

# An Introduction to (Student) GAMS and GAMSIDE

(General Algebraic Modelling System)

## Table of Contents

1. Introduction.....	2
2. Installing GAMS 2.50 with GAMSIDE.....	4
3. GAMSIDE .....	7
Using GAMS 2.50 with GAMS IDE .....	7
Creating a Project File .....	7
Preparing a GAMS Programme File.....	10
Configuring GAMSIDE.....	11
Model Library .....	12
GAMS Documents.....	13
Printing.....	14
4. Testing a GAMS/GAMSIDE Installation .....	15
5. Running a First GAMS Programme .....	17
Setting up Your Project.....	17
Running a GAMS Programme.....	18
6. Notes on Debugging a GAMS Programme .....	20
Compilation Errors.....	20
Execution Errors .....	21
7. Using a GAMS User Model Library.....	23
Some Tips .....	26

## 1. Introduction

The GAMS (General Algebraic Modelling System) (version 2.50) suite consists of the base GAMS module and a collection of solvers. An interface for Windows based PCs, GAMSIDE, is also available.<sup>1</sup> 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

---

<sup>1</sup> There are versions of GAMS for UNIX, LINUX, PowerPC and Mac OS X. The code created by any user on any of the platforms can (should) transfer to any other of the supported platforms. However the utilities from GAMS are most developed for the Windows platform, which is also the most common platform, and hence these exercises all assume the user is on a Windows platform.

- 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
- XAlinear and mixed integer solver
- DICOPT Mixed integer Nonlinear Programming
- MILES Mixed complementarity problem solver
- PATH Mixed complementarity problem solver
- PATHNLPNLP

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.

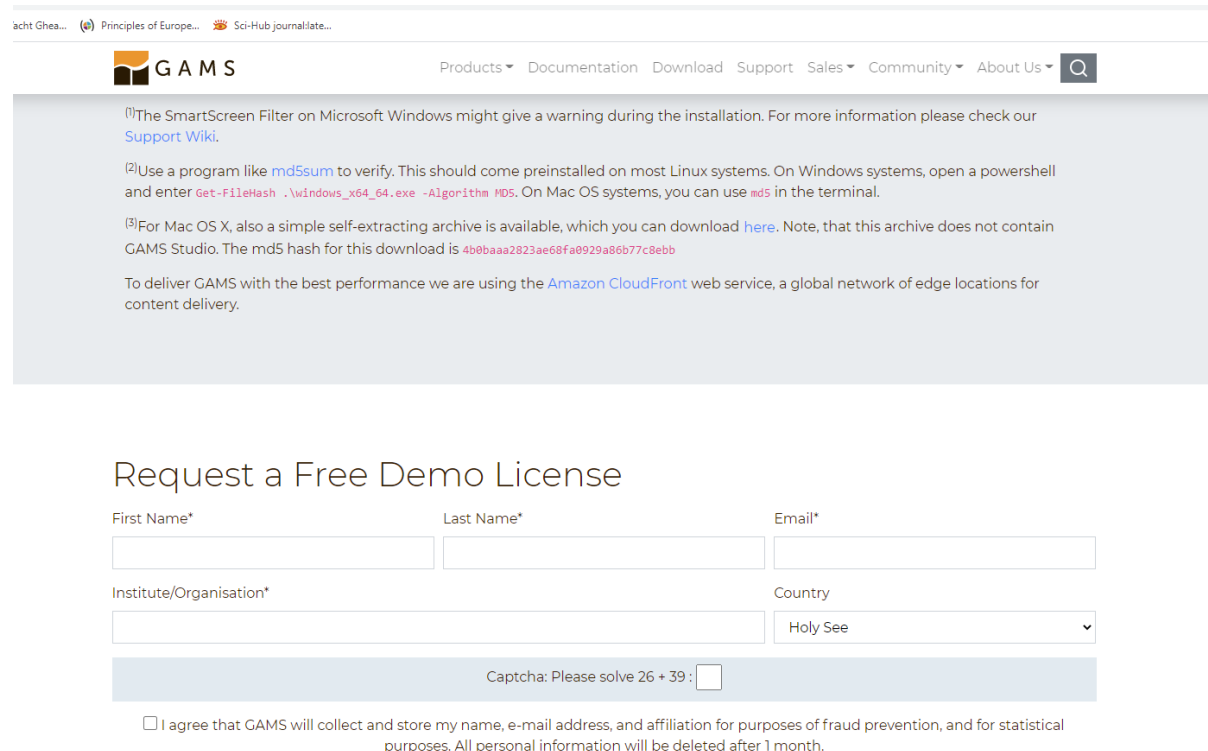
You can write/read GAMS files in any text editor (any DOS or Windows editor, even Word in txt mode).

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.

## 2. Installing GAMS 2.50 with GAMSIDE

### Getting a Demo License

GAMS operates a demo license scheme that requires completing an application form of the GAMS website. Go to [www.gams.com/download](http://www.gams.com/download), scroll down to find the request form (see figure). GAMS will send you a license file.<sup>2</sup>



The screenshot shows the GAMS website interface. At the top, there is a navigation bar with links: Products, Documentation, Download, Support, Sales, Community, and About Us. Below the navigation bar, there is a section titled 'Request a Free Demo License'. This section contains a form with the following fields:

- First Name\* (text input)
- Last Name\* (text input)
- Email\* (text input)
- Institute/Organisation\* (text input)
- Country (dropdown menu, currently showing 'Holy See')
- Captcha: Please solve 26 + 39 = (input box)

Below the form, there is a checkbox with the text: 'I agree that GAMS will collect and store my name, e-mail address, and affiliation for purposes of fraud prevention, and for statistical purposes. All personal information will be deleted after 1 month.'

### Download the GAMS Software

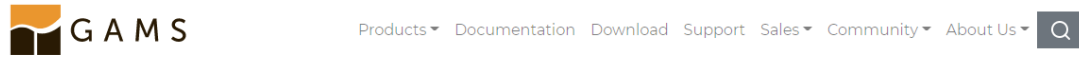
To download the software go to [www.gams.com/download](http://www.gams.com/download), and download the MS Windows version of GAMS. NB: the windows version runs with 64-bit architecture. While the GAMS software can be run with three different architectures – Windows, Linux and Mac OS X – the description here relates to the Windows architecture, which is the only architecture that supports GAMSIDE.<sup>3</sup> The GAMS software will be downloaded as a single exe file. Store this file together with the demo license file acquired from GAMS.

<sup>2</sup> The demo version of GAMS is identical to the full version except for the fact that the licence file restricts the size of model that can be implemented and is time limited.

<sup>3</sup> We are constrained to use GAMSIDE because GAMS Studio does not currently allow the use of User Model Libraries in a manner that is acceptable. GAMS Studio is offered for all three architectures.

