PFE-GAMS : GAMS under Windows

Philip A. Viton
Department of City and Regional Planning
The Ohio State University
190 W. 17th Avenue
Columbus OH 43210

November 8, 1999

Contents

1 Introduction **2**

2 Using the PFE-GAMS system **2**
2.1 An example 3
2.2 Large files — the gamsrd utility 3

3 Upgrading PFE-GAMS **5**

4 Getting and Setting Up PFE **6**
4.1 Obtaining PFE 6
4.2 Setting up PFE 7
4.3 Customizing PFE 7

5 Setting up PFE-GAMS **7**
5.1 Creating the GAMS configuration 8
5.2 PFE version 00.07.002 or later 9
5.3 Setting up the GAMS configuration 10
5.4 Last steps 12

6 Extensions **13**
6.1 Command-Line Parameters 13
6.2 Keyboard shortcuts 14
6.3 Read-Only Files 15
6.4 Autoloading results 15

7 Debugging **16**

1 Introduction

GAMS under MS-Windows is normally run from the command line in a DOS session; you would then use an DOS text editor and/or viewer to work with your output. This note describes an integrated environment, which I call `PFE-GAMS`, for running GAMS programs directly under Windows. With this environment you will be able to edit your GAMS source file in one window, view the results of GAMS runs in another, and switch between them just as you would with any multi-window editor. Changed files will be automatically updated.

This system is based on Alan Phillips' `PFE` text editor, which is available as free-ware — that is, it is absolutely free, and no licensing fee is required — from many Internet sites: see section 4.1. The next section describes how you would use `PFE-GAMS` once it is set up; after that we explain how to obtain `PFE` and set up `PFE-GAMS`. The setup explanation is geared to Windows95/NT, and assumes that you are using the 32-bit version of `PFE`. There is also a 16-bit version of `PFE` which runs under Windows 3.1/3.11 and for which there is one important difference: those systems do not use the Registry to store configuration information, but rather rely on configuration-specific `.ini` files. So instead of creating a new Registry key, you will create a new `.ini` file. See the `PFE` documentation on how to do this.[1]

2 Using the `PFE-GAMS` system

We assume that you have followed the instructions in the next sections, and now have an icon for `PFE-GAMS` which calls the `PFE` editor with the required GAMS settings. The way you use this system varies as between the first GAMS run and all others.

- For the *first* run, you load the source (`.gms`) file into `PFE`, make any modifications, and run it through GAMS. This creates one or more results files on disk. You then load into the editor any of these you wish to examine.
- For all *subsequent* runs, `PFE` will automatically notice that the results files have been changed, and ask you for permission to update them. Thus you can always view the latest results. (See section 6.4 for a way to automate this, too).

[1] Note that in this situation, the simplified way of setting up `PFE-GAMS` described in section 5.2 isn't available to you, even with the latest release of `PFE` since there appears to be no way to create an `.ini` file from a Registry Configuration file; I use only the 32-bit version, so I've never created an `.ini` file. If any user of the 16-bit system would like to send me an `.ini` file, I'd be glad to include it here.

2.1 An example

We can test the system using the GAMS file `swe9` located in `c:\gams\users`.

- Click on the `PFE-GAMS` icon to start `PFE` in "GAMS mode".
- Using File Open, load the file `swe9.gms` from your `gams\users` area. You may want to size it to fit in the upper half of the `PFE` window.
- Click Execute then either GAMS (to run GAMS), GAMS Compile (to just compile the source file, without running), or GAMS+Gamsrd (to run, and produce a summary file using the special utility `gamsrd` described in section 2.2). You will see GAMS executing in a DOS window.

This is the end of the *first* run. GAMS creates the files `swe9.lst` (if you chose GAMS, or GAMS Compile) or `swe9.lst` and `swe9.sum` (if you chose GAMS+Gamsrd).

- Load either or both of these files into the editor, and examine them. You may want to size them so that (say) both the `.gms` and the `.lst` files are both visible in the editing window.
- Now do a second run. With `swe9.gms` as the active window, click Execute then either GAMS or Gams Compile, or GAMS+Gamsrd. GAMS executes in a DOS window as before.
- Now click on one of the other document windows (`swe9.lst` or `swe9.sum`). `PFE` "knows" that the disk versions of these files have changed as a result of your second GAMS run, and offers to update the displayed versions. Click OK to allow this, and you will see the new results.

