Course Materials from GAMS 2 Class
Using GAMSIDE

Bruce A. McCarl

Specialist in Applied Optimization
Professor of Agricultural Economics, Texas A&M
Principal, McCarl and Associates

mccarl@tamu.edu
mccarl@bihs.net
agrinet.tamu.edu/mccarl

409-693-5694
409-845-1706

Using GAMSIDE
What Is It?

GAMS -- Generalized Algebraic Modeling System

\+

IDE -- Integreted Development Environment

A Windows graphical interface to run GAMS



A Product of



Using GAMSIDE
How Does it Fit In?

GAMS is used in two phases.

First, one uses a text editor and creates a file which contains GAMS instructions.

Second, one submits that file to GAMS which executes those instructions causing calculations to be done, solvers to be used and a solution file of the execution results to be created.

Two ways to do this.

Traditional method – use a text editor set up the model then use DOS (or UNIX) command line instructions to find errors in and run the model.

The GAMSIDE alternative. It is a graphical interface to create, debug, edit and run GAMS files. Here we will cover using the IDE.

Using GAMSIDE – Getting Started

Summary of steps to using (steps expanded below)

1. Install GAMS and the IDE on your computer making an icon
2. Open the IDE through the icon 

3. Create a project by going to the file selection in the upper left corner.
4. Define a project name and location. Put it in a directory you want to use.
5. Create or open an existing file of GAMS instructions
6. Prepare the file so you think it is ready for execution
7. Run the file with GAMS by punching the run button
8. Open and navigate around the output

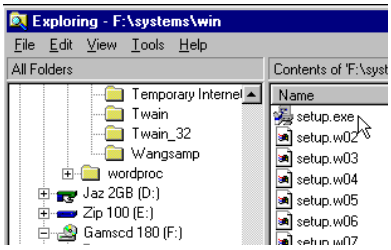
Using GAMSIDE
Installation

1. Install GAMS and IDE making an icon

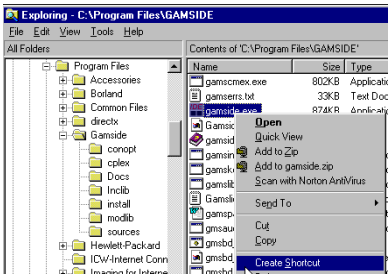
The GAMSIDE is automatically installed when GAMS is installed. To install do the following steps

a. Place the GAMS CD into your machine

b. Start the installation using the Windows Explorer. Do this by going into the systems subdirectory called win then double clicking on setup.exe



c. Make IDE icon (using explorer point at gamside.exe in c:\program files\gamside, then with right mouse click create shortcut and place that on desktop)



yielding

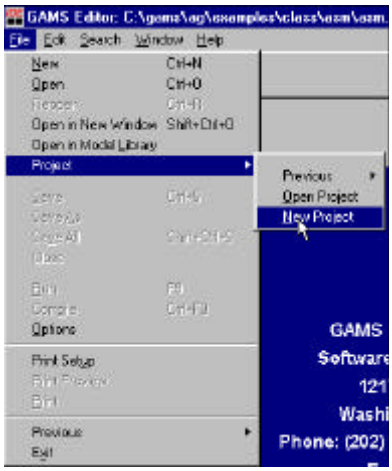


Using GAMSIDE
Getting Started after Installation

2. Open the IDE through the icon



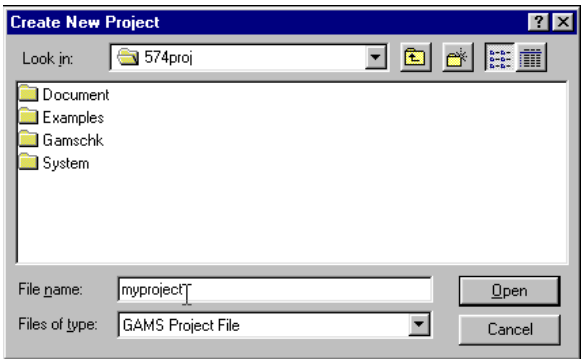
3. Create a project by going to the file selection in the upper left corner. Select to define a new project (Later you will use your previous projects).



What is a project? The GAMSIDE employs a "Project" file for two purposes. First, the project location determines where all saved files are placed (to place files elsewhere use the save as dialogue) and where GAMS looks for files when executing. Second the project saves file names and program options associated with the effort. We recommend that you define a new project every time you wish to change the file storage directory.

Using GAMSIDE
Getting Started

4. Define a project name and location. Put it in a directory you want to use. All files associated with this project will be saved in that directory.



In the "File name" area type in a name for the project file you wish to use. If I was doing this, I would go to a suitable subdirectory and create a subdirectory called useide and put in the name useide. In turn, your project name will be called useide.gpr where gpr stands for GAMS project.