Note that the GAMS download page also includes ‘release notes’, ‘detailed platform descriptions’ and ‘installation notes’.

We assume you will use the latest **major** release.



## Download GAMS Release 36.1.0

Released August 02, 2021

Please consult the [release notes](#) before downloading a system. We also have [detailed platform descriptions](#) and [installation notes](#). The GAMS distribution includes the [documentation](#) in electronic form.

MS-Windows	Linux	Mac OS X
Microsoft Desktop and Server Operating Systems <sup>1</sup>	GNU/Linux System	Package Installer for Mac <sup>3</sup>
x86_64 architecture	x86_64 architecture	x86_64 architecture
MD5 hash <sup>2</sup> 45ca2dab776f40cb33b0b6120d40b9c9	MD5 hash <sup>2</sup> 4aba509d6953e3256cc7465af37f27a7	MD5 hash <sup>2</sup> fe091e814ec16c4fe222962c265142e0
<a href="#">Download</a>	<a href="#">Download</a>	<a href="#">Download</a>

## Installing GAMS

The installation process is typical for Windows applications. Double click on the exe file, windows\_x86\_64.exe, and follow the on screen instructions referring at each stage to the instructions below.

- i) We recommend 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. If not you will need to adjust the path appropriately.)
- ii) Select the ‘classic’ GAMS IDE option in the ‘File Association of GAMS Files’.
- iii) You will be asked for a license file. Select ‘Browse for license file’ and direct the search to the location where you saved the demo license.

*Practical CGE Modelling: Introduction to GAMS and GAMSIDE*

- iv) If you have not acquired a license file you can continue with the installation of GAMS. Then when you get a demo license you can copy the license into the directory `c : \GAMS`.
- v) The installation will conclude by offering the option to launch GAMS IDE; select this option.
- vi) 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.

### 3. GAMSIDE

*(General Algebraic Modelling System Integrated Development Environment)*

#### Using GAMS with GAMS IDE

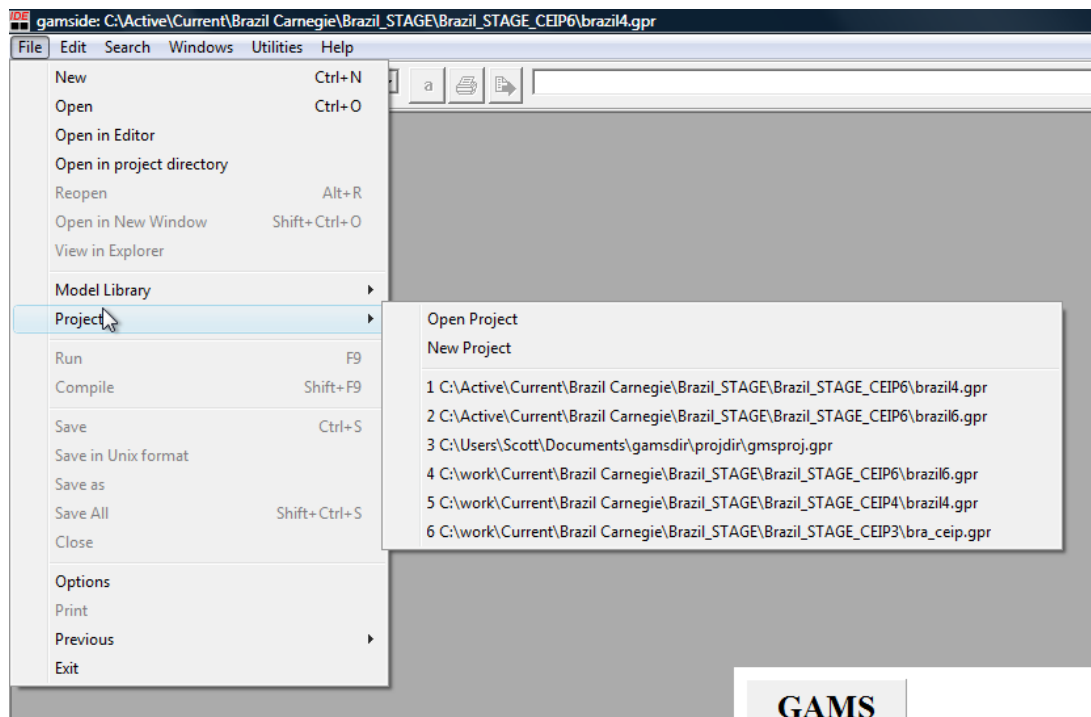
To use GAMS you need a programme file. All GAMS programme files are prepared as text files and saved as `[filename].gms`.<sup>4</sup> To prepare a GAMS programme it is necessary to use a text file editor. In the past it was necessary to use a text file editor to write the programme file and read the 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

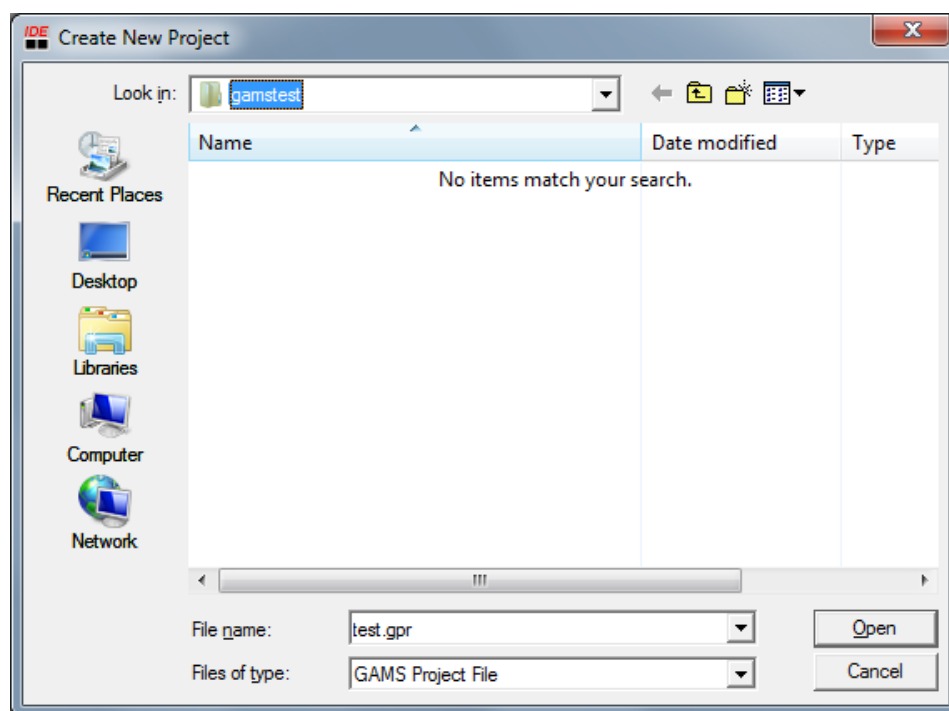
GAMSIDE uses a project file (`[project_name].grp`) to manage paths and keep track of files; the project file tells a GAMS programme the default location of files needed to run the programme and where GAMS should send the files generated by the programme. It cannot be recommended strongly enough that 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.

