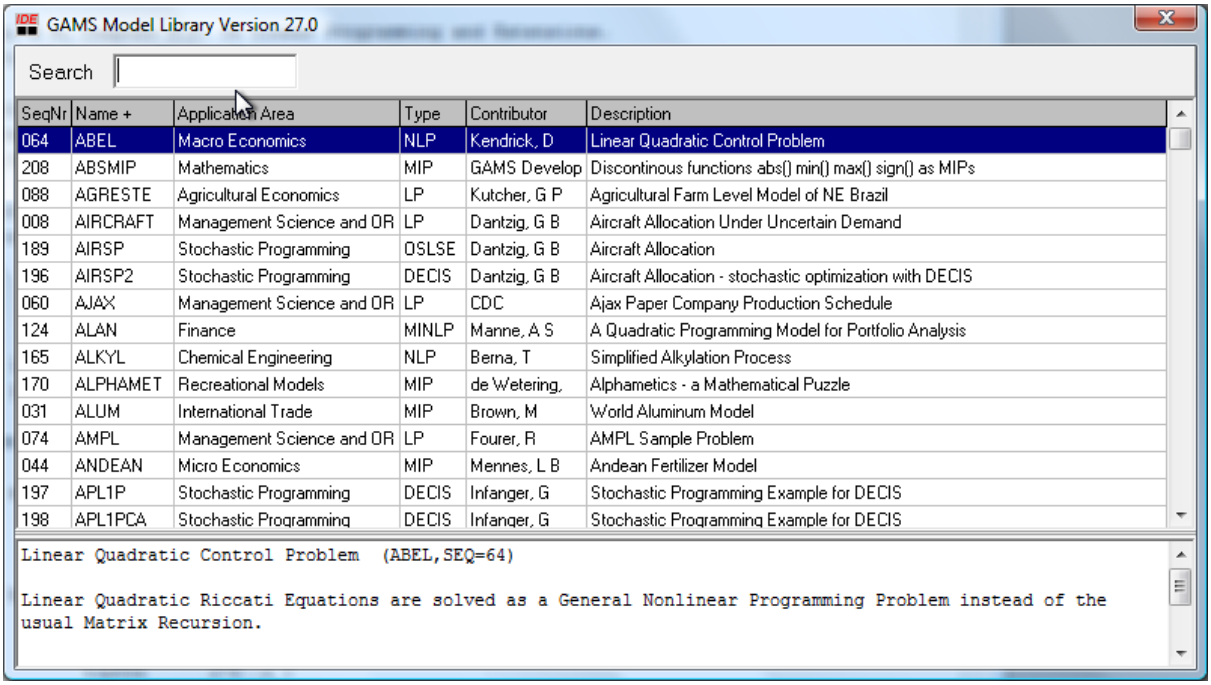
The `Print` command, which is on the `File` menu, has a couple of features that it is useful to know about. The font size and type can be set in the `Print and Review` box – make sure you use a fixed pitch font, e.g., `Courier New`. You can also set the print option so that 2 pages are printed on each sheet of paper in a landscape layout by selecting the `Two Pages` option – since GAMS files and output can be extensive this option can make real savings.

Testing a GAMS/GAMSIDE Installation

In order to test your GAMS installation you should run a number of sample programmes provided by GAMS in the GAMS Model library.[18] These programmes will all use the `'gamstest'` directory that has been created.

[18] This can be done simply by exploiting the fact that on a first installation GAMS installs a project sub directory together with a project file to control path. More details on projects and project directories are given below.

In GAMSIDE select the `File > Model Library > Open GAMS Model Library`, which will open the following window.



To text the installation it is recommended by GAMS to run 6 programmes. These are

1. `trnsport` (LP :objective value: 153.675)
2. `chenery` (NLP: objective value: 1058.9)
3. `bid` (MIP: optimal solution: 15210109.512)
4. `procsel` (MINLP: optimal solution: 1.9231)
5. `scarfmcp` (MCP: no objective function)
6. `scarfmge` (MPSGE: no objective function)

In turn find each programme in the Name column (the search box can be used to help) and double click of the name. This will download a copy of the programme to the default project directory. Then run the programme by selecting `File > Run`, pressing F9 or clicking on the icon  . If you are asked to select the default solvers accept the default settings.[19]

[19] It is possible to change the default settings later through the `options` menu. The default solvers simply determine the solver used to solve the type of programme, e.g., LP, NLP, MIP, MCP, etc., being run UNLESS a different choice is defined in the programme file.

The progress of a submitted programme is recorded in an `active process` window. The information from the process window is recorded as `[Filename].log` and saved in the project directory.

In each case use the output reported in the `active process` window to verify that output says `'*** Status: Normal Completion'` in blue at the bottom and that the reported (optimal) `Objective` value is the same as that given in the list above.

Running a First GAMS Programme

Setting up Your Project

Before working on **your** first GAMS programme it is necessary to set up/create a directory to contain the programme and result files and a project file.[20] The project file is needed so that GAMS knows where, i.e., in which directory, to find files and where to write out the results. It is assumed here that you will be following the exercises provided with this training programme.