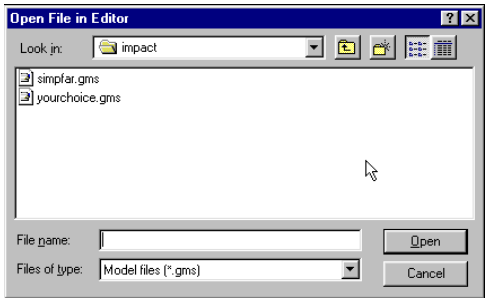
Using GAMSIDE
Getting Started

5. Create or open an existing file of GAMS instructions

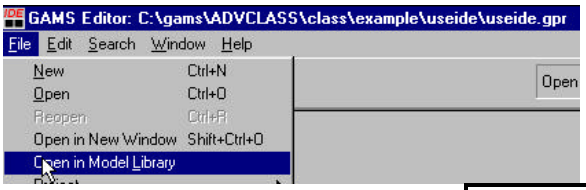
Several cases are possible

a. Create a new file (covered later)

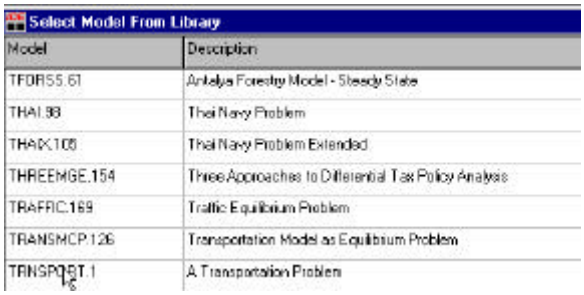
b. Open an existing file



c. Open a model library file (the simplest at this stage and the one we will use)



Select a model like transport 

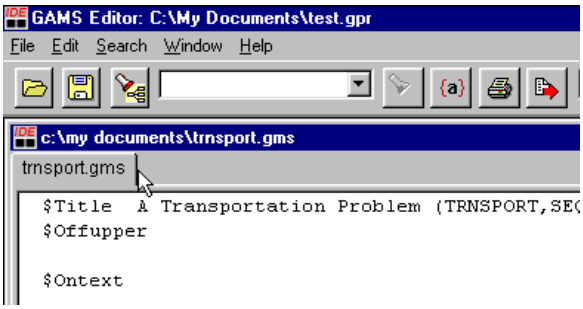


It will be automatically saved in your project file

Using GAMSIDE
Getting Started

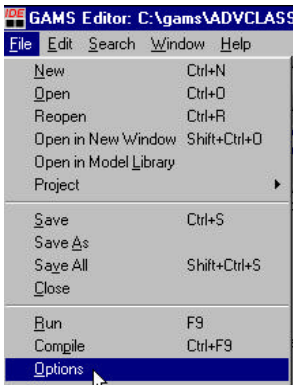
6. Prepare the file so you think it is ready for execution

When using model library transport.gms should now appear as part of your IDE screen

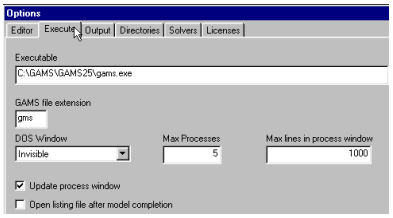


The IDE contains a full featured editor. Go through the file and change what you want.

Do a little housekeeping. Use the options dialog under file to set the output page length to 9999 and under the execute dialog check the box update process window

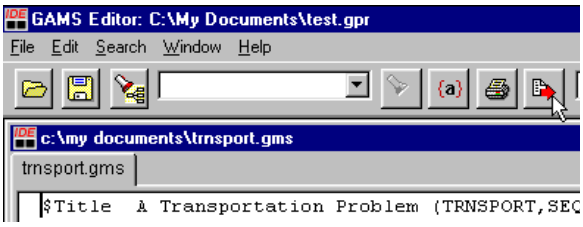




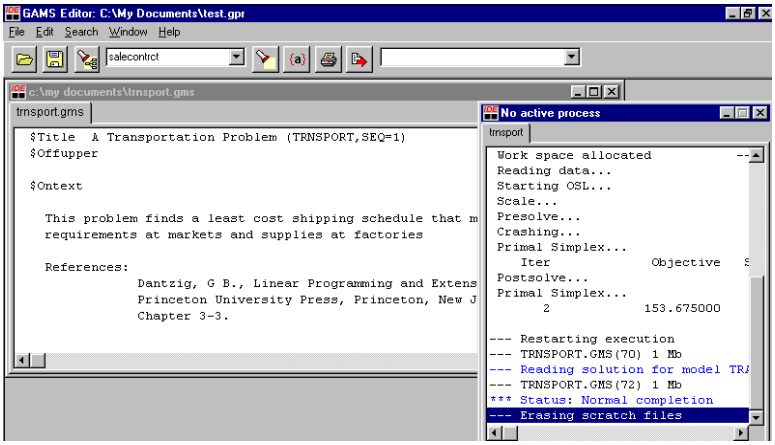


Using GAMSIDE
Getting Started

7. Run the file with GAMS by punching the run button

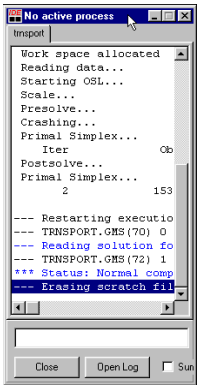


The so called process window will then appear which gives a log of the steps GAMS goes through in running the model and your model will run