---

<sup>4</sup> 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.

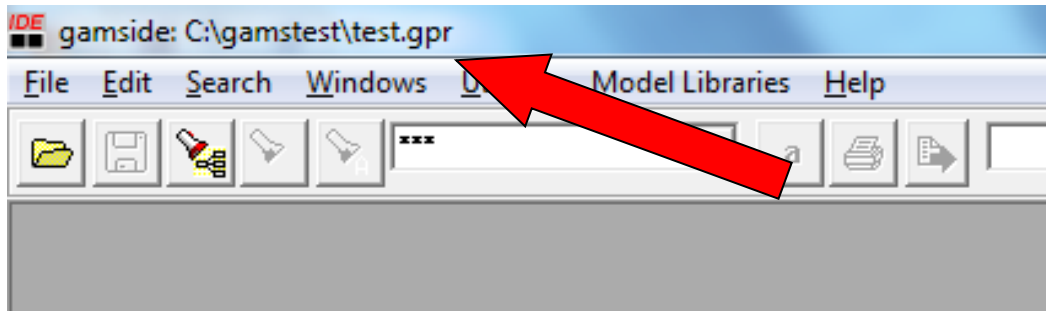


From File, Project choose New Project and a 'standard' window (see below) will appear. Navigate to the top level of your C drive – or an alternative used for storing data on your PC – and create a directory called 'gamstest', in which a project file can be created by typing 'test.gpr' in the File name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

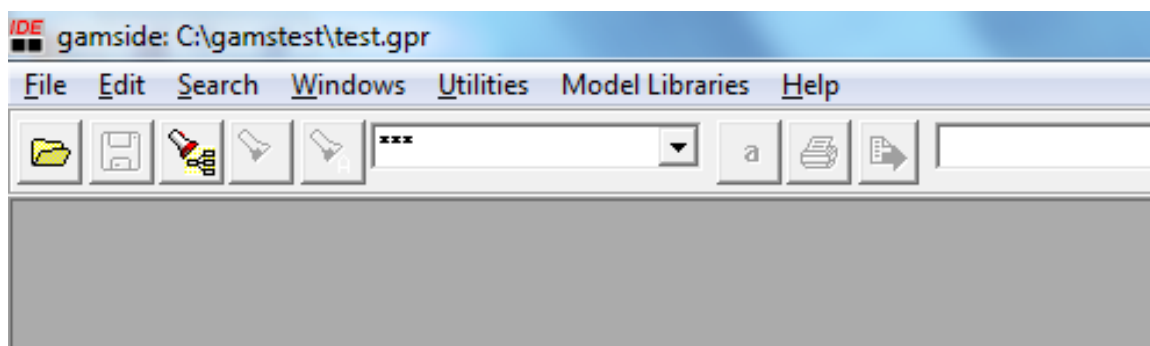
files after the session of work has been completed; deleting them ‘manually’, i.e., one by one, is slow and the possibility of making errors is high so it is necessary to be careful.<sup>5</sup>

### 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.

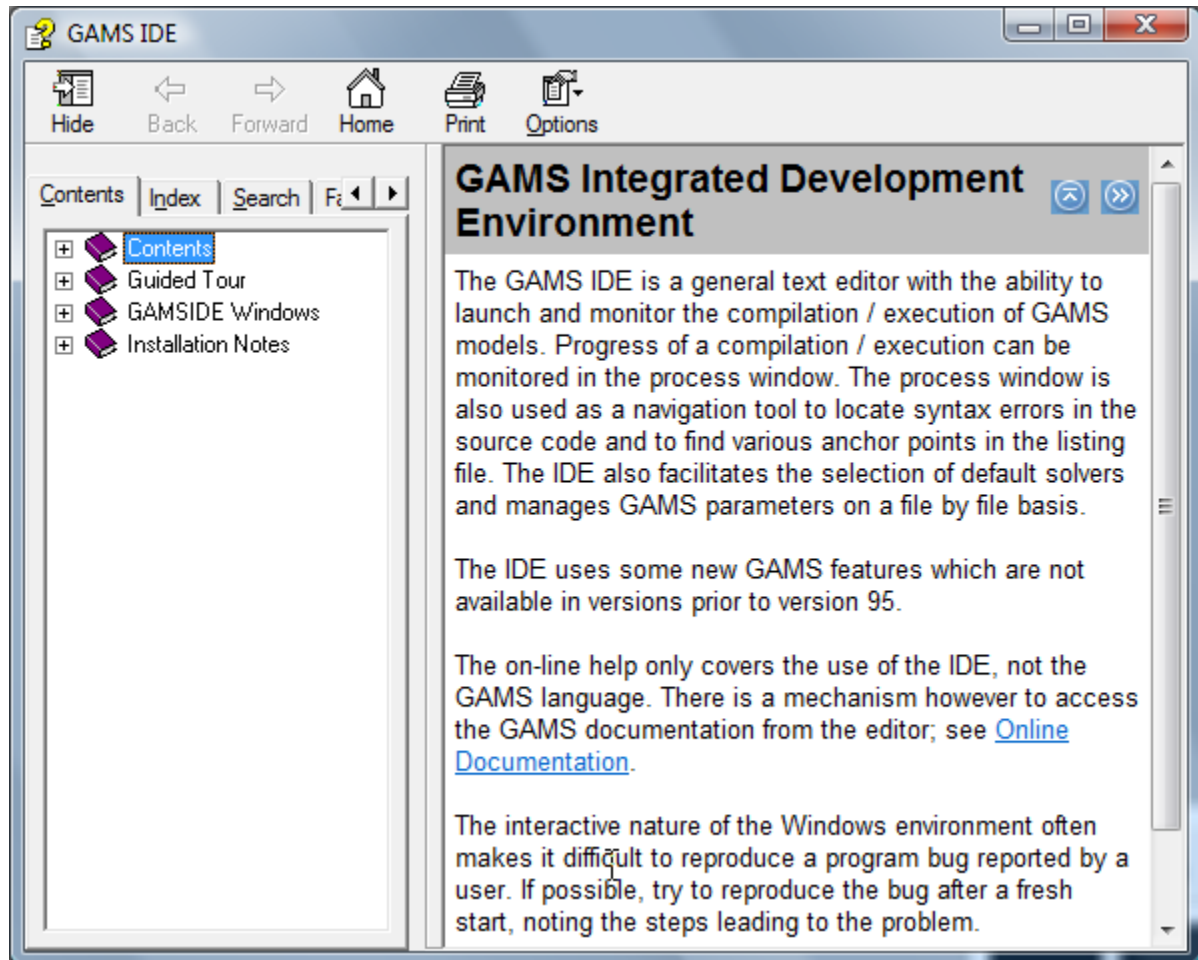
- The `File` menu has standard `open/close`, `save`, `print` etc., functions. It also has functions to allow the running of programmes, `run` and `compile`.
- The `Edit` menu has standard `copy`, `paste`, `replace` etc., functions.
- The `Search` menu has standard `find`, `replace` and `GoTo` functions. 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 `Utilities` menu allows access to a couple of specialized utilities.
- The `Model Libraries` menu is one method for accessing the libraries of programmes provided with the GAMS software.
- 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

<sup>5</sup> A simple batch file to delete files that satisfy defined criteria is one way to avoid simple errors.

