GAMS Python API documentation

version 24.0

GAMS Development Corporation, Washington, DC, USA

Contents

Chapter 1

GAMS Python API

Version

24.0

1.1 Overview

The gams namespace provides objects to interact with the General Algebraic Modeling System (GAMS). Objects in this namespace allow convenient exchange of input data and model results (GamsDatabase), help to create and run GAMS models (GamsJob), that can be customized by GAMS options (GamsOptions). Furthermore, it introduces a way to solve a sequence of closely related model instances in the most efficient way (GamsModelInstance).

A GAMS program can include other source files (e.g. $include), load data from GDX files (e.g. $GDXIN or execute_load), and create PUT files. All these files can be specified with a (relative) path and therefore an anchor into the file system is required. The base class GamsWorkspace manages the anchor to the file system. If external file communication is not an issue in a particular Python application, temporary directories and files will be managed by objects in the namespace.

With the exception of GamsWorkspace the objects in the gams namespace cannot be accessed across different threads unless the instance is locked. The classes themself are thread safe and multiple objects of the class can be used from different threads (see below for restrictions on solvers that are not thread safe within the GamsModelInstance class).

This alpha version of the gams namespace lacks support for the following GAMS components:

Acronyms, domain checking, support for GAMS compilation/execution errors (GamsJob.run just throws an exception), structured access to listing file, and proper support for solver options.

Currently only CplexD and Gurobi fully utilize the power of solving GamsModelInstances. Some solvers will not even work in a multi-threaded application using GamsModelInstances. For some solvers this is unavoidable because the solver library is not thread safe (e.g. MINOS), other solvers are in principle thread safe but the GAMS link is not (e.g. Cplex, note there is a thread safe version of Cplex called CplexD which lack some minor features e.g. support for BCH framework). Moreover, GamsModelInstances are not available for quadratic model types (QCP, MIQCP, RMIQCP).

1.2 Getting started

The object oriented GAMS Python API is built on top of the different low level component API's and provides convenient access to GAMS from within Python. Examples using the API are located in 'apifiles/Python' while the API itself is found in 'apifiles/Python/api'. The GAMS Python API requires Python 2.7. The bitness of the Python version has to be the same as the bitness of the GAMS system. Assuming that the current directory is <Path/To/GAMS>/apifiles/Python the API can be used as follows:

- Installing the API and the required low level API's to Python site-packages:

  ```
  cd api && python setup.py install && cd ..
  ```

- Using the API without installing:

  ```
  export PYTHONPATH=api  (on Windows: set PYTHONPATH=api)
  ```

Note that on non windows platforms it is required to set the load library path to the GAMS system directory to run scripts using the GAMS Python API.

Linux:

```
export LD_LIBRARY_PATH=<Path/To/GAMS>:$LD_LIBRARY_PATH
```

OS X:

```
export DYLD_LIBRARY_PATH=<Path/To/GAMS>:$DYLD_LIBRARY_PATH
```

Running the transport1.py example:

```
python transport1.py
```

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

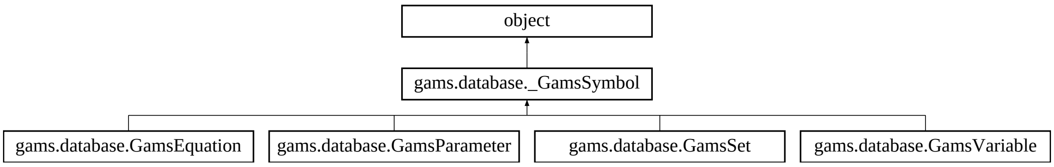
Chapter 4

Class Documentation

4.1 gams.database._GamsSymbol Class Reference

This is the representation of a symbol in GAMS.

Inheritance diagram for gams.database._GamsSymbol:


object
gams.database._GamsSymbol
gams.database.GamsEquation
gams.database.GamsParameter
gams.database.GamsSet
gams.database.GamsVariable

Public Member Functions

- def __len__

  Retrieve the number of records of the GamsSymbol.
- def copy_symbol

  Copys all records from the GamsSymbol to the target GamsSymbol (if target had records, they will be deleted)
- def delete_record

  Delete GamsSymbol record.
- def clear

  Clear symbol.
- def find_record

  Find record in GamsSymbol.
- def add_record

  Add record to GamsSymbol.
- def first_record

  Retrieve first record in GamsSymbol.

Properties

- dimension property(get_dimension)

  Get GamsSymbol dimension.
- text property(get_text)

  Get explanatory text of GamsSymbol.
- name property(get_name)

Get GamsSymbol name.

- database property(get_database)

  Get GamsDatabase containing GamsSymbol.

- number_records property(get_number_records)

  Retrieve the number of records of the GamsSymbol.

4.1.1 Detailed Description

This is the representation of a symbol in GAMS.

It exists in a GamsDatabase and contains GamsSymbolRecords which one can iterate through. Derived classes are GamsEquation, GamsParameter, GamsSet and GamsVariable.

4.1.2 Member Function Documentation

4.1.2.1 def gams.database._GamsSymbol.add_record (*self*, *keys* = None)

Add record to GamsSymbol.

Parameters

keys	List of keys

Returns

Reference to added record

4.1.2.2 def gams.database._GamsSymbol.clear (*self*)

Clear symbol.

Returns

True if everything worked, else False

4.1.2.3 def gams.database._GamsSymbol.copy_symbol (*self*, *target*)

Copys all records from the GamsSymbol to the target GamsSymbol (if target had records, they will be deleted)

Parameters

target	Target GamsSymbol

Returns

True if everything worked, else false

4.1.2.4 def gams.database._GamsSymbol.delete_record (*self*, *keys*)

Delete GamsSymbol record.

