

Bruce McCarl and Chengcheng Fei's GAMS Newsletter Number 49

March 2024

This Newsletter addresses the following

1	GAMS Releases 44-46	1
1.1	GAMS system	1
1.2	CONNECT	1
2	Put_toExcel_connect	2
3	TRANSFER	4
4	Studio	4
5	GAMSPy	4
6	Dropping and For Some Getting Back the McCarl Gude	5
7	Basic, Advanced and Combined courses offered soon	6
8	Unsubscribe or subscribe to future issues of this newsletter	7

1 GAMS Releases 44-46

Since the last newsletter there have been new releases 44 through 46 with the formal GAMS release notes available at https://www.gams.com/latest/docs/RN_MAIN.html. When we peruse these notes the things that strike us to be of broad interest are:

1.1 GAMS system

- *gdxload* – a new execution time command that loads data from a GDX file much as done by using *execute_load*, but where one can control whether the data replaces that in an existing symbol or merges it. Domain checking can also be used or turned off by *gdxload* using the *filter* command.
- The default for the option *threads* was changed to 0. That setting causes the solver to choose the number of processors to use. There are some solver-specific aspects of this as covered in the release notes.

1.2 CONNECT

GAMS continues to work on CONNECT which is a platform independent means of interchanging data from and to other platforms. Currently CONNECT allows data interchanges with spreadsheets, SQL databases, GDX and CSV files. The CONNECT framework consists of a database and a set of context-specific agents that move data back and forth between its database and a variety of file types. One can also manipulate items within the CONNECT database. These agents are used in an ordered set of steps that load, transform and unload the data.

To use CONNECT, users must develop instructions that use CONNECT agents in a sequence of actions involving: a) reading data from various file types or those stored in a GAMS program into the CONNECT database, b) transforming data in the CONNECT database, and c) writing data from the CONNECT database to several file types or to GAMS. Instructions are passed using [YAML syntax](#).

According to the CONNECT [write up](#) there are available agents to:

- Read items from GAMS into the CONNECT database and pass CONNECT database items into GAMS. Note GAMS is again using a two-step pass as is done when using GDX. Namely, when one is passing data out of GAMS to a spreadsheet or other file types, the first step is moving items from GAMS to CONNECT and then another is moving data from CONNECT to the destination. Similarly moving data from an external file into GAMS requires that one first move data into CONNECT and then move the data into GAMS.
- Read items from several file formats into the CONNECT database or pass items to those types of files. The file formats allowed as of now are GDX, EXCEL, SQL, and CSV;
- Change some domain information or deal with domain violations within an item in the CONNECT database.
- Filter the data by eliminating selected elements that are not to be passed into the external file format. One can also filter numerical entries using specified limits.
- Reorder sets when moving data including possibly eliminating some sets (doing things like defining $y(i,j,k)$ from $x(k,j,i)$ or constructing $y(i,k)$ as the $sum(j,x(i,j,k))$)
- Concatenate data from several GAMS parameters into a composite array with the name of the parameters becoming an entry in the first index position of the resultant composite item.
- Manipulate labels based on a set where the explanatory text is used as a new name in a CONNECT parameter for matching set elements.
- Execute custom Python code over database items.

CONNECT remains in Beta and it is likely that will occur in the next two releases.

Connect also just added the *ExcelReader* and *ExcelWriter* agents for reading and writing Excel files and these are used in the *put_toexcel_connect* code file discussed in the next section. To use the features that are employed in the file, users must have GAMS version 46 or later.

2 Put_toExcel_connect

I wanted to try out CONNECT and with the help of Clemens Westphal from GAMS we reimplemented the *put_toexcel* batinclude file I developed years ago using GDXXRW so it used CONNECT. We created an alternative version called *put_toexcel_connect* that will likely soon

replace the version of *put_toexcel* distributed with GAMS. In it, the instructions employing features of CONNECT are embedded in a BATINCLUDE file.

A GMS file of that code, an example using it and a writeup are in the [zip file here](#). Note this code may change as CONNECT moves out of Beta status.

The basic idea in the code is one can put a GAMS parameter or a set into a workbook with control over how many of the set dimensions go into rows and columns plus identify the placement of those set dimensions across the rows and columns in Excel.