That's all there is to it. You can start a second run from the `.lst` or `.sum` windows, because the Execute menu commands pass through everything about the file *except* its extension, which is supplied by a batch file.

If something goes wrong, then first, make sure that all the items on the Execute menu refer to the right files; see section 5.3.1. For some additional debugging hints, see section 7 below.

2.2 Large files — the `gamsrd` utility

GAMS can produce extremely large (multi-megabyte, under some circumstances) `.lst` files. While there are ways of reducing this, and while `PFE` can handle large

files up to the memory limits of your system, it can be time-consuming to load a large file, or to page through it in search of the information you want. This section describes one solution to the problem, a way of producing a small file containing only the optimization results you need to check.[2] While you can do the same thing using GAMS' `put` report-writing facility, the approach here is simpler — if less well formatted — requiring only the `display` statement.

To implement this solution, you simply include a series of `display` statements at the end of the GAMS source file. This writes the desired results to the end of the `.lst` file. After running GAMS and producing the `.lst` file, you run the program `gamsrd`, which extracts the information to a separate file.

For `gamsrd` to work, it needs to know where your summary information starts in the `.lst` file. To show this, you must include in your GAMS source file, a special `display` statement containing a keyword. This keyword must be unique to your program: it cannot be the name of any other variable or value, either supplied by you or produced by GAMS.[3] Place this statement immediately before the `display` statements summarizing your results, like this:

```
display "ZQQ Summary of Results";
```

where "ZQQ" is the keyword. The keyword statement can contain other information as well, but the keyword should normally be first. This statement should then be followed by the series of `display` statements which actually produce your summary output; and these should be the last part of your source code. Important: the keyword is case-sensitive: it must match the way the line is displayed in the GAMS `.lst` file.

Given this, `gamsrd` looks for the keyword, and then writes everything that comes after it to a file you name. So for example, to write the summary information contained in `myfile.lst` (which is produced by GAMS) to `myfile.sum` you would say

```
gamsrd  myfile.lst  myfile.sum  ZQQ
```

or in other words

```
gamsrd [input file] [output file] [keyword]
```

[2] Note that you cannot put into this file GAMS output associated with the model setup, such as equation listings, etc. For these, you must view the `.lst` file directly.

[3] So for example, the word "Summary" will not work, because GAMS uses it when it writes the results of your model solution. If you get unexpected results in the file produced by `gamsrd`, this is the first thing to check.

The batch file `pfegamsr` provides support in `PFE-GAMS` for this utility; the file `c\gams\users\swe9.gms` illustrates the source-file setup.

One problem with this is that your summary file gives no indication of when GAMS was run, and this information could be important. You cannot use the `display` statement to write this information directly, so you need to be indirect. One solution is to have GAMS create a file with the date (and time) of the run, which you will then append to the summary file created by `GAMSRD`. Here's some GAMS code to do this: it creates a file `datime.txt` in the same directory as your `.lst` file.

```
file datime / datime.txt/;
put datime /// "GAMS Run:  ",system.date,"   time:  ",system.time;
putclose datime;
```

Then add the following lines to `pfegamsr.bat`, to insert the contents of `datime.txt` at the end of the `.sum` file, and then delete it:

```
if exist %1datime.txt copy %1%2.sum+datime.txt %1%2.sum >nul
if exist %1datime.txt del %1datime.txt >nul
```

(The `>nul` just prevents any messages associated with the `copy` from appearing on the screen). It's important to delete the file, in case some other GAMS source file does *not* create one: if it had not been deleted, the (wrong) information would be appended to the other results.

3 Upgrading `PFE-GAMS`

It is occasionally necessary to upgrade the 32-bit version of `PFE-GAMS`, to fix bugs. Here's how (assuming that you've installed it into `c:\gams\users`):

1. Retrieve the updated `pfegams.zip` and unzip it to `c:\gams\users`
2. Start `PFE` in *non*-GAMS mode (this is important: you cannot override the current configuration)
3. Click Options Preferences then Configuration sets. The box "Current configuration" should *not* say `GAMS`.
4. Under Existing configuration sets select `GAMS`
5. Click Import to ...