Parameters

keys	List of keys

Returns

True if everything worked, else False

4.1.2.5 def gams.database._GamsSymbol.find_record (*self*, *keys* = None)

Find record in GamsSymbol.

```
print "Transportation costs from Seattle"
record = job.out_db.get_parameter("c").first_record(["seattle", " "])
```

Parameters

keys	List of keys

Returns

Reference to found record

4.1.2.6 def gams.database._GamsSymbol.first_record (*self*, *slice* = None)

Retrieve first record in GamsSymbol.

Parameters

slice	Define filter for elements whose record should be retrieved

Returns

Reference to record

4.1.3 Property Documentation

4.1.3.1 gams.database._GamsSymbol.number_records property(get_number_records) [static]

Retrieve the number of records of the GamsSymbol.

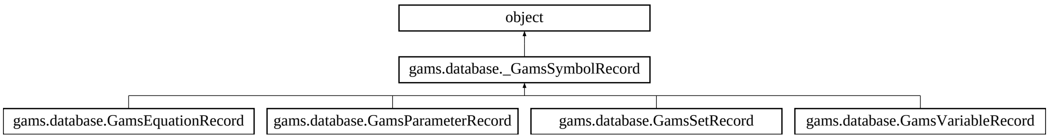
Note

This is the same as calling len(symbol)

4.2 gams.database._GamsSymbolRecord Class Reference

This is the representation of a single record of a GamsSymbol.

Inheritance diagram for gams.database._GamsSymbolRecord:


object
gams.database._GamsSymbolRecord
gams.database.GamsEquationRecord
gams.database.GamsParameterRecord
gams.database.GamsSetRecord
gams.database.GamsVariableRecord

Public Member Functions

- def move_next

  Iterate to next GamsSymbolRecord of GamsSymbol.
- def move_previous

  Iterate to previous GamsSymbolRecord of GamsSymbol.

Properties

- keys property(get_keys)

  Retrieve keys of GamsSymbolRecord.
- symbol property(get_symbol)

  Retrieve the GamsSymbol that contains this record.

4.2.1 Detailed Description

This is the representation of a single record of a GamsSymbol.

Derived classes are GamsEquationRecord, GamsParameterRecord, GamsSetRecord and GamsVariableRecord

4.2.2 Member Function Documentation

4.2.2.1 def gams.database._GamsSymbolRecord.move_next (*self*)

Iterate to next GamsSymbolRecord of GamsSymbol.

Returns

True if everything worked, False otherwise

4.2.2.2 def gams.database._GamsSymbolRecord.move_previous (*self*)

Iterate to previous GamsSymbolRecord of GamsSymbol.

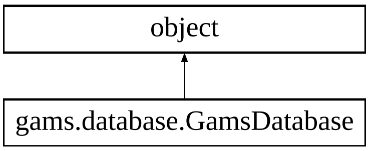
Returns

True if everything worked, False otherwise

4.3 gams.database.GamsDatabase Class Reference

An instance of GamsDatabase communicates data between the Python world and the GAMS world.

Inheritance diagram for gams.database.GamsDatabase:



Public Member Functions

- def __len__

  Retrieve the number of symbols in the GamsDatabase.
- def __del__

  Use this to explicitly free unmanaged resources associated with this GamsDatabase.
- def get_symbol

  Get GamsSymbol by name.
- def get_equation

  Get GamsEquation by name.
- def get_parameter

  Get GamsParameter by name.
- def get_variable

  Get GamsVariable by name.
- def get_set

  Get GamsSet by name.
- def add_equation

  Add equation symbol to database.
- def add_variable

  Add variable symbol to database.
- def add_set

  Add set symbol to database.
- def add_parameter

  Add parameter symbol to database.
- def export

  Write database into a GDX file.
- def clear

  Clear all symbols in GamsDatabase.
- def compact

  Dispose temporary external resources in database (advanced use)

Properties

- number_symbols property(get_nr_symbols)

  Retrieve the number of symbols in the GamsDatabase.
- workspace property(get_workspace)

  Get GamsWorkspace containing GamsDatabase.
- name property(get_name)

  Get GamsDatabase name.

4.3.1 Detailed Description

An instance of GamsDatabase communicates data between the Python world and the GAMS world.

A GamsDatabase consists of a collection of symbols (GamsDatabase implements __iter__() and next() to allow iterating conveniently through the symbols in a GamsDatabase). The symbol types available for a GamsDatabase correspond to the symbols types know from the GAMS language: Set, Parameter, Variable, and Equation are represented in Python by a derived class (e.g. GamsSet, GamsParameter, etc). Besides the type, a GamsSymbol has a name (this has to match the name inside the GAMS model), a dimension (currently up to 20 or gamsglobals.-maxdim) and some explanatory text.

Variables and equations also have a subtype: e.g. Binary, Positive, etc. for variables and e.g. E, G etc. for equations

GamsDatabases can be created empty, or initialized from existing GDX files or from another GamsDatabase (copy). Symbols can be added at any time (e.g. GamsDatabase.add_parameter), but once a symbol is part of a GamsDatabase, it cannot be removed. Only its associated data (GamsSymbolRecord) can be purged (see GamsSymbol.-clear()) or individually removed (GamsSymbol.delete_record). Individual data elements are accessed record by record. A record is identified by the keys (a vector of strings). The record data varies by symbol type. For example, a parameter record has a value property, a variable has the properties level, lower, upper, marginal, and scale. Adding a record with keys that already exist results in an exception. Similar, the unsuccessful search for a record also results in an exception.

GamsSymbol implements __iter__() and next() to conveniently iterate through the records of a symbol. There are also sliced access methods to symbol records that allow to iterate through all records with a fixed index at some positions. GamsDatabases can be exported as GDX files for permanent storage.