Two simple examples are as follows

```
$batinclude put_toexcel_connect 2 1 x (i,j,k) k i j workbook sheetname
```

or

```
$batinclude put_toexcel_connect 1 2 x (i,j,k) j i k workbook sheetname
```

In the top example, the three dimensional item *x* is put into an excel file called *workbook* on the sheet called *sheetname*. There we put **two** of the elements associated with sets in the row and **one** in the columns. In this case, we say put the sets **k** and **i** into the rows and **j** in the columns. The second example is similar but places **one** dimension in the rows (**j**) and **two** in the columns (**i** and **k**). More on this syntax is given in the writeup.

To illustrate the usage of CONNECT agent, this file uses the following CONNECT agents in the order specified. Namely in *put_toexcel_connect*

- GAMSReader:** is used to move the named item from the GAMS program into the CONNECT database.
- Filter** is used to reduce the entries within a data item to just those that are to be put into Excel. This reduces the elements in sets that are controlled as quoted items or in loops in GAMS. Essentially when a set is controlled via a quoted element (i.e. $x(i,j,k)$ is referenced as $x('cars',j,k)$) then all data entries for the set in the position where the quoted element appears, and that do not contain the quoted element are dropped (i.e. all $x(i,j,k)$ elements where the *i* element is not 'cars'). Similarly, when the GAMS program loops over *i* and within that loop calls the batinclude then the filter drops all $x(i,j,k)$ data entries where the entry associated with the *i* set but are not associated with the currently controlled element of *i*.
- Projection** is used to rearrange the data item into a new item with the sets in the order they are prescribed to be placed to Excel
- ExcelWriter** is used to write the filtered, projected parameter to Excel

There is also Python code that controls the process and a segment that updates the next available row number in *excelrownumber*.

3 TRANSFER

GAMS has also been working on a GAMS to other programming language linking product called TRANSFER and they have released Python and R versions. It provides a way to exchange, augment and maintain data in interaction with a programming language like Python, R or Matlab. There have been some Python related developments that are covered in the release notes. TRANSFER Matlab remains in Beta but this is expected to change soon.

4 Studio

Studio has had minor bug fixes, but nothing major that we can detect. IDE users may want to switch as GAMS has added a message to it saying they will soon drop it since it was programmed in a now unsupported version of the Delphi language that also embodied a now unsupported third party created editor module.

5 GAMSPy

Facing the growing prevalence of Python among computer users and the emergence of optimization related packages like PYOMO, the GAMS company decided to create a package that embedded the algebraic modeling features in GAMS for use within Python applications. GAMS calls that package GAMSPy and in the [GAMSPy documentation](#) states that “GAMSPy combines the high-performance GAMS execution system with the flexible Python language, creating a powerful mathematical optimization package. It acts as a bridge between the expressive Python language and the robust GAMS system, allowing you to create complex mathematical models effortlessly.”

To do this, they have created a PIP based installation package and a set of classes that represent GAMS sets, parameters, and other model components for inclusion in Python that declare sets, parameters, and other model components. They introduce a container class that manages essential data structures such as sets, parameters, variables, equations, and models. Those containers collectively allow definition of an optimization problem. Therein, much like within GAMS, every item needs to be declared within containers. Also, GAMSPy allows users to manipulate data and solution information plus create model result displays using Python libraries like Numpy, Pandas, Networkx. Furthermore, through the use of the broad variety of Python capabilities, it allows users to import and export data and results to many data formats.

As of now, GAMSPy is in beta with plans for a full release in the next few months. Moreover, it is probable that GAMSPy will be free to academic users and that it will be distributed with the ability to freely use a number of solvers (those not requiring a commercial license). Much more on capabilities and use is covered on GAMSPy in the writeup available at

<https://gamspy.readthedocs.io/en/latest/index.html#> with more on the product page at https://www.gams.com/sales/gamspy_facts/ .

6 Dropping and For Some Getting Back the McCarl Guide

With the next major release, GAMS will drop the presence of the McCarl Guide. This is a consequence of a five year or so ago commitment of GAMS to develop and maintain their own User Guide couple with an accompanying lack of document upgrades for the McCarl Guide. This occurred as of GAMS release version 24. It is also being dropped because of what GAMS perceives as confusion in the user community from having two documentations. Certainly, the McCarl Guide cannot be the sole User Guide since it is missing all new features that have been added in some GAMS v.22 releases during the last five or so years. But I think it still holds substantial value for the new user. Why? GAMS personnel often expressed the feeling the McCarl Guide was excessively long and that a much more summary approach was needed. Thus, they developed a User Guide that just laid out the syntax without a lot of discussion and often without examples or cross references. On the other hand, this leads to many passages in the GAMS version where features are described in a form that simply leaves one guessing as to what the feature is about and why you might use it. For example, consider the “filtered” command that I looked up today in association with `gdxLoad`. There the associated passage in the GAMS User Guide is reproduced in red below

filtered (*string*): Switch between filtered and domain-checked read from GDX

Available: Command line, Option statement

The command line parameter `Filtered` initializes both, the option `Filtered` to control the behavior of `gdxLoad` and the dollar control option **`$on/offFiltered`** for the compile-time equivalent behavior.