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.



The Help menu also offered options for accessing help from the GAMS Documentation (online and offline) and the McCarl guide

### Configuring GAMSIDE

While GAMSIDE looks and behaves like a standard Windows programme, and therefore it is not hard to make adjustments, there are some option setting 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.

## Practical CGE Modelling: Introduction to GAMS and GAMSIDE

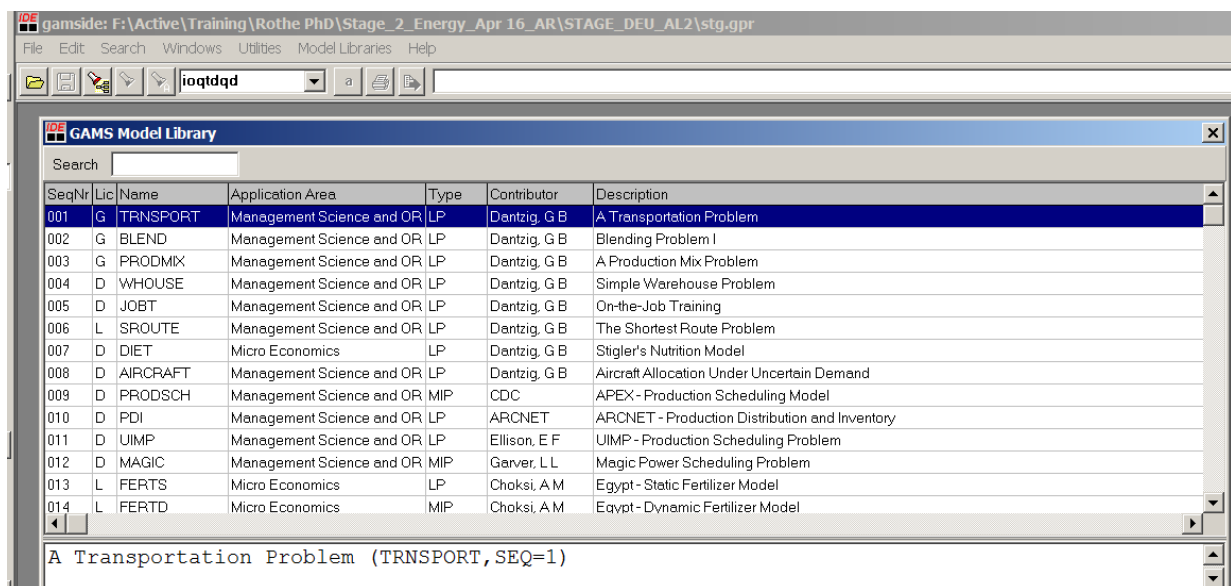
Choose Options from the File menu. There are some settings you should make

- 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” – 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.<sup>6</sup>

### Model Library

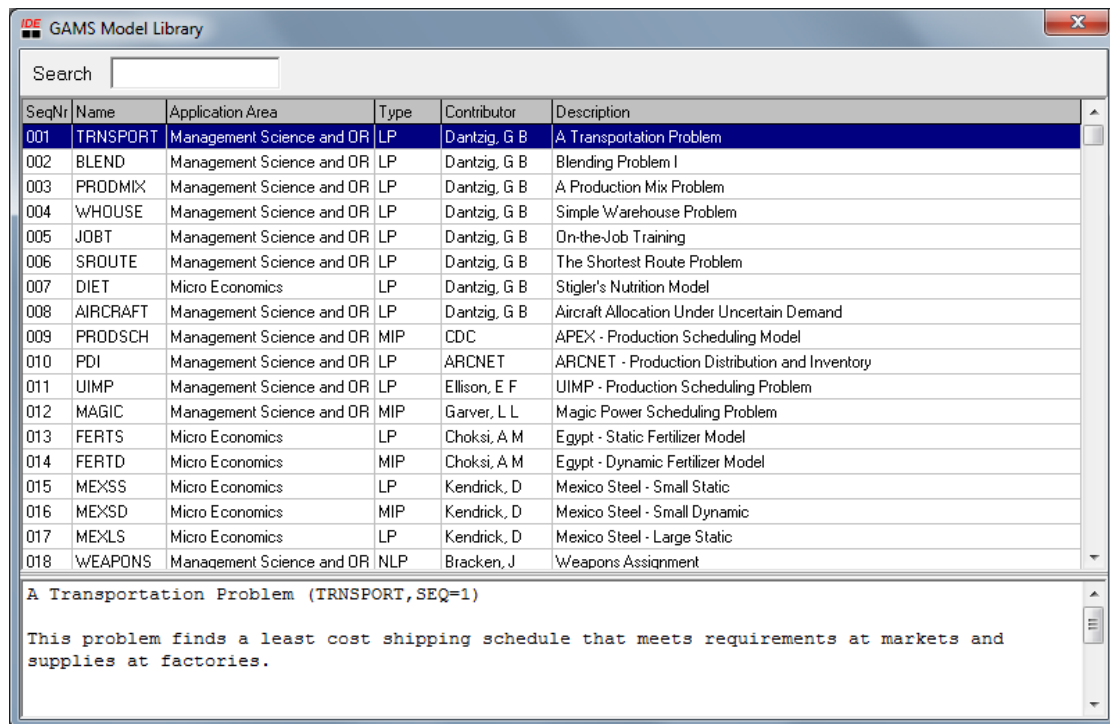
GAMS comes with several large model libraries. Go to the Model Libraries menu and select GAMS Model Library. A listing of sample GAMS files will appear, and when each is selected a brief description will come up at the bottom of the window.



<sup>6</sup> We will use these settings during the online course. During an on-site course we use ‘standard’ option settings; a common appearance makes it much easier for an instructor to provide support across multiple users.

## Practical CGE Modelling: Introduction to GAMS and GAMSIDE

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 'TRANSPORT' (see the screenshot below). Double click somewhere on the first row and the file 'TRANSPORT.gms' will open in the current GAMS window.