GamsJob.out_db and GamsModelInstance.sync_db provide instances of GamsDatabase to communicate results from a GAMS run or a solve. These databases should only be used in the context of the base object (GamsJob or GamsModelInstance). If a copy of such a database is required GamsWorkspace.add_database can be used to initialize a GamsDatabase from another database by specifying the optional parameter source_database (e.g. newdb = workspace.add_database(GamsJob.out_db)).

GamsDatabases often provide the input data for a GamsJob. Such GamsDatabases are listed in the GamsJob.-run method. Inside the GAMS model source the GamsDatabase is accessible through a GDX file. The GAMS model source requires a particular file name to connect to the proper GDX file (e.g. $GDXIN filename). A GamsDatabase can be created with a given name which can be then used inside the model (e.g. db = workspace.add-_database(database_name="SupplyData")) and then inside the GAMS model source: $GDXIN SupplyData) or an automatically generated name can be used. This name can be passed down to the GAMS model by using the defines dictionary of a GamsOptions instance:

```
db = workspace.add_database()
opt = workspace.add_options()
opt.defines["SupplyDataFileName"] = db.name
...
gamsjob.run(gams_options=opt, databases=db)
```

Inside the GAMS model source the name is accessed as follows:

```
$GDXIN %SupplyDataFileName%
```

One has to act with some caution when it comes to ordered sets which e.g. allow lag and lead. By not enforcing the "domain checking" for the GamsDatabase class we have aggravated the potential problems for ordered sets. For GAMS, the labels of set elements are just strings, so the order of a set is determined by the appearance of its elements. For example, if one has 'set k / 2,3,4,1,5 /', the order of k is exactly given by this sequence. So the lag (k-1) of k=4 is 3 and the lead (k+1) of k=4 is 1.

GAMS performs arithmetic with an extended number range. GAMS has special values for infinity (+INF, -INF), epsilon (EPS), not available (NA), and undefined (UNDEF). When GAMS evaluates expressions with these special values, the calculating engine ensures the correctness of the result (e.g. 5∗eps=eps or 5+eps=5). The GAMS model CRAZY in the GAMS Model Library documents the results of the arithmetic operations with respect to special values.

In the GAMS Python API we map the IEEE standard values for +/-infinity and NA to the corresponding GAMS values. The special value for UNDEF gets unfiltered through the GAMS Python API. The internal double value of UNDEF is 1.0E300 (or better use the constant sv_valund from gamsglobals).

Special attention needs to be given to the value of 0. Since GAMS is a sparse system it does not store (parameter) records with a true 0. If a record with numerical value of 0 is needed, EPS can help. For example:

```
set j /1*10 /; parameter b(j); b(j) = 1; b('5') = 0;
scalar s,c; s = sum(j, b(j)); c = card(b); display s,c;
```

will result in

```
----      3 PARAMETER s                    =        9.000
            PARAMETER c                    =        9.000
</code>
<p>but</p>
<code lang="GAMS">
b(j) = 1; b('5') = EPS;
</code>
<p>will result in</p>
<code lang="GAMS">
----      3 PARAMETER s                    =        9.000
            PARAMETER c                    =       10.000
```

What are the consequences for the GAMS Python API? If we read parameter b in case of b('5')=0, the GAMSDatabase will not have a record for b('5'). In case of b('5')=EPS, the GamsDatabase will have a record with value EPS. Unlike the IEEE values (e.g. float("inf")), arithmetic operations in Python will modify EPS (e.g. 5∗EPS==float("inf") but 5∗EPS!=EPS). The same rules apply for preparing input data for GAMS in a GamsDatabase. If a value of EPS is written, GAMS will see the special value EPS. All other small values (including 0) will be communicated unfiltered to GAMS. As mentioned before, zeros will not be entered as data records in GAMS. The compiler control $on/offEPS can help to automatically map zeros to EPS.

There is one oddity concerning values smaller than 1e-250 on GAMS input. Consider the following example:

```
b = db.add_parameter("b",1)
for i in range(1,11):
    b.add_record(str(i)).value = 1
b.find_record("5").value = 1E-251
job.run(db)
```

```
$load j b
scalar card_b; card_b = card(b); display card_b;
b(j) = 2*b(j); card_b = card(b); display card_b;
```

A record with values smaller than 1e-250 exists on input in GAMS, but as soon as the record gets updated by GAMS and is still smaller than 1e-250, the record gets removed.

The ordering of a set in GAMS can be non-intuitive: Consider "set i /5/, j /1∗5/;". Elements '5' gets internal number 1, '1' get 2, '2' gets 3 and so on. The last element of j '5' has already the internal number 1. The sequence of internal numbers in j is not ascending and hence GAMS considers set j as not sorted, i.e. one can't use the ord() function nor the lag or lead (-,–,+,++) operators. If 'j' would have been defined before 'i' in this example, the "set not ordered" problem would have been avoided.

Please note that the GamsDatabase actually does not implement a relational model for database management. It should be seen as a data storage or data container.

4.3.2 Member Function Documentation

4.3.2.1 def gams.database.GamsDatabase.add_equation (*self*, *identifier*, *dimension*, *equtype*, *explanatory_text* = " ")

Add equation symbol to database.

Parameters

identifier	Equation name
dimension	Equation dimension
equtype	Equation subtype (E: Equal, G: Greater, L: Less, N: No specification, X: External defined, C: Conic)
explanatory_text	Explanatory text of equation

See Also

add_parameter(), add_set(), add_variable()

4.3.2.2 def gams.database.GamsDatabase.add_parameter (*self, identifier, dimension, explanatory_text =* " ")

Add parameter symbol to database.

Parameters

identifier	Parameter name
dimension	Parameter dimension
explanatory_text	Explanatory text of parameter

See Also

add_equation(), add_set(), add_variable()

4.3.2.3 def gams.database.GamsDatabase.add_set (*self, identifier, dimension, explanatory_text =* " ")

Add set symbol to database.

