

Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 46
February 2021

This Newsletter addresses the following

1	GAMS Releases 32-34	2
1.1	GAMS system	2
1.1.1	Changed option defaults	2
1.1.2	A log file full of links	2
1.1.3	Dumping out information on define GAMS items	2
1.1.4	What files have I used and What is status of my control variables?	2
1.1.5	Marking places in the LXI file	3
1.1.6	Finding an unmatched \$IFTHEN	3
1.1.7	Some other changes	4
1.2	Solvers	4
1.2.1	Solvers dropped from the system	4
1.2.2	Solvers with new features	5
2	GAMS Studio	5
2.1	Automatically opening include files	5
2.2	Changing Syntax colors	5
2.3	Hiding code on the STUDIO display	5
2.4	Recalling and replicating previous cursor movements	6
2.5	Other Developments	6
3	GAMS ENGINE	6
4	MIRO releases 1.1, 1.2 and 1.3	7
4.1	Licensing	7
4.2	Importing and exporting scenarios	7
4.3	Importing Excel and text files	7
4.4	New Pivot table	7
4.5	Other developments	8
5	Running batch jobs	8
6	On line Basic, Advanced and Combined courses offered soon	8

1 GAMS Releases 32-34

Since the last newsletter we have new releases 32, 33 and 34 with release notes being at https://www.gams.com/latest/docs/RN_MAIN.html. When we peruse these notes things that strike of broad interest are:

1.1 GAMS system

1.1.1 Changed option defaults

GAMS has changed the default values for a number of options. Changes include new values for:

- *logOption* which now defaults to standard output (3),
- *errMsg* (1) which causes error messages to be included in the LST file where they arise,
- The *PS* and *PW* page attributes were made much larger,
- *OPTCR* the integer programming solution tolerance is now a lot smaller (0.0001 as opposed to 0.1),
- The *resLim* and *iterLim* time and iteration limits that are now bigger.

One can revert back to the older option settings by replacing *gamsconfig.yaml* with the file *gams31config.yaml* (which is in the GAMS system directory).

1.1.2 A log file full of links

Links are now placed in the log file that address source locations and give you the ability to link to include files and the place where they are included of which my model has perhaps thousands of includes and I see a forest of blue links in the IDE. As of now I am afraid that I don't think this is well implemented. My whole log file is now almost entirely colored links. This compromises the ability to visually identify: a) places in the LST file to navigate to (although the LXI file also does this); and b) program created GDX and PUT files that one could automatically open. I believe that the intent of the added links is to allow navigation to slow statements, included files and perhaps the location of execution errors. But as of now way too many links are present. I hope GAMS rethinks it.

1.1.3 Dumping out information on define GAMS items

A more user-oriented version (*dmpUserSym*) of the old option *DMPSYM* has been added that lists the parameters, sets etc. defined in a GAMS program, their dimensionality and size. In my view this needs a little more work as it is oddly formatted, does not do a good job of organizing the list (could be alphabetical) and contains the mysterious, probably non-informative items (*DIM-OK*, *ACCESS*, *SPECVAL*, *EXTERN*, *TABLE*, *DOMAIN*, and *LAGLEAD*).

1.1.4 What files have I used and What is status of my control variables?

When I am working with what I call conditional compile I often want some information on the control variables. This involves using commands like *\$setGlobal* to establish control variables then later, *\$if*, *\$endIf* etc to test their values. (For an explanation see the McCarl guide chapter on

Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 46
February 2021

conditional compile). This used to require using *\$show* which gave you a lot more than a list of control variables including an unreadable list of your macros.

GAMS has now added *\$showfiles* which lists the files used in the program, *\$showMacros* that shows macros (and really needs some effort on formatting if it is to be of much use), and *\$showVariables* that shows the status and values of any used control variables in a much more concise listing (but one that needs alphabetization).

1.1.5 Marking places in the LXI file

The *LXI* navigation window now contains quoted text used in *display* statements and by clicking on that one can navigate to it in the *LST* file. As I am debugging GAMS programs I employ a lot of *display* statements. This often involves displaying something before and after a calculation to see where it is taking on unusual values. Then the *LXI* file contains multiple references to the same thing and I end up either marking it with a quoted element in the *Display* statements or searching the *LST* file for the line number of that *display* then going to that spot in the source code. Now the quoted elements are included in the display go into the *LXI* navigation window. So if I employ the command

```
Display "firstplace", xparam;
```

Then the text "*firstplace*" will appear in the *LXI* navigation window making finding things a little easier. It might be nice if a right click on that or some other mouse action took you into the source file location where the *display* appears.