Using GAMSIDE
Getting Started

8. Open and navigate around the output



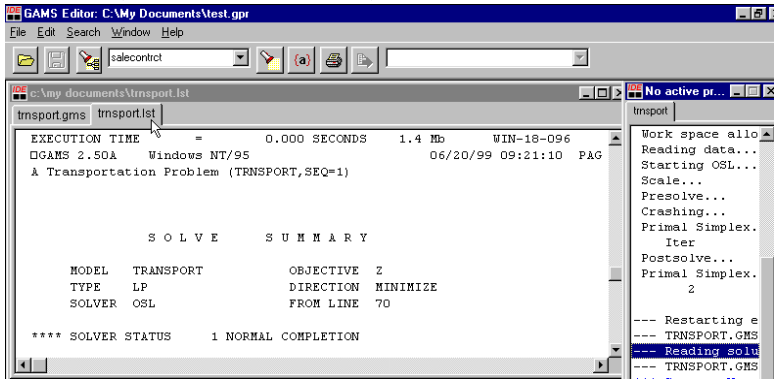
By double clicking on lines in the process window you can access program output both in general and at particular locations. The positioning of your access is determined by the color of the line you click on

Color of Line in Process Window	Function and Destination When Double Clicked
Blue line	Opens LST file and jumps to line in LST file corresponding to bolded line in Process file
Non bolded black line	Opens LST file and jumps to location of nearest Bolded Line
Red line	Identifies errors in source file. Cursor Jumps to Source (GMS) file location of error. Error description text in process window and in LST file which is not automatically opened.

Using GAMSIDE
Getting Started

8. Open and navigate around the output

After double-clicking on any non red line, our main editing window is augmented by the LST file (see the tabs)



We can navigate as we would with an editor or word processor, as we are automatically in the IDE text editor. The file is frequently partially obscured by the process window. Is yours? You might want to narrow the process window to the side as in the picture above.

Clicking at different points in the process window will cause you to be positioned at the line associated with the nearest blue line.

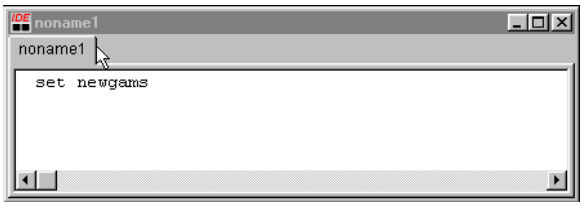
Using GAMSIDE
Working with your own file

Now you are ready to work with your own files. You may already have a file or you may need to create one.

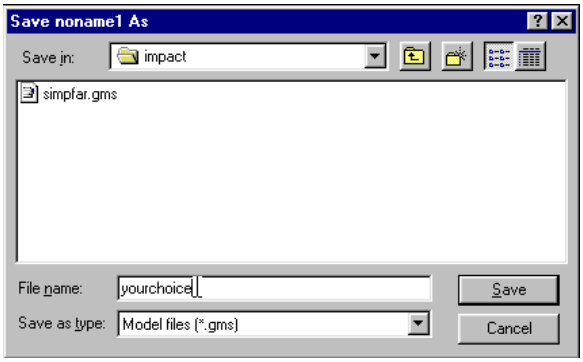
Creating a new file - Two principal ways

1. Open existing file and with save as dialogue from file menu change it's name. Now modify contents to what you want. You may cut and paste as in other Windows programs.

2. Open the file menu and use the new option. You will then get a file called noname which you may type GAMS instructions into