Parameters

identifier	Set name
dimension	Set dimension
explanatory_text	Explanatory text of set

See Also

add_equation(), add_parameter(), add_variable()

4.3.2.4 def gams.database.GamsDatabase.add_variable (*self, identifier, dimension, vartype, explanatory_text =* " ")

Add variable symbol to database.

Parameters

identifier	Variable name
dimension	Variable dimension
vartype	Variable subtype (Binary, Integer, Positive, Negative, Free, SOS1, SOS2, SemiCont, SemiInt)
explanatory_text	Explanatory text to variable

See Also

add_equation(), add_parameter(), add_set()

4.3.2.5 def gams.database.GamsDatabase.export (*self, file_path =* `None`)

Write database into a GDX file.

Parameters

file_path	The path used to write the GDX file. A relative path is relative to the GAMS working directory. If not present, the file is written to the working directory using the name of the database.

4.3.2.6 def gams.database.GamsDatabase.get_equation (*self*, *equation_identifier*)

Get GamsEquation by name.

Parameters

equation_identifier	Name of the equation to retrieve

Returns

Reference to equation

See Also

get_symbol(), get_parameter(), get_set(), get_variable()

4.3.2.7 def gams.database.GamsDatabase.get_parameter (*self*, *parameter_identifier*)

Get GamsParameter by name.

Parameters

parameter_identifier	Name of the parameter to retrieve

Returns

Reference to parameter

See Also

get_symbol(), get_set(), get_variable(), get_equation()

4.3.2.8 def gams.database.GamsDatabase.get_set (*self*, *set_identifier*)

Get GamsSet by name.

Parameters

set_identifier	Name of the set to retrieve

Returns

Reference to set

See Also

get_symbol(), get_parameter(), get_variable(), get_equation()

4.3.2.9 def gams.database.GamsDatabase.get_symbol (*self*, *symbol_identifier*)

Get GamsSymbol by name.

```
symbol = database.get_symbol("a")
if isinstance(symbol, GamsParameter):
    print "symbol is a GamsParameter"
if isinstance(symbol, GamsSet):
    print "symbol is a GamsSet"
if isinstance(symbol, GamsVariable):
    print "symbol is a GamsVariable
if isinstance(symbol, GamsEquation):
    print "symbol is a GamsEquation
```

Parameters

symbol_identifier	Name of the symbol to retrieve

Returns

Reference to symbol

See Also

get_parameter(), get_set(). get_variable(), get_equation()

4.3.2.10 def gams.database.GamsDatabase.get_variable (*self, variable_identifier*)

Get GamsVariable by name.

Parameters

variable_identifier	Name of the variable to retrieve

Returns

Reference to variable

See Also

get_symbol(), get_parameter(), get_set(), get_equation()

4.3.3 Property Documentation

4.3.3.1 gams.database.GamsDatabase.number_symbols property(get_nr_symbols) [static]

Retrieve the number of symbols in the GamsDatabase.

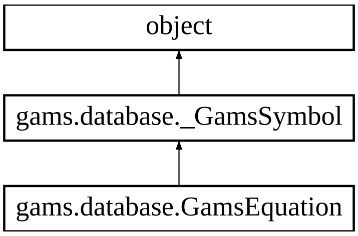
Note

This is the same as calling len(database)

4.4 gams.database.GamsEquation Class Reference

This is the representation of an equation symbol in GAMS.

Inheritance diagram for gams.database.GamsEquation:



Properties

- equtype property(get_equtype)

  Retrieve subtype of Equation (E: Equal, G: Greater, L: Less, N: No specification, X: External defined, C: Conic)

Additional Inherited Members

4.4.1 Detailed Description

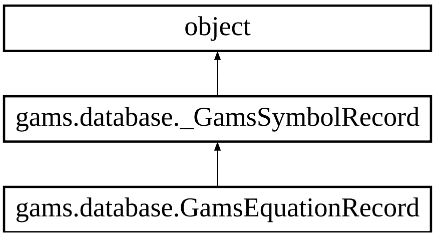
This is the representation of an equation symbol in GAMS.

It exists in a GamsDatabase and contains GamsEquationRecords which one can iterate through.

4.5 gams.database.GamsEquationRecord Class Reference

This is the representation of a single record of a GamsEquation.

Inheritance diagram for gams.database.GamsEquationRecord:


object
gams.database._GamsSymbolRecord
gams.database.GamsEquationRecord

Public Member Functions

- def __str__

  Retrieve a string representation of this record.

Properties

- level property(get_level, set_level)

  Get or set the level of this record.
- marginal property(get_marginal, set_marginal)

  Get or set the marginal of this record.
- upper property(get_upper, set_upper)

  Get or set the upper bound of this record.
- lower property(get_lower, set_lower)

  Get or set the lower bound of this record.

- scale property(get_scale, set_scale)

  Get or set the scale of this record.

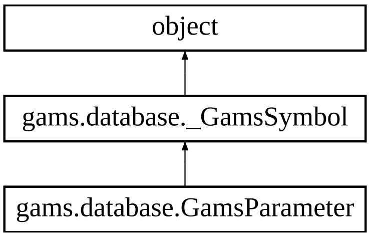
4.5.1 Detailed Description

This is the representation of a single record of a GamsEquation.

4.6 gams.database.GamsParameter Class Reference

This is the representation of a parameter symbol in GAMS.

Inheritance diagram for gams.database.GamsParameter:



Additional Inherited Members

4.6.1 Detailed Description

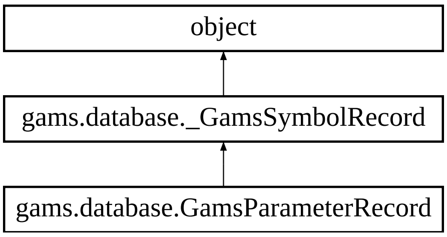
This is the representation of a parameter symbol in GAMS.