1.1.6 Finding an unmatched \$IFTHEN

In the conditional compile world you can use the command \$IFTHEN and this must be matched somewhere later in the code with a \$ENDIF. Sometimes you end up without a matching \$ENDIF and this causes an error. Now there is a [link to the place](#) in the source code identifying the statement that caused this. For example here is a log file excerpt

```
--- $ifthen from 1st line 3168091 (line 13 in source file scenario_loop.gms) has no matching $endif
--- scenario_loop.gms(13) 875 Mb 1 Error
*** Error 674 in C:\gamsstuff\fasom\sourceandstudies2020forest\studies\climatechange-ag\scenario_loop.gms
    Could not find $elseif/else/endif
```

This is especially useful because the **error message** in the LOG and LST files point to the very bottom of your program saying in effect I cannot find the ENDIF. But you really needed to know which IFTHEN is unmatched and this started a search for that statement. But now the LOG file shows where that is which is quite helpful. I do think it would be nice if that message also appeared in the LST file and perhaps the message might appear just after the 674 error message in both LST and LXI.

Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 46
February 2021

1.1.7 Some other changes

A wide variety of other things have been done including

- Extensions on the names of the *save* and *restart* files are now retained.
- Searches using *inputDir* now include GDX files loaded with \$gdxIn.
- The advanced basis/solution saving procedure (*option savePoint*) now has new settings that cause the resultant GDX file to be saved in the scratch directory.
- The *Embedded Code Facility* is now fully supported.
- Execution Speed was improved for some operations with out-of-order index positions. For more about the underlying issue see the discussion in the McCarl guide in the section on "Speed" or the in the user guide under "Finding the Causes for Slow Program Execution"
- GAMS has dropped support for Python 2 and now supports versions 3.6, 3.7, 3.8 and 3.9.
- A new experimental, beta code version has been developed that enables the use of GAMS in Jupyter notebooks through PYTHON. This is discussed in the document [How to use GAMS in Jupyter Notebooks](#).
- There is also a new *gams2numpy* API package that does bulk movement of data between GAMS and Python into numpy arrays and permits transfer into panda dataframes. For now this is not documented but is resident in new versions of GAMS in the apifiles subdirectory of the GAMS system and can be imported into embedded PYTHON code. There are GAMS examples at https://www.gams.com/34/docs/apis/examples_python/g2np_example1_8py.html and https://www.gams.com/34/docs/apis/examples_python/g2np_example2_8py.html
- I also see mention of a couple of APIs under PYTHON that are *gamstransfer* which simplifies the work of reading and writing GDX files and *gams_engine* that allows the creation of Python clients for [GAMS Engine](#). I do not believe there are write-ups for these but again they are resident in the apifiles subdirectory and can be imported into embedded PYTHON code. There are examples for gamstransfer at https://www.gams.com/34/docs/apis/examples_python/gt_example1_8py.html https://www.gams.com/34/docs/apis/examples_python/gt_example2_8py.html and GAMS_engine is used in <https://www.gams.com/engine/clients.html#custom-clients>.

1.2 Solvers

1.2.1 Solvers dropped from the system

A number of solvers were or are about to be dropped. This includes AMPL, BDMLP, BENCH, COUENNE, LINGO, LS, MPECDUMP, and PYOMO. For those who wish to still access these solvers

- Paths to some of them through CONVERT.
- A PYTHON based alternative to LS is given in the [LeastSquares.gms](#) example model within the GAMS Data Library.
- A statement that indicates that there are other faster free solvers than BDMLP.
- A way to reinstall COUENNE using a zip file available [here](#). That zip file contains the solver libraries and a version of a gamsconfig.yaml that includes instructions to make the solver available within GAMS.

1.2.2 Solvers with new features

- Many solvers were updated and a few seem to have some new major features. To us the items worthy of mention are
 - KNITRO now has an improved MIP/MINLP branch-and-cut solver and enhancements to the presolver.
 - XPRESS now has new solvers: XPRESS SLP and XPRESS Knitro that solve nonlinear problems. The latter requires a GAMS/Knitro license in addition to an XPRESS license.
 - CONOPT has a number of new numerical precision features
 - CPLEX and CPLEXD have been merged and now have common features. Also changes were made in options for MIP, infeasibility (conflict or IIS) analysis and presolve. Beyond that GAMS has updated the CPLEX documentation to cover some options relevant to multi-objective optimization.
 - GUROBI has had improvements in MIP options, presolve and BIGM/feasibility relaxation features Remote access to GUROBI now involves the license file.
 - LINDO/LINDOGLOBAL added capabilities for detecting irreducible infeasible and unbounded sets.
 - A number of the possible CONVERT output formats were eliminated

2 GAMS Studio

GAMS STUDIO has a number of new features the main ones of which are

2.1 Automatically opening include files

One can open files that are included in a program by moving the cursor on top of a file name addressed an include statement. Then can cause STUDIO to open that file by either using the F2 command or a Ctrl+ Click.