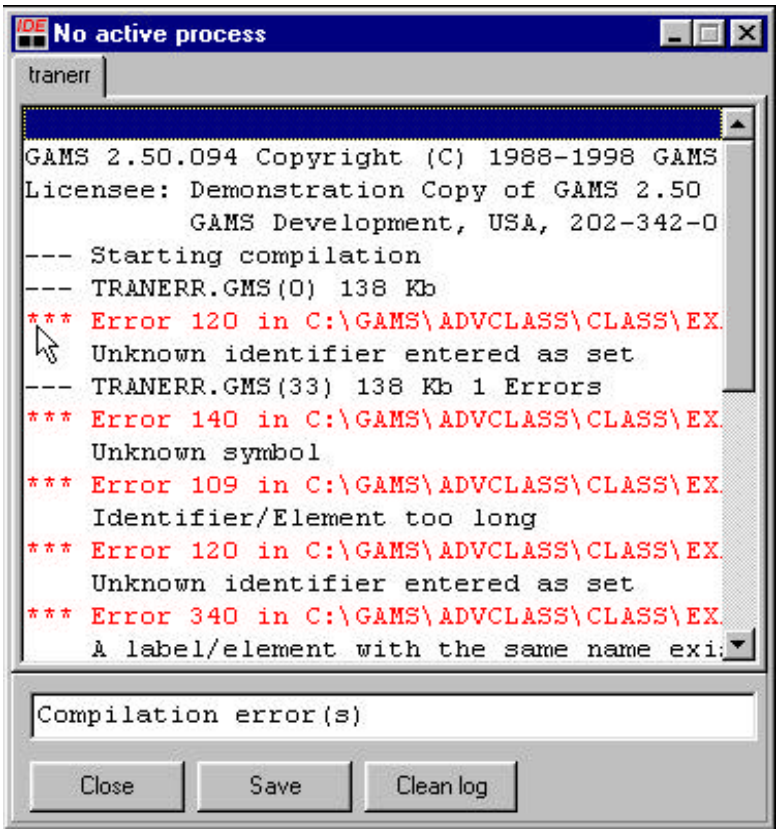
Save that file with whatever name you want. Note by default it will be assigned the extension .gms



Using GAMSIDE
Fixing Compilation Errors (tranerr.gms)

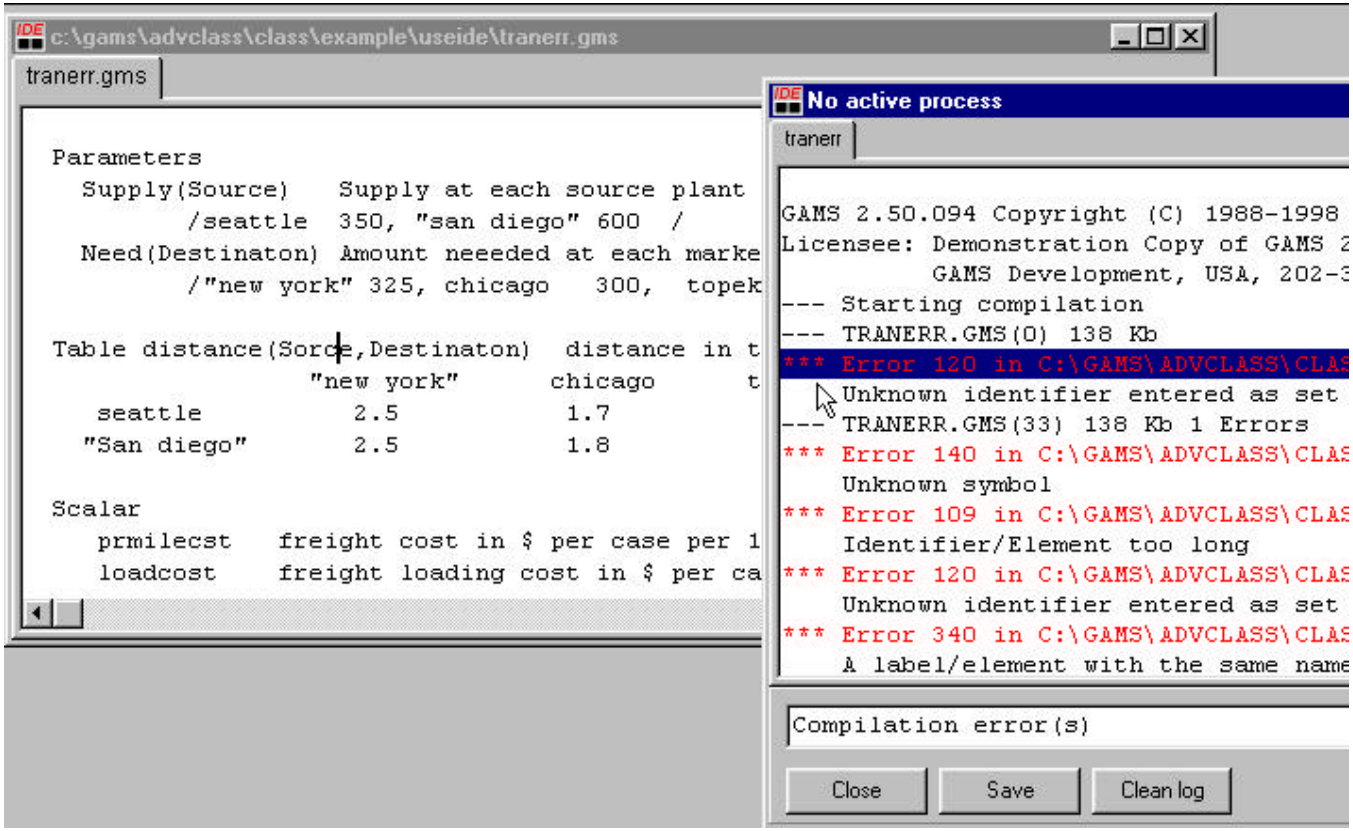
No one is perfect, errors occur in everyone's GAMS coding. The IDE can help you in finding and fixing those errors.