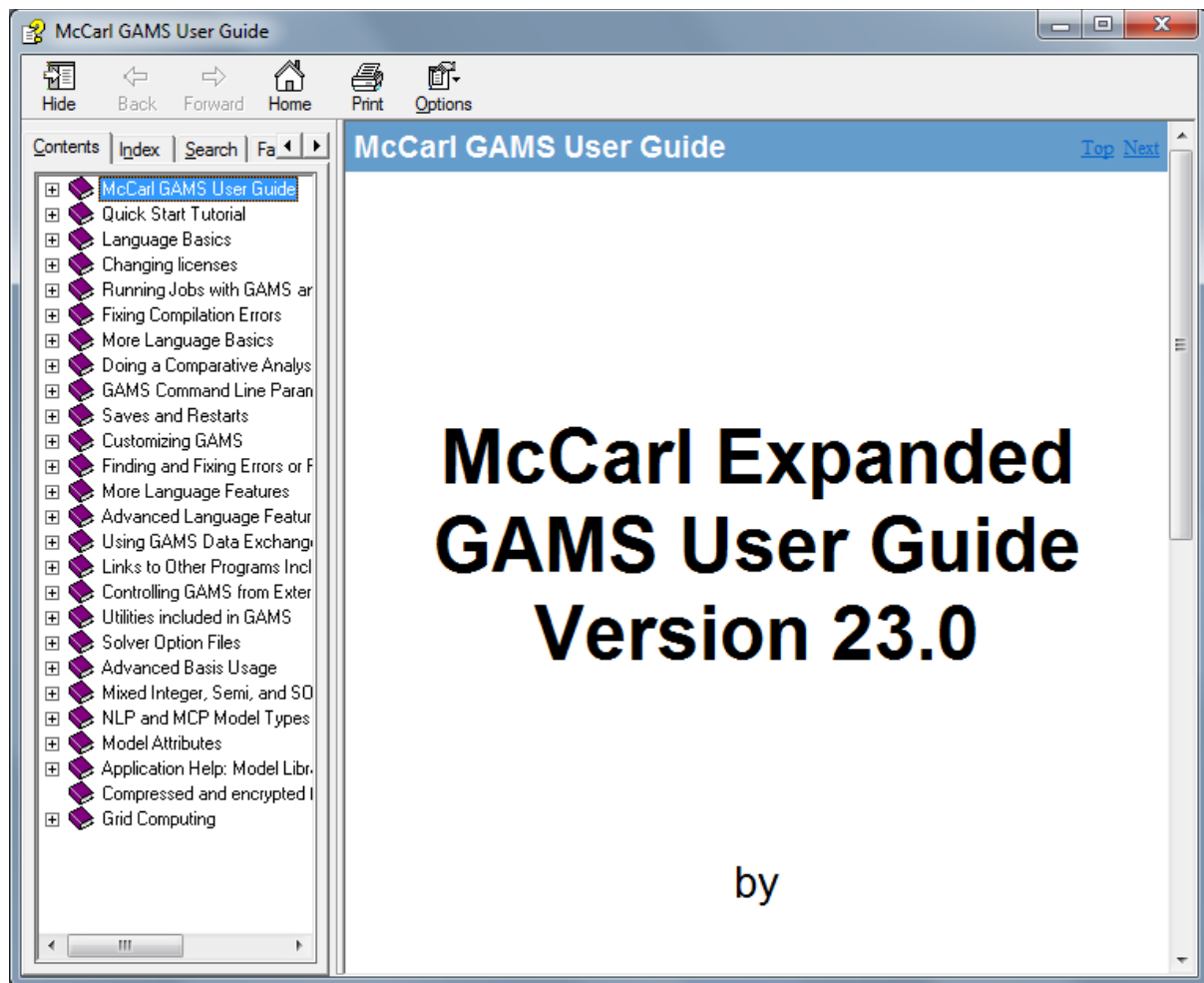
SeqNr	Name	Application Area	Type	Contributor	Description
001	TRANSPORT	Management Science and OR	LP	Dantzig, G B	A Transportation Problem
002	BLEND	Management Science and OR	LP	Dantzig, G B	Blending Problem I
003	PRODMIX	Management Science and OR	LP	Dantzig, G B	A Production Mix Problem
004	WHOUSE	Management Science and OR	LP	Dantzig, G B	Simple Warehouse Problem
005	JOB	Management Science and OR	LP	Dantzig, G B	On-the-Job Training
006	SRROUTE	Management Science and OR	LP	Dantzig, G B	The Shortest Route Problem
007	DIET	Micro Economics	LP	Dantzig, G B	Stigler's Nutrition Model
008	AIRCRAFT	Management Science and OR	LP	Dantzig, G B	Aircraft Allocation Under Uncertain Demand
009	PRODSCH	Management Science and OR	MIP	CDC	APEX - Production Scheduling Model
010	PDI	Management Science and OR	LP	ARCNET	ARCNET - Production Distribution and Inventory
011	UIMP	Management Science and OR	LP	Ellison, E F	UIMP - Production Scheduling Problem
012	MAGIC	Management Science and OR	MIP	Garver, L L	Magic Power Scheduling Problem
013	FERTS	Micro Economics	LP	Choksi, A M	Egypt - Static Fertilizer Model
014	FERTD	Micro Economics	MIP	Choksi, A M	Egypt - Dynamic Fertilizer Model
015	MEXSS	Micro Economics	LP	Kendrick, D	Mexico Steel - Small Static
016	MEXSD	Micro Economics	MIP	Kendrick, D	Mexico Steel - Small Dynamic
017	MEXLS	Micro Economics	LP	Kendrick, D	Mexico Steel - Large Static
018	WEAPONS	Management Science and OR	NLP	Bracken, J	Weapons Assignment

A Transportation Problem (TRANSPORT,SEQ=1)

This problem finds a least cost shipping schedule that meets requirements at markets and supplies at factories.