It exists in a GamsDatabase and contains GamsParameterRecords which one can iterate through.

4.7 gams.database.GamsParameterRecord Class Reference

This is the representation of a single record of a GamsParameter.

Inheritance diagram for gams.database.GamsParameterRecord:


object
gams.database._GamsSymbolRecord
gams.database.GamsParameterRecord

Public Member Functions

- def __str__

  Retrieve a string representation of this record.

Properties

- value property(get_value, set_value)

  Get or set the value of this record.

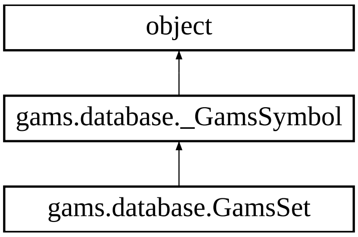
4.7.1 Detailed Description

This is the representation of a single record of a GamsParameter.

4.8 gams.database.GamsSet Class Reference

This is the representation of a set symbol in GAMS.

Inheritance diagram for gams.database.GamsSet:



Additional Inherited Members

4.8.1 Detailed Description

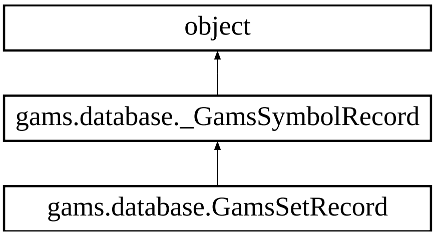
This is the representation of a set symbol in GAMS.

It exists in a GamsDatabase and contains GamsSetRecords which one can iterate through.

4.9 gams.database.GamsSetRecord Class Reference

This is the representation of a single record of a GamsSet.

Inheritance diagram for gams.database.GamsSetRecord:



Public Member Functions

- def __str__

  Retrieve a string representation of this record.

Properties

- text property(get_text, set_text)

  Get or set the explanatory text of this record.

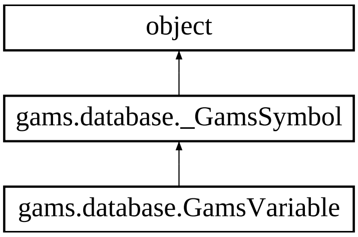
4.9.1 Detailed Description

This is the representation of a single record of a GamsSet.

4.10 gams.database.GamsVariable Class Reference

This is the representation of a variable symbol in GAMS.

Inheritance diagram for gams.database.GamsVariable:



Properties

- vartype property(get_vartype)

  Retrieve subtype of variable (Binary, Integer, Positive, Negative, Free, SOS1, SOS2, SemiCont, SemiInt)

Additional Inherited Members

4.10.1 Detailed Description

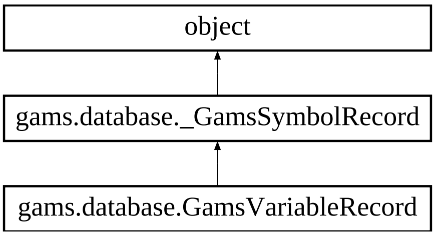
This is the representation of a variable symbol in GAMS.

It exists in a GamsDatabase and contains GamsVariableRecords which one can iterate through.

4.11 gams.database.GamsVariableRecord Class Reference

This is the representation of a single record of a GamsVariable.

Inheritance diagram for gams.database.GamsVariableRecord:


object
gams.database._GamsSymbolRecord
gams.database.GamsVariableRecord

Public Member Functions

- def __str__

    Retrieve a string representation of this record.

Properties

- level property(get_level, set_level)

    Get or set the level of this record.
- marginal property(get_marginal, set_marginal)

    Get or set the marginal of this record.
- upper property(get_upper, set_upper)

    Get or set the upper bound of this record.
- lower property(get_lower, set_lower)

    Get or set the lower bound of this record.
- scale property(get_scale, set_scale)

    Get or set the scale of this record.

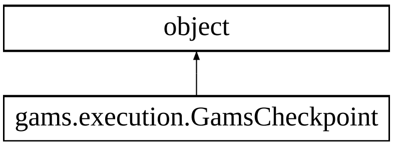
4.11.1 Detailed Description

This is the representation of a single record of a GamsVariable.

4.12 gams.execution.GamsCheckpoint Class Reference

A GamsCheckpoint class captures the state of a GamsJob after the GamsJob.run method has been carried out.

Inheritance diagram for gams.execution.GamsCheckpoint:



Public Member Functions

- def __init__

    Constructor.
- def add_modelinstance

    Create model instance.

Properties

- workspace property(get_workspace)

    Get the GamsWorkspace.
- name property(get_name)

    Get the checkpoint name.

4.12.1 Detailed Description

A GamsCheckpoint class captures the state of a GamsJob after the GamsJob.run method has been carried out.

Another GamsJob can continue (or restart) from a GamsCheckpoint. A GamsCheckpoint constructed with a file name will create a file (extension .g00) for permanent storage when supplied as parameter on the GamsJob.run method. Moreover, a GamsModelInstance is also initialized from a checkpoint that contains the model definition of the model instance.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 def gams.execution.GamsCheckpoint.__init__ (*self*, *workspace*, *checkpoint_name* = `None`)

Constructor.

Parameters

workspace	GamsWorkspace containing GamsCheckpoint
checkpoint_name	Identifier of GamsCheckpoint (determined automatically if omitted)

4.12.3 Member Function Documentation

4.12.3.1 def gams.execution.GamsCheckpoint.add_modelinstance (*self*, *modelinstance_name* = `None`)

Create model instance.