2.2 Changing Syntax colors

- One can now manipulate STUDIO syntax coloring using File>Settings and the Colors tab. The first time you do it you have to click the Copy button to gain access. After that by clicking on the small square colored boxes allows you to choose new colors for many syntax categories. You can also employ bolding and italics.

2.3 Hiding code on the STUDIO display

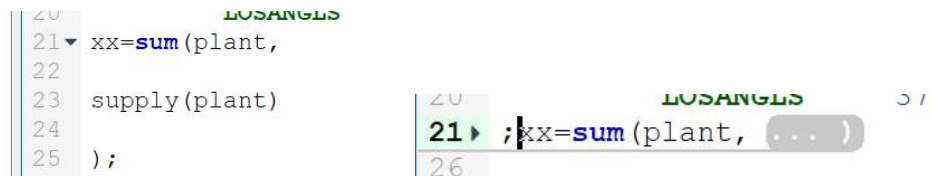
STUDIO now permits you to visually hide or collapse blocks of code from viewing on your screen using a feature called folding. In particular a block of code surrounded by a number of operators can be collapsed into a single line on the screen. Note

- This does not change the contents on your code just what is displayed.
- The operators that can surround the code include
 - Pairs of parentheses that span multiple lines
 - \$ifthen to \$endif and
 - Dollar Control Options that come in pairs.
 - \$Ontext and \$Offtext
 - \$On/\$OffEmbeddedCode,

**Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 46
February 2021**

- \$On/\$OffEcho,
 - \$On/\$OffPut,
 - \$On/\$OffExternalInput and
 - \$On/\$OffExternaloutput,
 - The new commands \$onFOLD and \$offFold.
- When these operators surround lines of GAMS code this defines blocks that can be collapsed at the users discretion.
 - The block is collapsed by moving the mouse within it and typing Alt+L. Subsequently it can be expanded by using Alt+L when you cursor is on the collapsed line.
 - All such blocks in your code can be collapsed by using Alt+O and this is reversed using Shift+Alt+O. I actually wonder if this collapses too much given that it collapses material within matched parentheses.

A small example is perhaps useful. In the left screen shot just below there is a potentially collapsible block between the open and close parentheses used in the sum. Moving the cursor within that block to the word "supply" and hitting Alt-L folds the block of code so it appears as in the screen shot on the right.



```
20
21 xx=sum(plant,
22
23 supply(plant)
24
25 );
```

```
20
21 ;xx=sum(plant, ... )
26
```

The presence of collapsed code is designated by the small grey box as it appears just above and contains three dots.

2.4 Recalling and replicating previous cursor movements

Recent cursor movements can now be recalled and replicated. Hitting alt and the right or left arrow moves you backwards or forwards through the recent cursor movements. There are other variants on windows involving mouse buttons and Ctrl+[/] on Mac OS X.

2.5 Other Developments

- Models can be executed remotely on the NEOS servers via the menu GAMS -> Run NEOS.
- One can now print a number of file types by using the File print choice or Ctrl+P.
- GAMS has developed an API for linking with MATLAB. It allows: a) Exchange of input data and model results; b) Creation and running of GAMS models; c) Customizing GAMS options and d) Efficiently solving sequences of models. This is discussed in the document [A new object oriented MATLAB API](#).

3 GAMS ENGINE

A first version of server software (GAMS Engine) was just released. It facilitates remote runs of GAMS on centralized computing resources whether on the premises or in the cloud. Engine

Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 46
February 2021

installation provides a gateway service or “broker” for accepting jobs from GAMS Studio, MIRO Desktop, MIRO Server, or custom clients. A web interface is also provided that allows submitting Jobs and user administration. On the remote platform jobs are placed in a queue, and then are assigned to available processors. A short write-up on Engine is at https://www.gams.com/sales/engine_facts/ and a more complete description at <https://www.gams.com/engine/>.

As mentioned above there is a Python API called `gams_engine` that allows the creation of Python clients for linking to [GAMS Engine](#). I do not believe there is a write-ups on this but it is resident in the `apifiles` subdirectory and can be imported into embedded PYTHON code. There is an example at

<https://www.gams.com/engine/clients.html#custom-clients>.

4 MIRO releases 1.1, 1.2 and 1.3

[MIRO](#) has had release of versions 1.1-1.3 since the last newsletter. Some comments on MIRO and the releases follow.

4.1 Licensing

Use with GAMS requires a GAMS licensed MIRO connector. The connector is free for academics, but commercial users have to pay for it.

4.2 Importing and exporting scenarios

MIRO can now import and export scenarios via MIROSCEN files. Those files contains all scenario metadata including views and attachments.