## 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.



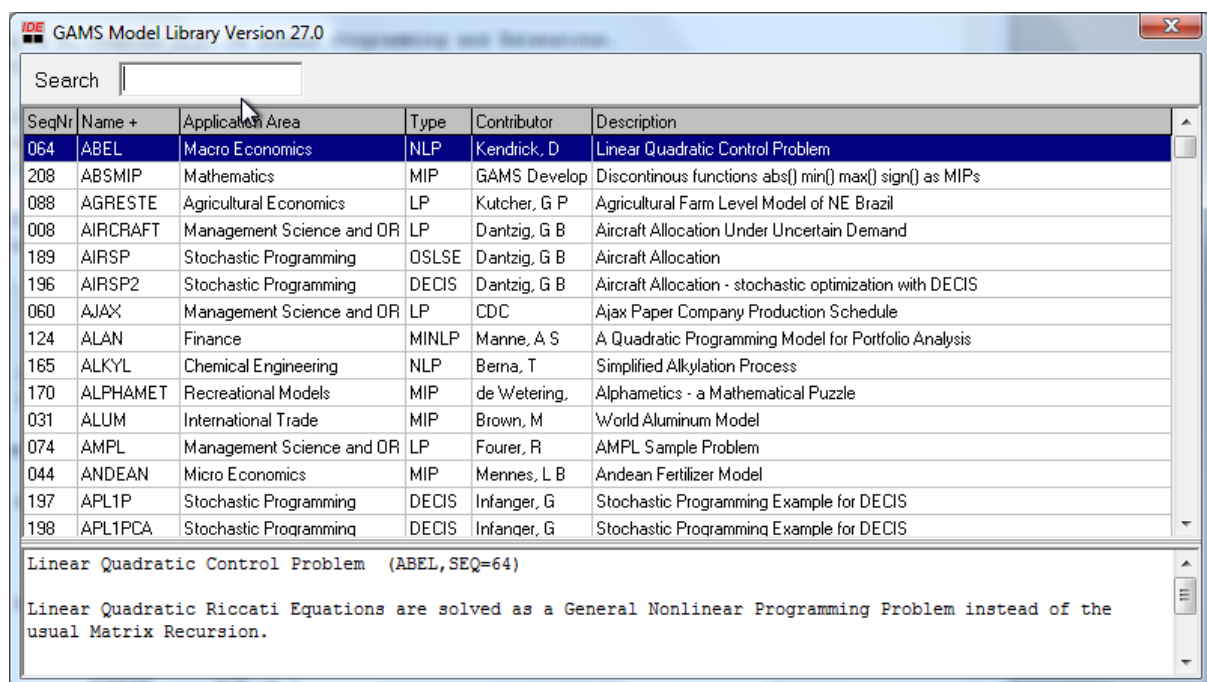
## Printing

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.

## 4. 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.<sup>7</sup> These programmes will all use the 'gamstest' directory that has been created.

In GAMSIDE select the Model Libraries > GAMS Model Library, which will open the following window.




To test 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

<sup>7</sup> 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.

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.<sup>8</sup>

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.

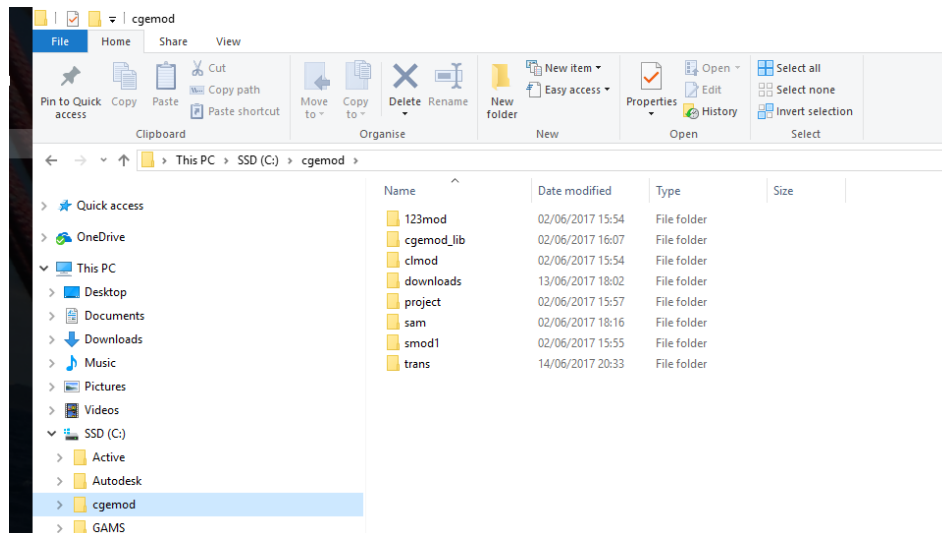
---

<sup>8</sup> 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.



## 5. Running a First GAMS Programme

From now on all the data, code and exercises for this course will be stored in a single master directory on your PC. This directory will be called `cgemod`. As the course progresses a number of sub directories will be added and each sub directory will also have sub directories. At the end of the course it will look something like this



### 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.<sup>9</sup> 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 '`cgemod`' .
- In the directory '`cgemod`' create a subdirectory and call it '`trans`', and in the '`trans`' directory create a subdirectory called '`trans1`' .
- Open GAMSIDE.
- In the directory `trans1` create a project file called `trans` – note that the file type must be GAMS Project file, e.g., `trans.gpr`.


<sup>9</sup> 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.

### *Practical CGE Modelling: Introduction to GAMS and GAMSIDE*

- 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.
  - When a previous project opens: 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 `Model Libraries` menu select `>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 `TRANSPORT` model, so double click on the top row, i.e., `SeqNr '001'`, and Name `'TRANSPORT'`. The file `'transport.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; once you have set the tile layout it will remain fixed for that project.

*Practical CGE Modelling: Introduction to GAMS and GAMSIDE*

- The GAMS output file is returned automatically as a tabbed file in the editor window as `'transport.lst'`. (You can choose for this to not happen by changing the settings in the `File > Options` menu.)

## 6. Notes on 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) 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 out 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.

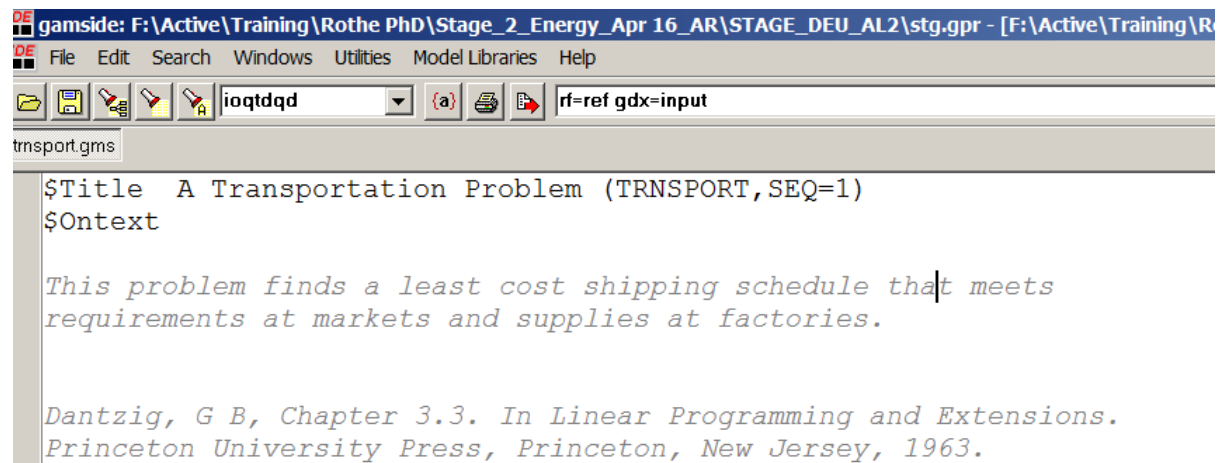
### Execution Errors

Execution errors are trickier. They can often arise because the model has been incorrectly specified. There are 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.
- Outputting all model information to GDX will later prove useful. This requires the use of the syntax `gdx=*` in the command line box. (See screen shot below.)
- The listing of values for variables from the model should be consistent with the values from the database.
- Generating a reference file will later prove useful. This requires the use of the syntax `rf=*` in the command line box. (See screen shot below.)
- 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.