Let's use the example tranerr.gms to illustrate. A run of it yields the process window



The red lines mark errors. To see where the errors occurred double-click on the top one.

Using GAMSIDE – Fixing Compilation Errors (tranerr.gms)

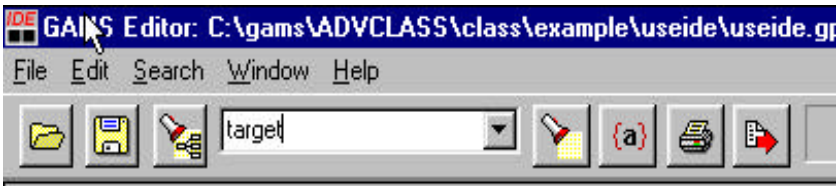


A double-click takes you to the place in the source where the error was made.
The tip here is always start at the top of the process file when doing this.

Using GAMSIDE
Ways to find and/or replace text strings.

For finding two dialogs can be used

The fundamental ones involve use of the flashlight and search windows



Type the text string target you are after in the widow

Hitting the  finds what you want in the current file

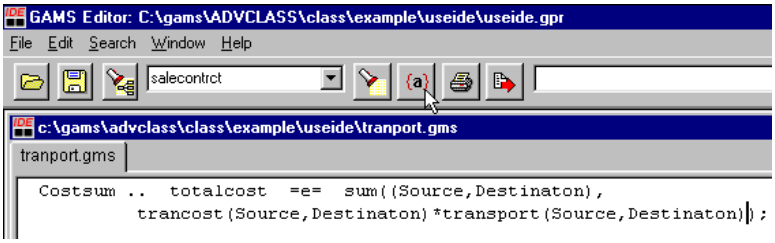
Hitting the  finds what you want in the directory where the project is located

You can also access search and replace through the search menu. That dialogue gives more options, but only searches or replaces within the current file

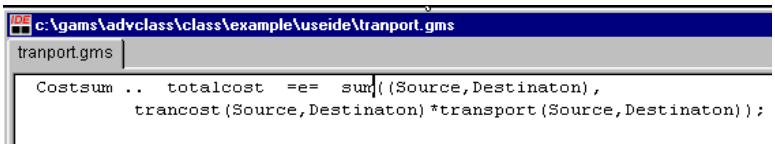
Using GAMSIDE
Matching Parentheses
Using file tranport.gms

The IDE provides you with a way of checking on how the parentheses match up in your GAMS code. This involves usage of the symbol  from the menu bar coupled with appropriate cursor positioning.

Suppose we have a line of GAMS code like



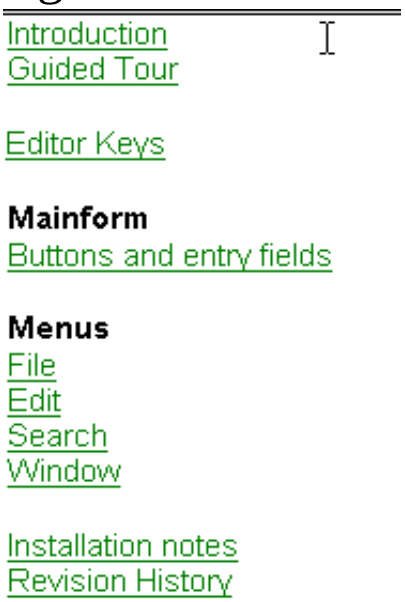
Positioning the cursor right before the ending parentheses and tapping the symbol will lead you to the beginning parentheses whether it be 1, 100, or 1000+ lines away and vice versa.