6. In the resulting dialog box, move to `c:\gams\users` and select the file `pfegams.rcf`.

7. Click Open. `PFE` will tell you that the registry configuration set has been imported. Click OK. You can now exit `PFE` and then re-start it in GAMS mode, by clicking on the `PFE-GAMS` icon.

4 Getting and Setting Up PFE

This section describes how to obtain and set up `PFE` in "non-GAMS" mode. In the next section, we set up the GAMS-specific features.

4.1 Obtaining PFE

The first thing is to obtain the correct version of `PFE`. There are separate builds for Windows 3.1/3.11 and for Windows 95/NT, and it is important to obtain the one matched to your operating system, because though the editor will run under an operating system mis-match, the DOS Command facility — which is at the heart of the `PFE-GAMS` system — will not. If you set up `PFE` and then find that the first menu item under Execute is grayed out, then you are probably using the wrong version. The site

`http://www.lancs.ac.uk/people/cpaap/pfe/pfefiles.htm`

is the primary source for the files, though they are widely mirrored.

I strongly recommend that if you already use `PFE`, you upgrade to the latest version. If you are using Windows 3.1/3.11 then you *must* use `PFE` version 00.07.002 or later: although the configuration-specific `.ini` file option was present on earlier versions, it didn't work correctly. Version 00.07.002 fixed it.

Important note, November 1999. Alan Phillips has announced that development of PFE has ceased. I very much regret this — PFE is a wonderful editor — and I'm leaving the link to Lancaster in place in the hopes that he'll reconsider. Meanwhile, the final versions of PFE are available at

`http://www.winsite.com`

— look there for `pfe101i.zip` (Win 95/95/NT/2000) or `pfe101.zip` (Win3.1x).

4.2 Setting up PFE

Create a directory on your hard disk for PFE, calling it (say) c:\pfe. Change to that directory, and unzip PFE's .zip file into that area. If you're updating your PFE system, just over-write the older files (updating preserves Registry information in the 32-bit version).

Now create a shortcut for the basic PFE program. Choose a folder or your Desktop, then right-click. Select New then Shortcut. Reply to the dialog as follows:

Command Line: c:\PFE\PFE32.exe (or browse to the PFE folder and select the executable). Remember to select the 32-bit version under Windows95/NT even if both it and the 16-bit versions are present.

Name: choose something like PFE; *don't* refer to GAMS yet.

Click Finish and the PFE icon will appear.

4.3 Customizing PFE

Start PFE by double-clicking on its icon. Then click Options then Preferences. You see a number of items displayed in a list on the left of the window. You can now make any changes that you might want to have available in *all* configurations of PFE. You can look through the list and see what you want to change: one thing is to create backup files with the extension .bak: the default is to use the extension .$$$, which I for one do not like. To change this:

- Click Backup.
- Enter bak into the box to the right of the phrase "Into file with extension ..."
- Do *not* click "Add extension to existing name" even if it is not grayed out.

when you have made any changes you want, click OK. This updates PFE's default configuration set.

5 Setting up PFE-GAMS

In this section we describe how to set up the features which make GAMS usable from within PFE. Begin by creating an area (folder) for some files associated

with the PFE-GAMS system. In what follows I assume that this directory is c:\gams\users. Copy the file pfegams.zip to this folder, unzip them; you may now delete the .zip file. You will have the following files:

- pfegams.ico (optional icon for the PFE-GAMS system)
- pfegams.bat (batch file for running GAMS program from within PFE)
- pfegamsr.bat (batch file for running GAMS followed by gamsrd)
- gamsrd.exe (the gamsrd utility described in section 2.2)
- wait.exe (utility program to have DOS windows persist on screen)
- swe9.gms (test file)
- pfegams.rcf (registry configuration file for PFE version 000.07.02 or later)
- gamsrd.c (source file for gamsrd)

The two batch files are the crucial ones. As distributed, they assume that your GAMS executables reside in c:\gams\nt and subdirectories. If you have installed GAMS to a different location, you must edit both batch files and fill in the correct locations.

5.1 Creating the GAMS configuration

We now create a special PFE setup for running GAMS programs.

Start PFE, and click Options then Preferences and click on the item Configuration Sets. This opens a dialog box which allows you to create special configurations in the system Registry. To the right you see three boxes, for Default Startup Configuration Set, Current Active Configuration Set, and Configuration Sets. All currently say the same thing, namely, DefaultConfiguration.