*Practical CGE Modelling: Introduction to GAMS and GAMSIDE*



The screenshot shows the GAMSIDE application window. The title bar reads: "gamside: F:\Active\Training\Rothe PhD\Stage\_2\_Energy\_Apr 16\_AR\STAGE\_DEU\_AL2\stg.gpr - [F:\Active\Training\Rothe PhD\Stage\_2\_Energy\_Apr 16\_AR\STAGE\_DEU\_AL2\stg.gpr]". The menu bar includes: File, Edit, Search, Windows, Utilities, Model Libraries, Help. The toolbar contains icons for file operations and a dropdown menu showing "ioqtdqd". The main text area displays the following GAMS code:

```
$Title A Transportation Problem (TRANSPORT,SEQ=1)
$Ontext

This problem finds a least cost shipping schedule that meets
requirements at markets and supplies at factories.

Dantzig, G B, Chapter 3.3. In Linear Programming and Extensions.
Princeton University Press, Princeton, New Jersey, 1963.
```

## **7. 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 compiler of the library. GAMS uses this facility, in the form of the GAMS Model Libraries, to make available a wide range of GAMS models that illustrate the various capabilities of the GAMS language (the GAMS Model Libraries typically only download 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.

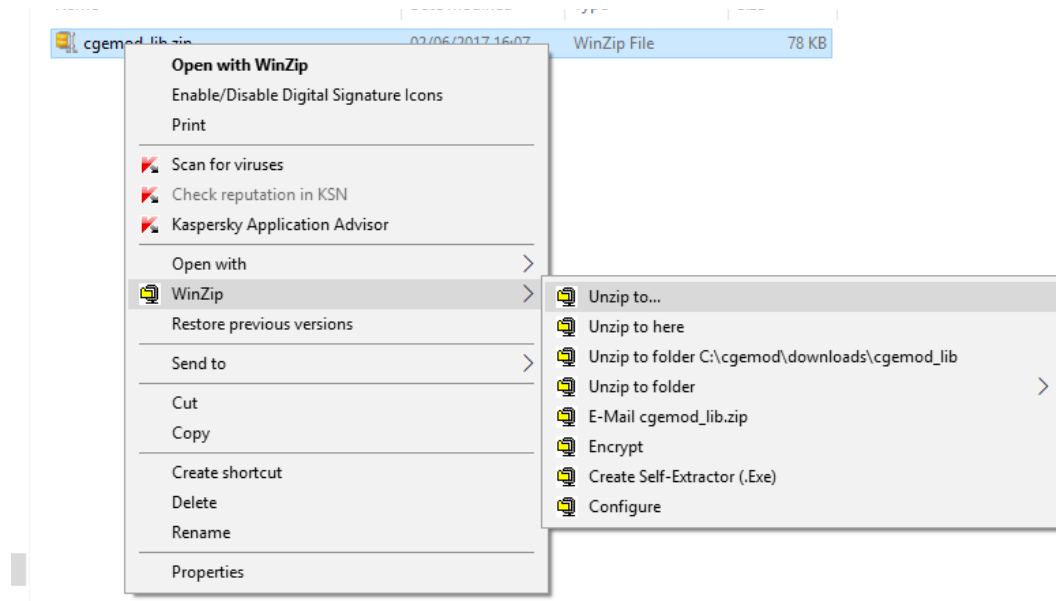
The programmes made available as part of this Introduction to GAMS/GAMSIDE are all supplied as WinZip archives. Create a directory called `cgemod_lib` in the directory `cgemod` that you created earlier. Each week, download the archive `**_lib.zip` from Moodle.

Since you will download a number of files during the course it is a good idea to have a single directory into which you first download files. So please create a directory in the `cgemod` directory called `downloads` (`C:\cgemod\downloads` – we will assume this has been done in future instructions)

The files from each archive must go into the directory `cgemod_lib`; each archive will contain many files. If asked if you want to overwrite files with the same name do so.

If you right click on the WinZip archive you have downloaded the menu illustrated below will become available. When unzipping the archive with WinZip you should use the ‘Unzip to ...’ option and this will unzip the archive into a directory in your `downloads` directory. Copy these files and paste them into your library directory `cgemod_lib`. The files in this directory must NOT be in sub directories.

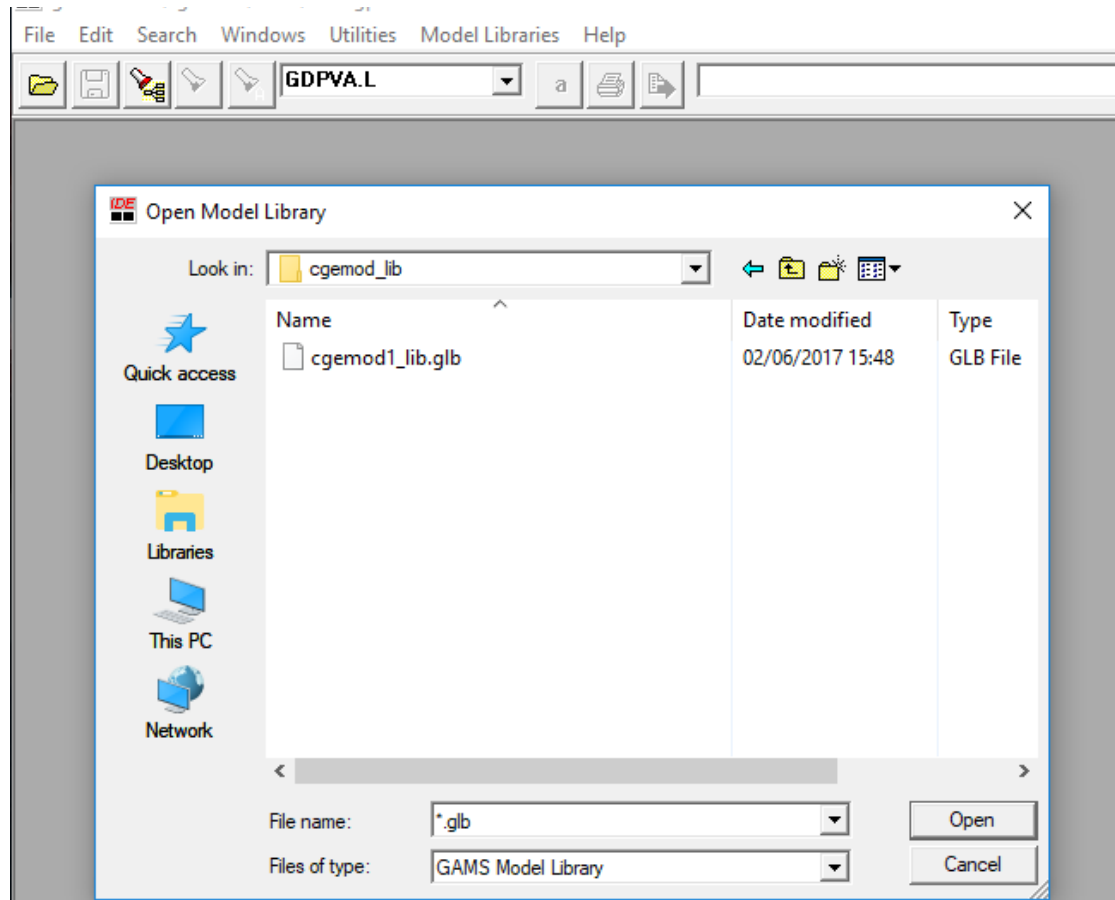
## Practical CGE Modelling: Introduction to GAMS and GAMSIDE