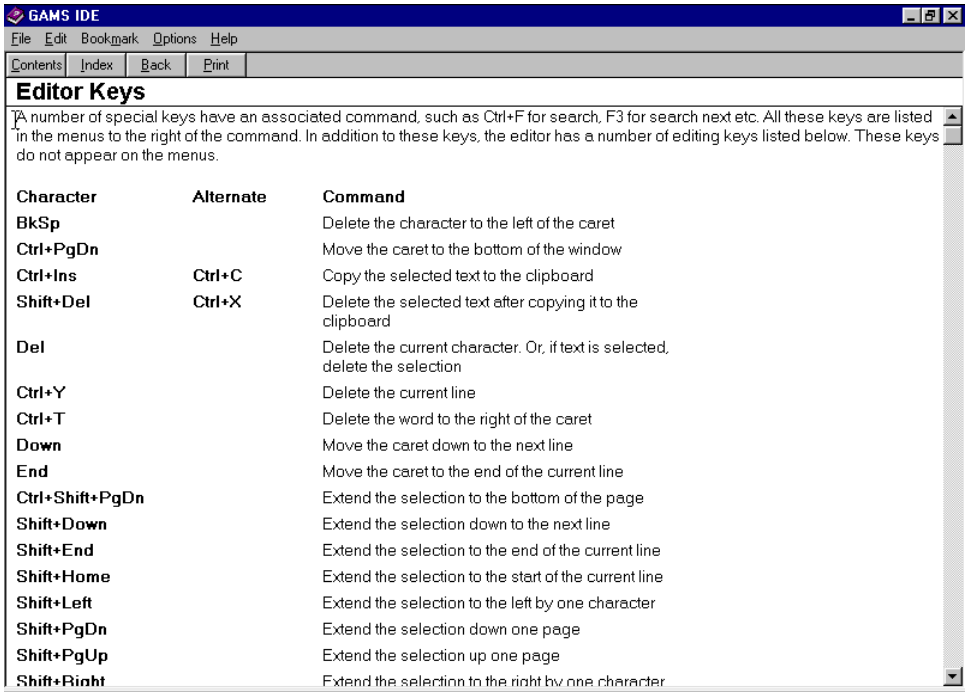
This feature will also matchup { } or [ ]

Using GAMSIDE -- Finding out more
Use the Help

Choose help and you get



Which contains such things as

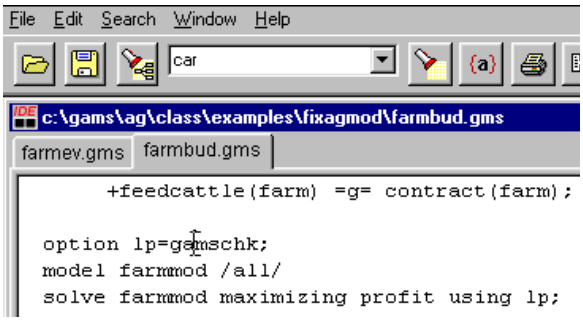


Read it throughly.

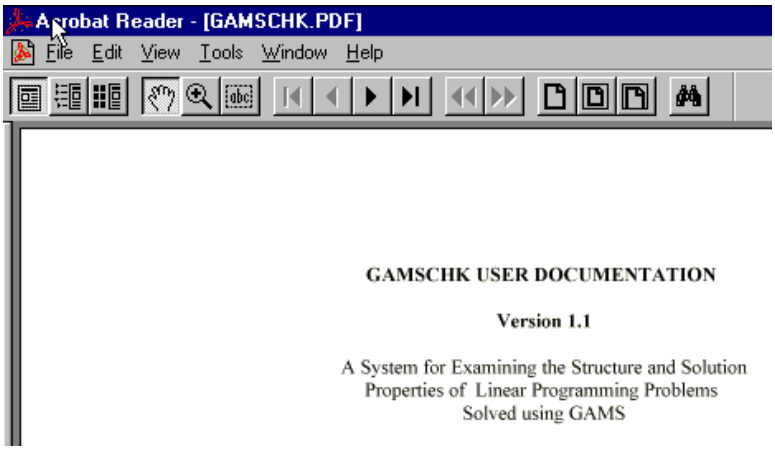
Using GAMSIDE
Accessing documentation on GAMS through the IDE.

The GAMSIDE has a tie in to documentation. In particular suppose we wish to know about a particular item and there happens to be a file on that item. For example suppose we are going to use GAMSCHK and our source code contains the line option lp=gamschk;