- Click on DefaultConfiguration in the Existing Configuration Sets box.
- Click Copy.
- In the new dialog box, type GAMS, and click OK. You are told that this configuration does not exist and asked if you want to create it; reply Yes, then click OK when you are told that the new set has been created.
- Back in the main dialog, click OK to finish the dialog.

Quit PFE. We now create a separate shortcut for the PFE-GAMS system.

- Select a folder (or the Desktop) for the PFE-GAMS shortcut. If you selected a folder, open it; if you chose the Desktop click on an empty spot. Right-click. Select New Shortcut.

- In the Command Line dialog box, type `c:\PFE\PFE32.exe /r GAMS` or browse to the location of the executable, and then add the `/r GAMS` by hand: this tells `PFE` to start with the `GAMS`-specific configuration.
- For the shortcut name, use a name which reminds you that this is `GAMS`-specific, say `PFE-GAMS`
- Click Finish.

At this point you can customize the shortcut. We supply a `PFE-GAMS` icon (`c:\gams\users\pfegams.ico`), which you can substitute for the default `PFE` icon. In addition, you may want to have `PFE-GAMS` start in a more convenient folder, one closer to your GAMS source files. Either of these changes can be made by right-clicking on the `PFE-GAMS` icon, selecting Properties and making the necessary changes.

If you are using `PFE` version 00.07.002 or later go to section 5.2. Otherwise go to section 5.3

5.2 `PFE` version 00.07.002 or later

Version 00.07.002 of the `PFE` editor was released in January 1998. The 32-bit version contains a quick way of setting up the `PFE-GAMS` system, the ability to write and read a configuration file. The distribution of the `PFE-GAMS` system includes a file `pfegams.rcf` which incorporates all the facilities discussed below, so you don't have to set each item by hand. Here's how to import the configuration:

- Start `PFE` using the *non-GAMS* `PFE` icon.
- Click Options then Preferences then Configuration Sets.
- Under Existing Configuration Sets choose the `GAMS` set.
- Click Import To. In the resulting dialog, browse to `c:\gams\users` and select the file `pfegams.rcf`. Click Open. `PFE` will import all the configuration information. Click OK.

If you have placed the `PFE-GAMS` batch files in a directory other than `c:\gams\users` you will need to change their locations on the Execute menu. To do this:

- Click Options then Preferences then Execute Menu. You will see a list of three items.

- Select each one in turn, click Edit, and under Command change the beginning of the line to point to wherever you installed the batch files.

That's all there is to it (except that you may want to read the material in section 6); you can now test the PFE-GAMS system, as described in section 2.1.

5.3 Setting up the GAMS configuration

This section contains detailed steps for creating the GAMS-specific configuration for PFE. Remember, if you are using PFE version 00.07.002 or later, you can avoid all this work by using the configuration-import technique described in section 5.2 Otherwise, double-click on the PFE-GAMS shortcut you created in section 5.1. PFE will start. Click Options then Preferences, then click on the item Configuration Sets. You should see that the Current Active Configuration Set is GAMS. You're now ready to modify this for GAMS-specific settings.

5.3.1 Add the Execute Menu items

We first add support for running GAMS from within PFE. Click Execute Menu.

- Click New
- In the resulting dialog box, fill in the items as follows:
  - Menu Item: GAMS
  - Command: `c:\gams\users\pfegams.bat %u\%d\  %n`
    Note the backslash between the %u and the %d and after the %d: it is very important that this text be entered *exactly* as written.
  - Directory: (leave blank)
  - Under Start Command click " ...as independent process"
  - Under Options click "First check for unsaved files"
  - Click OK

We now create a second GAMS command, which allows you to compile without executing your file.