- Using Windows Explorer create a directory on your base drive and call it `'GMS-training'`.
- In the directory `'GMS-training'` create another directory and call it `'trans'`.
- Open GAMSIDE.
- In the directory `trans` create a project file called `trans` – note that the `file type` must be GAMS Project file, i.e., `trans.gpr`.
- GAMS will open either with (i) a dialogue box asking you to create a new project (if this is the first time you have used this installation) or with (ii) the last project file. In each case the process is slightly different:
    - When a dialogue box appears: select the destination directory and then type the project file name in the filename box.

[20] Strictly this is not true, because while installing GAMS a directory (gamsdir) will have been created in the Windows directory, and this can be used in conjunction with the GAMS Model Library, which is what was suggested above for testing the GAMS installation. The approach advocated in the text has the advantage of being general.

  - When a previous project open: from the `File` menu select `Project` and then `New Project`, then select the destination directory and then type the project file name in the filename box.
- The project name and file path will now appear in the title bar at the top of the screen.
- From the `File` menu select `>Model Library>Open GAMS Model Library`, which will open up a window. Click on the column header `SeqNr`, which will sort the entries by their sequence number. The 'files' wanted are for the `TRNSPORT` model, so double click on the top row, i.e., SeqNr `'001'`, and Name `'TRNSPORT'`. The file `'trnsport.gms'` will automatically open in GAMSIDE. Now check the contents of the directory `'trans'`.

You are now ready to work with your first GAMS programme.

Running a GAMS Programme

To run a GAMS programme file select `Run` from the `File` menu, or `F9` or the `Run GAMS` button  on the toolbar. The run command will submit the programme file for compilation and, presuming it compiles without error, execute the programme. If you wish solely to implement the compilation stage choose `Compile` from the `File` menu, or `Shift F9`.

- The progress of a submitted programme is recorded in an `active process` window. Information recorded in that window is very useful and provides an easy way to debug a programme file (see below). The information from the process window is recorded as `[Filename].log` and saved in the project directory.
- The layout of the various windows is a matter of personal choice. One layout that is relatively easy to use is `Tile Vertical` from the `Window` menu.
- The GAMS output file is returned automatically as a tabbed file in the editor window as `'trnsport.lst'`. (You can choose for this to not happen by changing the settings in the `File > Options` menu.)

Debugging a GAMS Programme

Computer programmes nearly always have numerous errors. GAMSIDE provides easy ways to find where the errors occur, especially compilation errors.

- Compilation error messages appear in RED in the `active process` window. `Click` on these messages and the editor window for the `[Filename].gms` file is chosen and the cursor moves to where the error was noticed. `Shift Click` on these messages and the editor window for the *.`lst` file is chosen and the cursor moves to where the error was noticed.
- In the `[Filename].lst` file you will find information about the type of error. The error is marked by **** and a $#, where # is a number, on the line below where the error was noticed. More information on the meanings of the $# codes is given at the end of the list file (search for the string '`Error Messages`'). The information given is (usually) very helpful.

Compilation Errors

Compilation errors are essentially syntax errors. In the listing file GAMS will provide useful suggestions about the likely cause of the error message. The listing file also contains markers that make it easy to find the compilation errors: search for the text string "****".

Some general principles may be helpful

- start from the top of the programme and work down;
- solve each error as it appears - do not skip onto the next error without a good **programming** reason;
- do not make too many changes at a time - as you become more familiar with GAMS and compilation errors the number of errors corrected at each stage will increase, but when starting it is easy to compound errors;
- if substantial changes are made to the code save the input file with a revised name before running the file.

Common syntax errors include the following

- failing to end an operation with a ";" - GAMS often identifies this type of error as occurring on the next line of code or at the next keyword;
- assigning or using a parameter or variable before it has been declared;
- using a parameter or variable before it has been declared or assigned;
- spelling mistakes;

- the "*" used for comments and/or to comment out lines of code is NOT the first item in the line of code (NB: a space is an item in a line of code);
- set operations trying to use sets that are already under control - this is where the aliases become very useful.

The solutions to most syntax problems are relatively simple. Controlling sets is one type of syntax error that can prove somewhat less tractable.

Execution Errors

Execution errors are trickier. They can often arise because the model has been incorrectly specified. There are three basic consistency checks in the CGE models that can be used to eliminate common execution errors:

- The WALRAS variable should return a zero value - if it does not it indicates an error or errors in the system of equations.
- The model should return a consistent and balanced SAM which is identical to the SAM database. Identifying the rows and columns that either do not balance or return values different to those in the database should help to identify the equations containing errors.
- Since the model should be homogeneous of degree zero in prices it should solve for relative prices. Hence if all prices increase by some constant factor the **real** variables should remain unchanged. If the model is not homogeneous of degree zero in prices the real variables will be changed. If the initial model solution does not return unit prices it indicates a problem.

There are some ways to help find execution errors.