If we place the cursor over the word GAMSCHK and press the <F1> key as follows

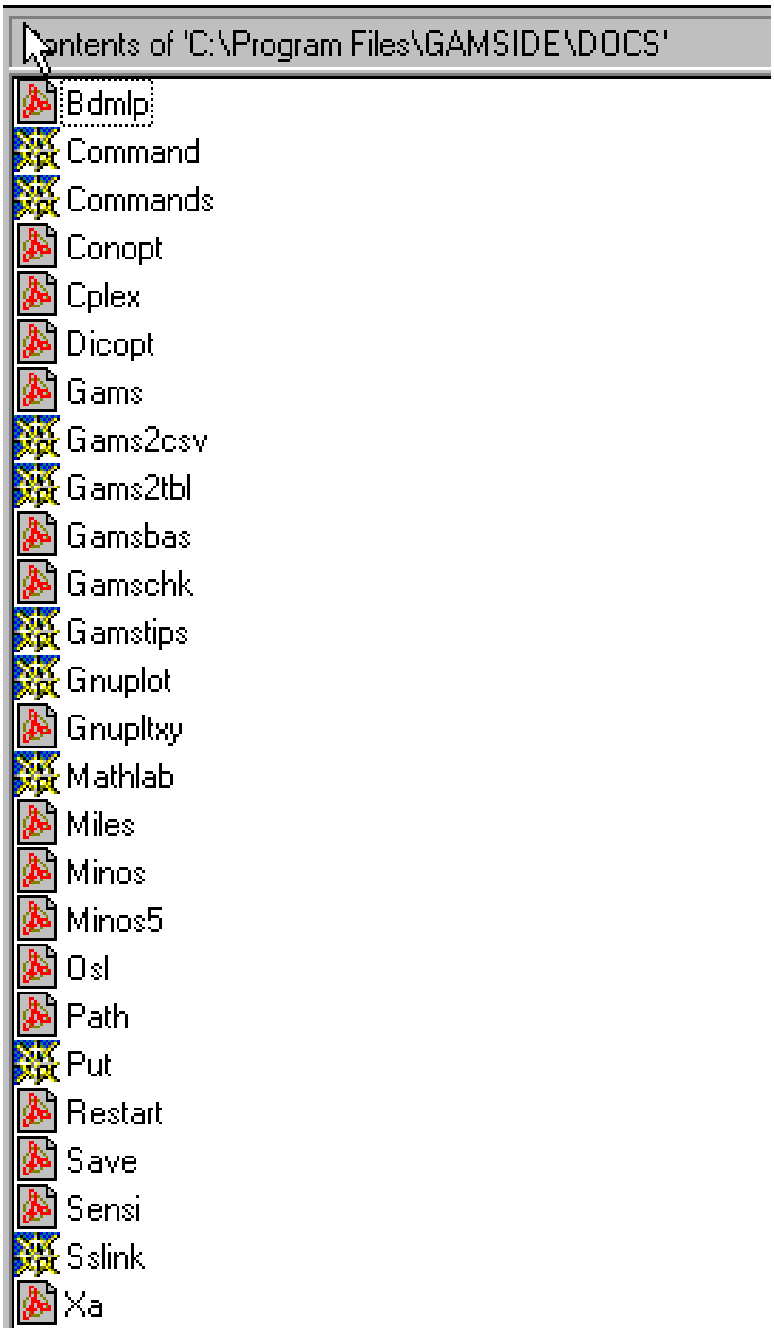


we get



Using GAMSIDE
Accessing documentation on GAMS through the IDE.

In fact using the docs directory associated with this file allows us to get any of the following



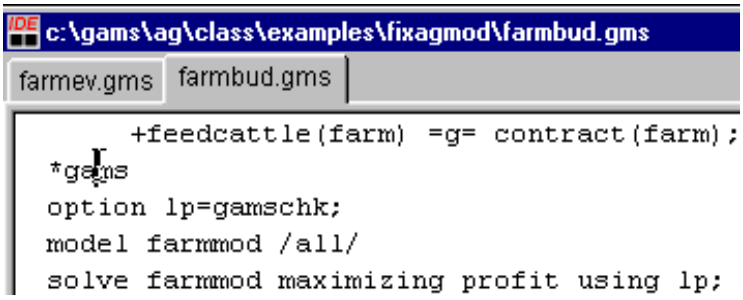
Using GAMSIDE
Accessing documentation on GAMS through the IDE.

The files used are those in the docs directory that were created for this document. You can add more. Any file with a pdf or html extension will work if you add it to c:\program files\gamside\docs.

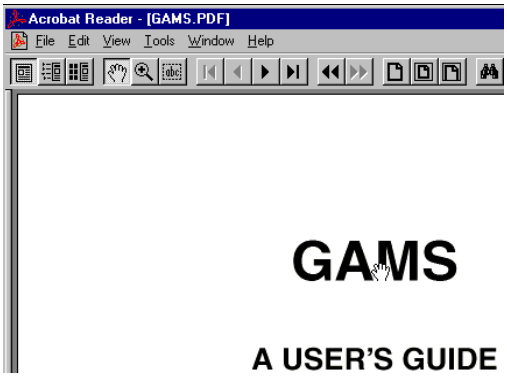
The documents looked up do not have to beones with the same name as a command that GAMS recognizes.