- Click New
- In the resulting dialog box, fill in the items as follows:

- Menu Item: `GAMS Compile`
- Command: `c:\gams\users\pfegams.bat %u\%d\  %n  a=c`
  Note the backslash between the `%u` and the `%d` and after the `%d`: it is very important that this text be entered *exactly* as written. Note also that the only difference between this entry and the one for `GAMS` is the `a=c` at the end, so you may want to cut-and-paste.
- Directory: (leave blank)
- Under Start Command click " ...as independent process"
- Under Options click "First check for unsaved files"
- Click OK

The last step is optional, and should be added only if you want to work with the `GAMSRD` utility for creating short summary files from GAMS' `.lst` file, as described in section 2.2. In that case:

- Click New
- In the resulting dialog box fill in the items as follows:
  - Menu Item: `GAMS+Gamsrd`
  - Command: `c:\gams\users\pfegamsr.bat %u\%d\  %n`
    Note the backslash between the `%u` and the `%d` and after the `%d`: it is very important that this text be entered *exactly* as written. Note also that the only difference between this entry and the one for `GAMS` is the name of the batch file (`pfegamsr` rather than `pfegams`) so you may want to cut-and-paste.
  - Directory: (leave blank)
  - Under Start Command click " ...as independent process"
  - Under Options click "First check for unsaved files"
  - Click OK

5.3.2 Configure the File Filters for GAMS files

Next, click on File Filters. We'll add a file filter so that, when you click on the Open dialog, you see only files with extensions relevant to GAMS.

- In the box File Filter type `*.gms,*.lst,*.sum`
- In the box Description type `GAMS files`
- Click Add at Top

- In the box File Filter type `*.*`
- In the box Description type `All Files`
- Click Add at Bottom

5.3.3 Other configuration items

Under General Options, make sure that the box "Check disk files for changes by other applications" is checked. Make any other changes you'd like in the list.

Under MRU List you may want to change the number of files shown in the list from 5 to 9; it will make your work easier if you check the box "Change directory on opening file".

Under Screen Font, you may want to change the default font used for your editing: I prefer Courier New, Bold, 9 point; but it's up to you.

Click OK. This saves the configuration in the Registry.

Lastly, set up `PFE`'s print facility.

- From the main `PFE` window, click File then PageSetup.
- Work through the dialog items to tailor the page appearance to your preferences. One nice feature of `PFE` is that the screen and printer fonts can be different; I use Courier New, 8 point for printing.

Note that there is an interaction between the printer font point size and the `PW` entry in the `pfegams` batch file. `PW` controls the width of the lines written by GAMS: you should adjust the `PW` value so that at your chosen printer font point-size the entire line fits on a page.

5.4 Last steps

If you created the `GAMS+Gamsrd` command, the batch file `pfegamsr` will first run GAMS on your source file, and then run the `gamsrd` program to extract a summary file. The batch file is set up to produce a summary file with extension `.sum`, and to expect that the required "marker"(see section 2.2) is `ZQQ`. If you wish to change either of these, edit `pfegamsr.bat`: note that if you change the extension you should also change the file filters as described in section 5.3.2.

If you would like the DOS window created by the batch files to remain visible for a few extra seconds after GAMS has finished, you can edit the batch files

pfegams.bat or pfegamsr.bat in your \gams\users directory, and add the command c:\gams\users\wait n as the last line, where n is the number of seconds to wait.

You can now test the system, as described in section 2.1. However, the next section describes some additional features you may want to consider adding to your setup.

6 Extensions

6.1 Command-Line Parameters

If you need to include additional command-line parameters (as described in Appendix B of the GAMS manual) in all your GAMS work you can do so in two ways:

- You can add up to 3 parameters (2 for the GAMS+Compile command) to the commands as described in section 5.3.1: the setup for the Gams Compile command, where the parameter a=c is passed to the batch files, illustrates the technique.
- Alternatively (or if you need more than two or three additional parameters) you can modify the batch files and add them: the way the batch file handles the parameters PW and PS illustrates. This is a more efficient technique, especially if the parameters involve either the source file name or the path to it, since in all the batch files of PFE-GAMS the full path (including the drive letter) and a trailing backslash is available as %1, and the name of the source file (without a trailing . and extension) is available as %2.

  For example, to construct in a batch file the full name of a file which differs from your GAMS source file c:\mydir\myfile.gms only in having an extension .gmq you would write %1%2.gmq : in this expression %1 is c:\mydir\ and %2 is myfile (without period or extension). This is the way the name of the output file (with extension .lst) and the input-file directory (the idir parameter) are constructed in the batch files.

PFE has a sophisticated set of ways to pass directory and file information to external programs: look under sub[stituting current filename] in PFE's Help Search for Help On feature for more information.