- DISPLAY statements for parameters and initial values for variables can be used to check the values returned by the programme.
- The listing of values for variables from the model should be consistent with the values from the database.
- If the model is consistent the left-hand and right-hand sides of the equations should equate, or at least contain no "significant" discrepancies. This can be checked in the equation listing (controlled by the "limrow" and "limcol" options linked to the solve statement). Searching for the text string "***" is the quick way to find

errors. A discrepancy is indicated by the statement "LHS = <value>", where the error is only likely to be important if the value is greater than about 1.0E-5. This indicates that the problem is probably associated with the definitions of the parameters and variables in that equation.

As with compilation errors it is unwise to try and do too much at once. If substantial changes are made to the code it is wise to do so in a new version of the input file.

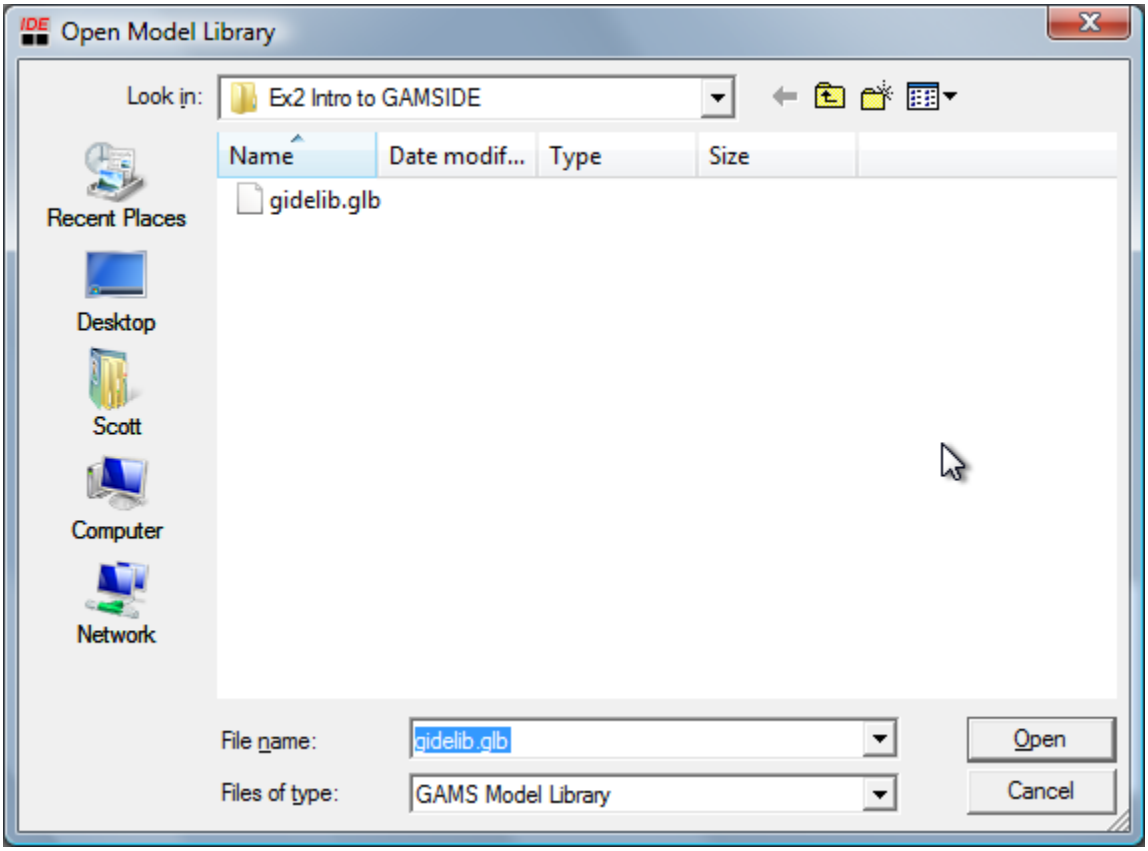
Using a GAMS User Model Library

The 'User Model Library' facility in GAMSIDE provides an indexing facility whereby collections of models can be archived and then easily accessed from GAMSIDE. All the files contained within a Model Library are stored in a single directory and accessed via a simple Library manager that is accessed from GAMSIDE. The Library manager allows the user to download a file, OR selection of files, to the **current** working directory according to the criteria specified by the complier of the library. GAMS uses this facility, in the form of the GAMS Model Library, to make available a wide range of GAMS models that illustrate the various capabilities of the GAMS language (the GAMS Model Library only downloads single files to the current working directory).

Generic details about the GAMS Model Library and User Model Libraries are provided in the McCarl guide that is distributed with GAMS/GAMSIDE; to access the McCarl guide choose `Help > Expanded GAMS Guide (McCarl)`, which opens the guide in HTML Help format.

The programmes made available as part of this Introduction to GAMS/GAMSIDE are all supplied as one or more WinZip archives. Each archive should be unzipped into its own directory – suggested names are given with each library and then these names are used in the documentation. For each library the Sequence number (SeqNr) details the order in which it is assumed the exercises will be conducted.

To open a User Model Library, choose `File > Model Library > Open User Model Library` and then search for the directory into which the library was unzipped and open the `library_name.glb` file



This would produce the following User Library manager.