Parameters

modelinstance_name	Identifier of GamsModelInstance (determined automatically if omitted)

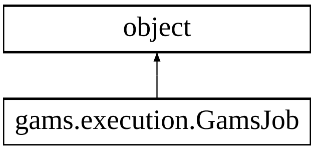
Returns

GamsModelInstance instance

4.13 gams.execution.GamsJob Class Reference

The GamsJob class manages the execution of a GAMS program given by GAMS model source.

Inheritance diagram for gams.execution.GamsJob:



Public Member Functions

- def __init__

  Constructor.
- def run

  Run GamsJob.

- def interrupt

  Send Interrupt to running Job.

Properties

- name property(get_job_name)

  Retrieve name of GamsJob.
- workspace property(get_workspace)

  Get GamsWorkspace containing GamsJob.
- out_db property(get_out_db)

  Get GamsDatabase created by run method.

4.13.1 Detailed Description

The GamsJob class manages the execution of a GAMS program given by GAMS model source.

The GAMS source (or more precisely the root of a model source tree) of the job can be provided as a string or by a filename (relative to the working directory of the GamsWorkspace) of a text file containing the GAMS model source. The run method organizes the export of the input GamsDatabases, calls the GAMS compiler and execution system with the supplied options and on successful completion provides through the property out_db (of type GamsDatabase) the results of the model run.

While the result data is captured in a GamsDatabase, the run method can also create a GamsCheckpoint that not only captures data but represents the state of the entire GamsJob and allows some other GamsJob to continue from this state. In case of a compilation or execution error, the run method will throw an exception. If the log output of GAMS is of interest, this can be captured by providing the output parameter of the run method (e.g. sys.stdout).

4.13.2 Constructor & Destructor Documentation

4.13.2.1 def gams.execution.GamsJob.__init__ (*self, ws, file_name* = `None`, *source* = `None`, *checkpoint* = `None`, *job_name* = `None`)

Constructor.

Note

It is not allowed to specify both file_name and source at the same time.

Parameters

ws	GamsWorkspace containing GamsJob
file_name	GAMS source file name
source	GAMS model as string
checkpoint	GamsCheckpoint to initialize GamsJob from
job_name	Job name (determined automatically if omitted)

4.13.3 Member Function Documentation

4.13.3.1 def gams.execution.GamsJob.interrupt (*self*)

Send Interrupt to running Job.

Returns

False if no process available, True otherwise

4.13.3.2 def gams.execution.GamsJob.run (*self*, *gams_options* = `None`, *checkpoint* = `None`, *output* = `None`, *create_out_db* = `True`, *databases* = `None`)

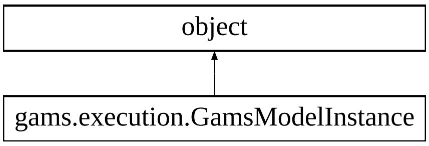
Run GamsJob.

Parameters

gams_options	GAMS options to control job
checkpoint	GamsCheckpoint to be created by GamsJob
output	Stream to capture GAMS log
create_out_db	Flag to define if out_db should be created
databases	Either a GamsDatabase or a list of GamsDatabases to be read by the GamsJob

4.14 gams.execution.GamsModelInstance Class Reference

Inheritance diagram for gams.execution.GamsModelInstance:



Public Member Functions

- def __init__

  Constructor.
- def __del__

  Use this to explicitly free unmanaged resources.
- def instantiate

  Instantiate the GamsModelInstance.
- def solve

  Solve model instance.

Properties

- model_status property(get_model_status)

  Status of the model.
- solver_status property(get_solver_status)

  Solve status of the model.
- checkpoint property(get_checkpoint)

  Retrieve GamsCheckpoint.
- name property(get_name)

  Retrieve name of GamsModelInstance.
- sync_db property(get_sync_db)

  Retrieve GamsDatabase used to synchronize modifiable data.

4.14.1 Detailed Description

The GamsJob class is the standard way of dealing with a GAMS model and the corresponding solution provided by a solver. The GAMS language provides programming flow that allows to solve models in a loop and do other sophisticated tasks, like building decomposition algorithms.

In rare cases, the GAMS model generation time dominates the solver solution time and GAMS itself becomes the bottleneck in an optimization application. For a model instance which is a single mathematical model generated by a GAMS solve statement, the GamsModelInstance class provides a controlled way of modifying a model instance and solving the resulting problem in the most efficient way, by communicating only the changes of the model to the solver and doing a hot start (in case of a continuous model like LP) without the use of disk IO.

The GamsModelInstance requires a GamsCheckpoint that contains the model definition. Significant parts of the GAMS solve need to be provided for the instantiation of the GamsModelInstance. The modification of the model instance is done through data in sync_db (a property of GamsModelInstance of type GamsDatabase). One needs to create GamsModifiers which contain the information on how to modify the GamsModelInstance. Such a GamsModifier consists either of a GamsParameter or of a triple with the GamsVariable or GamsEquation to be updated, the modification action (e.g. Upper, Lower or Fixed for updating bounds of a variable, or Primal/Dual for updating the level/marginal of a variable or equation mainly used for starting non-linear models from different starting points), and a GamsParameter that holds the data for modification. GamsSymbols of a GamsModifier must belong to sync_db. The list of GamsModifiers needs to be supplied on the instantiate call. The use of GamsParameters that are GamsModifiers is restricted in the GAMS model source. For example, the parameter cannot be used inside $(). Such parameters become endogenous to the model and will be treated by the GAMS compiler as such. Moreover, the rim of the model instance is fixed: No addition of variables and equations is possible.