6.2 Keyboard shortcuts

With the setup described thus far, you run or compile a GAMS program by making its window active and then using the mouse to click on Execute and then the menu item you want. I find that it's a lot easier to avoid the mouse, and start GAMS using a keyboard shortcut. Here's how to set this up. First, we'll have `Alt+R` ("R" for "run") do a GAMS run — ie act as a shortcut for item `1` on the Execute menu:

- Click Options then KeyMapping
- In the dialog, under Key/Key Sequence, check `ALT` and then use the drop-down dialog to find `R`.
- Under Functions, click the down-arrow and scroll through to find the item `ExecUserCommand1`
- Click Add

Next, we'll set up `ALT+C` ("C" for "compile") do do a GAMS compliation, assuming that this is item "3" on the Execute menu:

- Under Key/Key Sequence, check `ALT` and then use the drop-down dialog to find `C`.
- Under Functions, click the down-arrow and scroll through to find the item `ExecUserCommand3`
- Click Add

Now save the file:

- Click SaveAs
- Select a location — a good choice is with your other `PFE-GAMS` files, say in `c:\gams\users` — and a name, say `gams.key`; click Save; click OK.

Finally, tell `PFE-GAMS` to load this file at startup:

- Click Options then Preferences, and in the resulting dialog under Category, click KeyMapping.
- In the right-hand part of the dialog, click the radio button `Use this file`
- Click on the "`...`" and find the keymap file you just saved; click Open.
- Back in the main dialog, click OK. You will need to restart `PFE` for this to take effect.

6.3 Read-Only Files

PFE "knows" if files have been altered, and will ask you whether you want to save any altered files before it will Compile or Run a GAMS program. You can save yourself some effort by declaring certain files as Read-Only from the start. Here's how to make GAMS's `.lst` files Read-Only. In `PFE-GAMS`:

- Click Options then Default Modes
- Make sure the radio button marked `Files whose name is in a mode group` is checked; click Edit Mode Groups.
- In the resulting dialog, click New. In the new dialog, provide a name like `Lst Files`, and under FileName Patterns enter `*.lst`. Click OK twice to return you to the top-level Default Modes dialog.
- Click Edit Modes...
- In the new Mode Type dialog, select File Properties and check `File is Read-only in all windows`.
- If you'd like a visual reminder that the file is read-only, you can change the appearance of the file by selecting Text Colours under Mode Type, then Use Specific Colours and selecting (say) light gray for the Normal Background in the Change Object Colour dialog.
- Click Apply Now then OK (to return you to the Default Modes dialog) then Close.

6.4 Autoloading results

You can set up `PFE-GAMS` to automatically load files produced by the run: if you have an old version of the file already loaded into `PFE`, doing this avoids the dialog which tells you that the file has been changed, and asks your permission to update it.

To force `PFE-GAMS` to always load the `.lst` file produced by your run, add the line

```
c:\pfe\pfe32 /r gams %1%2.lst
```

to the end of your batch file `pfegams.bat`.

You can go even further and *conditionally* load a file. For example, to load the `.lst` file after a compilation only if something went wrong, add the following

to `pfegams.bat` *immediately after the call to GAMS* (in particular, this must precede the call to the `wait` program if you use it):

```
if "%4"=="c" goto testit
goto end
:testit
if errorlevel 1 c:\pfe\pfe32 /r gams %1%2.lst
:end
```

Note that this relies on the batch file containing the string `a=c` with the "`a`" being the fourth batch parameter, and the "`c`" being lowercase. If you alter the batch file you will have to check that this is still correct.

7 Debugging

If something goes wrong, you will need to see what information `PFE` is passing to the batch files, and to do this, you will need the DOS window created by the batch files to stay on the screen after the batch file finishes, long enough for you to diagnose what's going on. To have the window persist, edit the batch files in `c:\gams\users` and add a line saying

```
c:\gams\users\wait 10
```

to the end of the files (or a larger value if you're a slow reader): the effect of this is that the window remains longer on screen before being closed by `PFE`, and you can see what the batch files are doing.

If you are doing complicated parameter passing, you may want to see directly what each batch file parameter is. One way to do this is to add a series of lines of the form

```
echo Parameter 1 is %1
```

to the batch file you're testing, and then use the `wait` feature described above to make the information remain on screen.