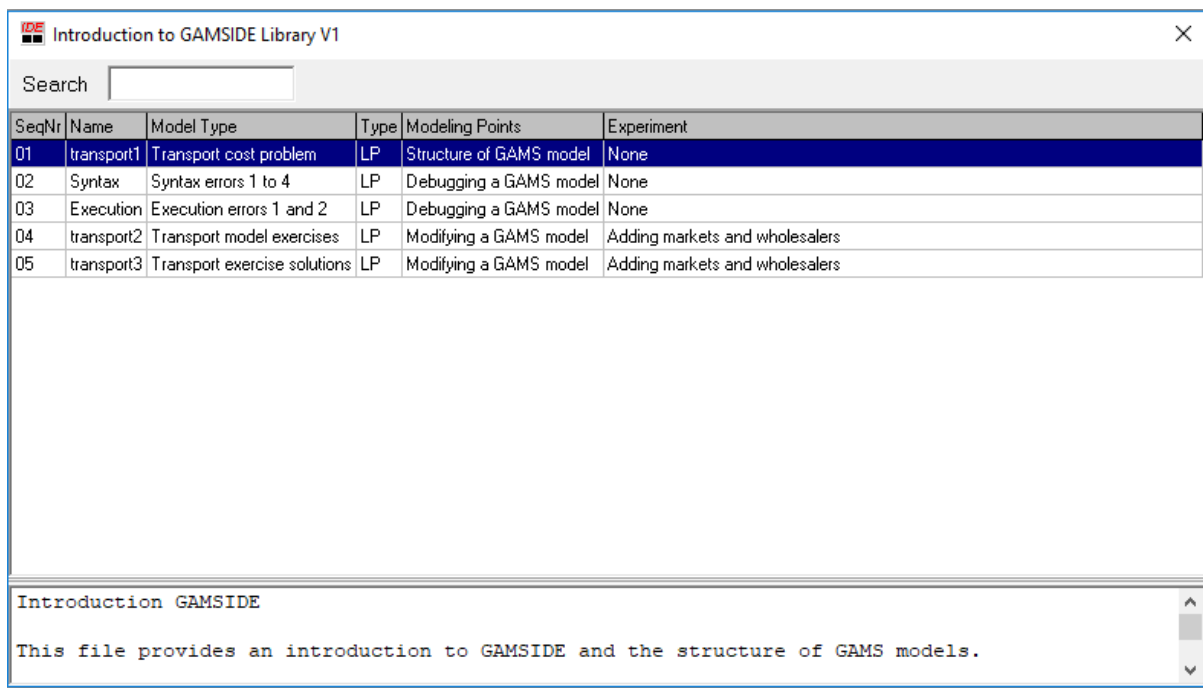
To open a User Model Library from GAMSIDE, choose File > User Model Library > Open User Model Library and then search for the directory into which the library was unzipped and open the `cgemod1_lib.glb` file. The Sequence number (SeqNr) details the order in which it is presumed the exercises will be conducted.



## Practical CGE Modelling: Introduction to GAMS and GAMSIDE



This would produce the following User Library manager.



Note how at the bottom of the library manager window there is a text box. The information in the box includes a brief description of the material associated with each item/group of items in the library and how this relates to other material in the library. Also note that that this text box can be scrolled and/or resized. (The library manager window opens with a fixed size for this text box so be careful to make sure you read all the information in the text box – some of it may be useful!)

### Some Tips

1. The material in the library is organised under the assumption that users will work through the material sequentially. If you choose to operate otherwise you may encounter difficulties because you are not familiar with a technique, or techniques, covered at a previous stage.<sup>10</sup>
2. The arrangement of the teaching materials in the library assumes that the user has progressed through the sequence in a specific order.
3. Use lots of directories. When teaching GAMS/CGE techniques it is our practice to require participants to create one directory for each group of exercises, e.g., `trans`, `clmod`, etc., and then within that directory to create a directory for each project/exercise.
4. Make **ONE** change at a time and save discrete steps in code development as unique files. Only (computer) geniuses – self defined – and fools assume that they will not make mistakes. Our standard advice to users is to assume that you **will** make mistakes and learn early how to (a) find them and (b) resolve them.
5. Avoid impatience: we all want to get to the interesting/'cutting edge' material/exercises as fast as possible, but relatively small amounts of extra time (1 or 2 hours) spent on exercises at each stage can in the long run save a lot of time.
6. A small batch file is included with each project/exercise that enables the user to clean out surplus files from the directories. The file is called `clean_*.txt`, where `*` is a name linked to the project/exercise. Before this file can be used to clean out surplus

---

<sup>10</sup> If you choose to operate in a different sequence and ask for help we will advise you to complete the previous exercises.

files it needs to be renamed as `clean_**.bat` – bat files can cause problems with computer security programmes even when they are included within WinZip archives.

7. At no time in the course should you explore the contents of the ‘`cgemod_lib`’ directory. You should only access the contents of the library directory from within GAMSIDE. We will never advise you to explore the content of the library directory or copy files directly from this directory.
8. If you try to use this library directory as a working directory you will have problems that we cannot solve. You will corrupt your previous work. If an ‘expert’ advises you otherwise, please ignore them. Even if they can manage the process, they will undermine your learning experience.