The instantiate call will only query the symbol information of the GamsModifiers, not the data of sync_db, e.g. to retrieve the dimension of the modifiers. That's why the modifier symbols have to exist (but don't have to have data) in sync_db when instantiate is called. The GamsParameters that contain the update data in sync_db can be filled at any time before executing the solve method. The solve method uses this data to update the model instance. The solve method will iterate through all records of modifier symbols in the model instance and try to find update data in sync_db. If a record in sync_db is found, this data record will be copied into the model instance. If no corresponding record is found in SyncDB there are different choices: 1) the original data record is restored (update_type=SymbolUpdateType.BaseCase) which is the default, 2) the default record of a GamsParameter (which is 0) is used (update_type=SymbolUpdateType.Zero, and 3) no copy takes place and we use the previously copied record value (update_type=SymbolUpdateType.Accumulate). After the model instance has been updated, the model is passed to the selected solver.

After the completion of the Solve method, the sync_db will contain the primal and dual solution of the model just solved. Moreover, the GamsParameters that are GamsModifiers are also accessible in sync_db as GamsVariables with the name of the GamsParameter plus "_var". The Marginal of this GamsVariable can provide sensitivity information about the parameter setting. The status of the solve is accessible through the model_status and solver_status properties.

In general, file operations in GAMS Python API take place in the working_directory defined in the GamsWorkspace. Execptions to this rule are files read or created due to solver specific options in the solve routine of GamsModelInstance. These files are written to (or read from) the current directory, meaning the directory where the application gets executed.

Example on how to create a GAMSModelInstance from a GAMSCheckpoint that was generated by the Run method of GAMSJob.

```
ws = GamsWorkspace()
cp = ws.add_checkpoint()

ws.gamslib("trnsport")

job = ws.add_job_from_file("trnsport.gms")
job.run(checkpoint=cp)

mi = cp.add_modelinstance()
b = mi.sync_db.add_parameter("b", 1, "demand")

mi.instantiate("transport us lp min z", GamsModifier(b))

bmult = [ 0.7, 0.9, 1.1, 1.3 ]
for bm in bmult:
    b.clear()
    for rec in job.out_db.get_parameter("b"):
        b.add_record(rec.keys).value = rec.value * bm
    mi.solve()
    print "Scenario bmult=" + str(bm) + ":"
    print "  Modelstatus: " + str(mi.model_status)
    print "  Solvestatus: " + str(mi.solver_status)
    print "  Obj: " + str(mi.sync_db.get_variable("z")[()].level)
```

4.14.2 Constructor & Destructor Documentation

4.14.2.1 def gams.execution.GamsModelInstance.__init__ (*self*, *checkpoint*, *modelinstance_name* = `None`)

Constructor.

Parameters

checkpoint	GamsCheckpoint
modelinstance_name	Identifier of GamsModelInstance (determined automatically if omitted)

4.14.3 Member Function Documentation

4.14.3.1 def gams.execution.GamsModelInstance.instantiate (*self*, *model_definition*, *modifiers* = `[]`, *options* = `None`)

Instantiate the GamsModelInstance.

Parameters

model_definition	Model definition
modifiers	List of GamsModifiers
options	GamsOptions

4.14.3.2 def gams.execution.GamsModelInstance.solve (*self*, *update_type* = SymbolUpdateType.BaseCase, *output* = `None`, *mi_opt* = `None`)

Solve model instance.

Parameters

update_type	Update type
output	Used to capture GAMS log, (e.g. sys.stdout)
mi_opt	GamsModelInstance options

4.14.4 Property Documentation

4.14.4.1 gams.execution.GamsModelInstance.model_status property(get_model_status) `[static]`

Status of the model.

(available after a solve)

4.14.4.2 gams.execution.GamsModelInstance.solver_status property(get_solver_status) `[static]`

Solve status of the model.

(available after a solve)

4.15 gams.execution.GamsModelInstanceOpt Class Reference

The GamsModelInstanceOpt can be used to customize the GamsModelInstance.solve() routine.

Public Member Functions

- def __init__

  Constructor.

Public Attributes

- solver

  GAMS Solver.
- opt_file

  GAMS Optionfile number.
- no_match_limit

  Controls the maximum number of accepted unmatched scenario records before terminating the solve.
- debug

  Debug Flag.

4.15.1 Detailed Description

The GamsModelInstanceOpt can be used to customize the GamsModelInstance.solve() routine.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 def gams.execution.GamsModelInstanceOpt.__init__ (*self*, *solver* = `None`, *opt_file* = `-1`, *no_match_limit* = `0`, *debug* = `False`)

Constructor.

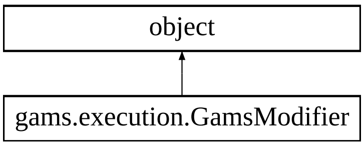
Parameters

solver	GAMS Solver
no_match_limit	Controls the maximum number of accepted unmatched scenario records before terminating the solve
debug	Debug Flag

4.16 gams.execution.GamsModifier Class Reference

Instances of this class are input to GamsModelInstance.instatiate()

Inheritance diagram for gams.execution.GamsModifier:



Public Member Functions

- def __init__

  Constructor.

Properties

- gams_symbol property(get_gams_symbol)

  Symbol in the GAMS model to be modified.
- update_action property(get_update_action)

  Type of modification.
- data_symbol property(get_data_symbol)

  Symbol containing the data for the modification.

4.16.1 Detailed Description

Instances of this class are input to GamsModelInstance.instatiate()

A GamsModifier consists either of a GamsParameter or a triple: A GamsVariable or GamsEquation to be modified, the modification action (e.g. . Upper, Lower or Fixed for updating bounds of a variable, or Primal/Dual for updating the level/marginal of a variable or equation mainly used for starting non-linear models from different starting points), and a GamsParameter that holds the data for modification.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 def gams.execution.GamsModifier.__init__ (*self*, *gams_symbol*, *update_action* = None, *data_symbol* = None)

Constructor.