Default: on

Value Meaning

Off Load domain checked

On Load filtered

Perhaps such a documentation passage is clear to many, and we are just not smart enough to figure it out. But it leaves us guessing about what this really does and why we might choose one option or the other. It also means if we want to know what it does, we either have to dialogue with technical support or experiment. (Note from other reading “load filtered” may well mean GAMS will read in all entries that correspond to things in the unique element list ignoring the rest but that is a pure guess. Similarly, we would guess that the domain checked load will only bring in items which match the known domain for an item). Additionally, we wonder whether some kind of messages will appear for things that fail to load. We have no idea.

So where does this leave us. For those wanting the longer more instructionally oriented aspects of the McCarl Guide you can still get it. We personally think that is superior for new users. So we got GAMS to move the McCarl Guide and all associated files and examples to the contributed documentation section on GAMSWORLD. It is resident there in the form of a zip file (note it is buried as the last entry at the bottom of the second page in a section containing McCarl's contributed documentation). That file is now accessible through <https://forum.gamsworld.org/viewtopic.php?p=27375#p27375> although we don't know how persistent that link will be. There the file *mccarl.zip* delivers pdf and chm versions of the McCarl Guide plus delivers a subdirectory with all the example GMS files. We have been told that this if unzipped in the GAMS system directory will put it all in there, although we do not believe that STUDIO will have a reference to it. In turn you would need to place the pdf on your desktop or in some other convenient place for your use. You can make the IDE see it by unzipping in the *docs* directory of the GAMS system. You can also download the source through a second zip file called *mccarlsources.zip* if you want to edit, update and add to the documentation. To do this you would need to use the program "Help and Manual" to integrate the material and allow you to rebuild the chm and pdf.

As a final note, we use both Guides and often just the GAMS one as we know most of the McCarl Guide well and sometime just want to look up syntax or command options. The McCarl Guide though takes one step by step through many commands in a much more extensive fashion and as such I think would be particularly useful to new users. But it is missing many aspects since version 24.9. However, the basics are treated in both Guides. Also, for those familiar with the McCarl Guide you will find that the GAMS User Guide contains numerous exact or slightly modified passages and examples from the McCarl Guide.

7 Basic, Advanced and Combined courses offered soon

This year, McCarl will again be teaching GAMS courses for basic and advanced users. These courses will be offered in early June via ZOOM. Dates, times and content are

- Basic to Advanced GAMS class June 3, 2024- June 10, 2024 (6 sessions of 6 hours per day via ZOOM with the weekend off). The course spans from Basic topics to an Advanced GAMS class. Details are found at <https://www.gams.com/courses/2024-06-basic-to-advanced-gams-modeling-mccarl-copy/>.
- Basic GAMS class June 3, 2024- June 6, 2024 (4 sessions of 6 hours per days via ZOOM). The course starts assuming no GAMS background. Details are given at <https://www.gams.com/courses/2024-06-basic-gams-course-mccarl-copy/>
- Advanced GAMS class June 5, 2024- June 10, 2024 (4 sessions of 6 hours per days via ZOOM with the weekend off). The course is for users who have a GAMS background. Details are found at <https://www.gams.com/courses/2024-06-advanced-gams-modeling-mccarl-copy/>.

Further information and other courses are listed on <http://www.GAMS.com/courses.htm>. Note I also give custom courses for individual groups a couple of times a year.

8 Unsubscribe or subscribe to future issues of this newsletter

Please unsubscribe through the web form available at: <https://gams.us18.list-manage.com/unsubscribe?u=f1497c76718404eae593beb11&id=45ccea2ee0>

Those who wish to subscribe to future issues can do this through the newsletter section of <http://www.GAMS.com/maillist/index.htm>.

This newsletter is not a product of GAMS Corporation although it is distributed with their cooperation.

March 25, 2024