Try putting in the following

*gams



In turn you get the GAMS Users Guide

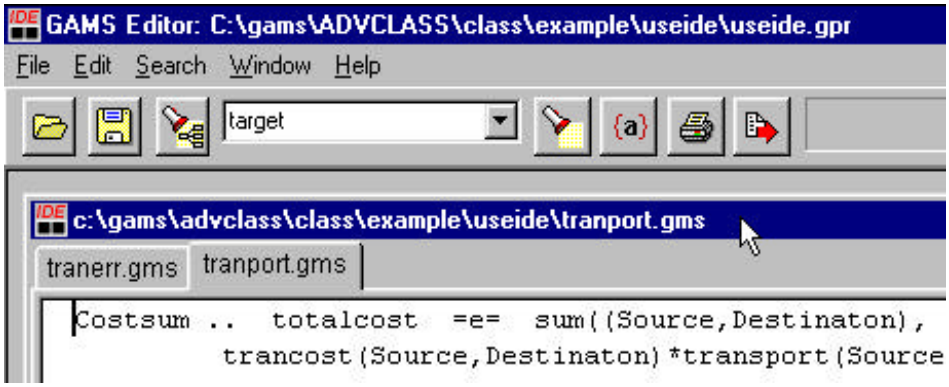


Using GAMSIDE
A difficulty you will have

When using and teaching the IDE, I find that IDE project location in interaction with file placement gives me and my students fits. I have a rule of thumb to avoid problems, but you will inadvertently not follow it.

Make sure that you are working on files located in the same directory location as the project is located.

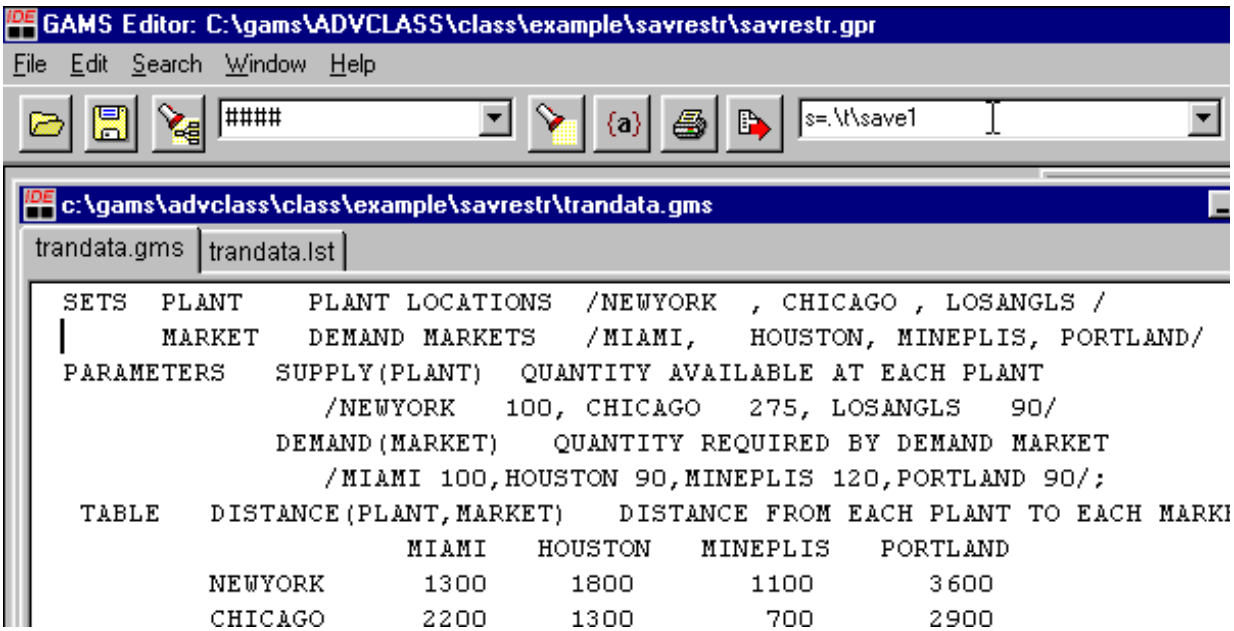
You do not have to follow this rule but deviations are the same as asking for trouble. When GAMS executes a file in a different directory it will not look for options files, GCK files, include files etc where you are pointing and logically expect them to be. GAMS will look in the directory where the project is located.



Another rule of thumb - Whenever you need to work in a new directory define a new project

Using GAMSIDE
Command Line Parameters

Experienced DOS or UNIX based GAMS users are used to having command line parameters associated with their GAMS execution commands. In the IDE a box is available just to the right of the execute button where we can associate a set of execution time parameters with a file and the IDE will remember these whenever the file is opened in this project.



The IDE saves this file specific parameter information in the project file. This is particularly useful for save and restart parameters as once they are definedthey are associated with every subsequent use of the file provided you're using the right project and have not changed the restart information.

Using GAMSIDE
When is it Not Worth Using?

There are costs and benefits of these approaches.

The IDE is much easier for simple models but is currently limited to PCs.

The DOS/command line approach is generally better for models in customized environments.

A development strategy for more complex implementations

A. Use the IDE to get it right

B. Debug components of large models using save and restart

C. Then use DOS/UNIX with batch files such as

```
gams mymodel -codex 1 -lo 0 -s ./t/save1
call myprogram.exe
gams moremod  -lo 0 -r ./t/save1
```