4.3 Importing Excel and text files

- One can now import Excel (xls/xlsx) files with an index sheet. The syntax of the index sheet is very similar to that used in GDXXRW. Read [here](#) for more details.
- Delimited text files that are imported or exported can use the delimiters ", tab ; | :". MIRO tries to detect the delimiters automatically but resorts to manual selection if it cannot uniquely find the delimiters. It also contains a wizard that supports a user-defined mapping of column headings to symbol headings. See the discussion at <https://www.gams.com/miro/start.html#import-data>.

4.4 New Pivot table

A new form of pivot tables has been introduced and are documented in <https://www.gams.com/miro/charts.html#pivot-chart>. They now

- Allow you to filter, aggregate and pivot data.
- Support data visualization as charts (line chart, bar chart, etc.) or heatmaps.
- Permit slices, or views of data can be exported with charts transferring as a PNG file and data as a CSV file.

- Allow naming and storage of "views" - the current representation of the data - and subsequent reloading through the scenario data. Then they can be accessed using the "Load view" button.
- When exporting spreadsheets includes an index sheet that reveals the layout of the exported data.
- Can now also be used for data input.
- One can hide columns in big data tables.
- Can handle large datasets (> 1 million elements)

4.5 Other developments

- Tabs for output symbols can now be hidden.
- Added legend configuration options to time series diagrams.

5 Running batch jobs

I do a lot of running of batch jobs using commands like

```
$set env ide=%gams.ide% lo=%gams.lo%
execute 'gams aaa.gms %env% s=. \t\a1 rf=list pw=150 ps=0'
execute 'gams model_forthisstudy.gms %env% r=. \t\a1 s=. \t\a2 '
execute 'gams scenario_loop.gms %env% r=. \t\a2 s=. \t\a6 u1=base pw=333'
execute 'gams untitled_2.gms %env% r=. \t\a6 pw=144'
```

as this runs it automatically closes the LST files for aaa, model_forthisstudy etc that you might have open in the IDE or STUDIO. Then I have to go in and manually locate them to see the results. But one of the GAMS staff informed me that I could use the

```
put_utility 'click' / 'filename ';
```

command to put links to the LST files in the LOG file for this run. So now I run

```
set env ide=%gams.ide% lo=%gams.lo%
execute 'gams aaa.gms %env% s=. \t\a1 rf=list pw=150 ps=0'
put_utility 'click' / '%system.FP%aaa.lst' ;
$setglobal yeartouse 2020
execute 'gams model_forthisstudy.gms %env% r=. \t\a1 s=. \t\a2 '
put_utility 'click' / '%system.FP%model_forthisstudy.lst' ;
execute 'gams scenario_loop.gms %env% r=. \t\a2 s=. \t\a6 u1=base pw=333'
put_utility 'click' / '%system.FP%scenario_loop.gms.lst' ;
execute 'gams untitled_2.gms %env% r=. \t\a6 pw=144'
put_utility 'click' / '%system.FP%untitled_2.lst' ;
```

which gives me clickable links to my LST files for when I need to look into them. I like this.

6 On line Basic, Advanced and Combined courses offered soon

This year I will again be teaching my family of GAMS courses for basic and advanced users. These courses will be offered online in May. Dates, times and content are

Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 46
February 2021

- Basic to Advanced GAMS class May 10 - May 14 and May 17, 2021 (six 5 1/2 hour sessions each 8-1:30 central). The course spans from Basic topics to an Advanced GAMS class. Details are found at https://www.gams.com/courses/2021_05_basic_to_advanced-gams-modeling_mccarl/Basic_to_Advanced_GAMS_Modeling_online.pdf with the course outline at https://www.gams.com/courses/2021_05_basic_to_advanced-gams-modeling_mccarl/introduction_basic_to_advanced_gams_class_online.pdf
 - Basic GAMS class May 10 - May 13 (four 5 1/2 hour sessions each 8-1:30 central) The course starts assuming no GAMS background. Details are found at https://www.gams.com/courses/2021_05_basic-gams-modeling_mccarl/Basic_GAMS_Modeling_online.pdf with the course outline at https://www.gams.com/courses/2021_05_basic-gams-modeling_mccarl/introduction_basic_online.pdf .
 - Advanced GAMS class May 12 - May 14 and May 17, 2021 (four 5 1/2 hour sessions each 8-1:30 central). The course is for users who have a GAMS background. Details are found at https://www.gams.com/courses/2021_05_advanced-gams-modeling_mccarl/Advanced_GAMS_Modeling_online.pdf with the course outline at https://www.gams.com/courses/2021_05_advanced-gams-modeling_mccarl/Introduction_adv_gams_class_online.pdf
- Note I also give custom courses for individual groups a couple of times a year.

Further information and other courses are listed on <http://www.GAMS.com/courses.htm>

February 17, 2021