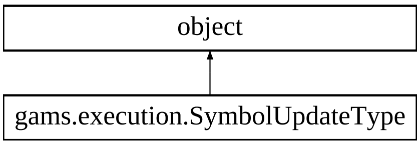
Parameters

gams_symbol	Symbol in the GAMS model to be modified. This can be a GamsParameter, GamsVariable or GamsEquation. If a variable or an equation is specified a data_symbol and an update_action have to be specified as well.
update_action	Modification action
data_symbol	Parameter containing the data for the modification

4.17 gams.execution.SymbolUpdateType Class Reference

Symbol update type.

Inheritance diagram for gams.execution.SymbolUpdateType:



Static Public Attributes

- int Zero 0

  If record does not exist use 0 (Zero)

- int BaseCase 1

  If record does not exist use values from instantiation.
- int Accumulate 2

  If record does not exist use value from previous solve.

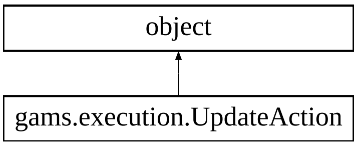
4.17.1 Detailed Description

Symbol update type.

4.18 gams.execution.UpdateAction Class Reference

What field to update.

Inheritance diagram for gams.execution.UpdateAction:



Static Public Attributes

- int Upper 1

  Supplies upper bounds for a variable.
- int Lower 2

  Supplies lower bounds for a variable.
- int Fixed 3

  Supplies fixed bounds for a variable.
- int Primal 4

  Supplies level for a variable or equation.
- int Dual 5

  Supplies marginal for a variable or equation.

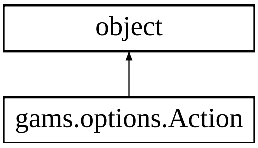
4.18.1 Detailed Description

What field to update.

4.19 gams.options.Action Class Reference

Gams processing requests.

Inheritance diagram for gams.options.Action:



Static Public Attributes

- string RestartAfterSolve "R"

  Restart After Solve.
- string CompileOnly "C"

  CompileOnly.
- string ExecuteOnly "E"

  ExecuteOnly.
- string CompileAndExecute "CE"

  Compile and Execute.
- string GlueCodeGeneration "G"

  Glue Code Generation.
- string TraceReport "GT"

  Trace Report.

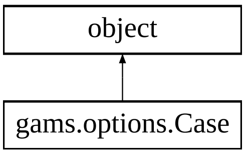
4.19.1 Detailed Description

Gams processing requests.

4.20 gams.options.Case Class Reference

Output case option.

Inheritance diagram for gams.options.Case:



Static Public Attributes

- int MixedCase 0

  write listing file in mixed case
- int UpperCase 1

  write listing file in upper case only

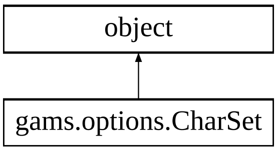
4.20.1 Detailed Description

Output case option.

4.21 gams.options.CharSet Class Reference

Character set flag.

Inheritance diagram for gams.options.CharSet:



Static Public Attributes

- int LimitedGAMSCharSet 0

  use limited GAMS characters set
- int AnyChar 1

  accept any character in comments and text items (foreign language characters)

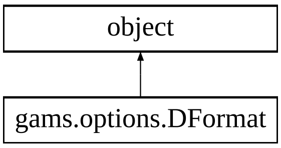
4.21.1 Detailed Description

Character set flag.

4.22 gams.options.DFormat Class Reference

Date format.

Inheritance diagram for gams.options.DFormat:



Static Public Attributes

- int Slash 0

  date as mm/dd/yy
- int Dot 1

  date as dd.mm.yy
- int Dash 2

  date as yy-mm-dy

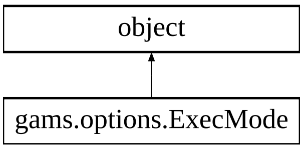
4.22.1 Detailed Description

Date format.

4.23 gams.options.ExecMode Class Reference

Limits on external programs that are allowed to be executed.

Inheritance diagram for gams.options.ExecMode:



Static Public Attributes

- int EverythingAllowed 0

  everything allowed
- int InteractiveShellsProhibited 1

  interactive shells in $call and execute commands are prohibited
- int CallAndExecuteProhibited 2

  all $call and execute commands are prohibited
- int EchoAndPutOnlyToWorkdir 3

  $echo or put commands can only write to directories in or below the working or scratchdir
- int EchoAndPutProhibited 4

  $echo and put commands are not allowed

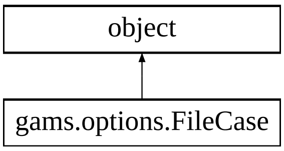
4.23.1 Detailed Description

Limits on external programs that are allowed to be executed.

4.24 gams.options.FileCase Class Reference

Casing of new file names (put, gdx, ref etc.)

Inheritance diagram for gams.options.FileCase:



Static Public Attributes

- int DefaultCase 0

  causes GAMS to use default casing
- int UpperCase 1

  causes GAMS to uppercase filenames
- int LowerCase 2

  causes GAMS to lowercase filenames

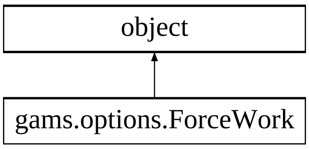
4.24.1 Detailed Description

Casing of new file names (put, gdx, ref etc.)

4.25 gams.options.ForceWork Class Reference

Force newer GAMS systems to translate and read save files generated by older systems.

Inheritance diagram for gams.options.ForceWork:



Static Public Attributes

- int NoTranslation 0

  no translation
- int TryTranslation 1

  try translation

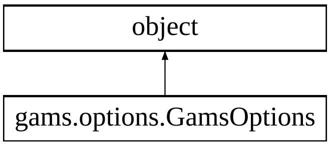
4.25.1 Detailed Description

Force newer GAMS systems to translate and read save files generated by older systems.

4.26 gams.options.GamsOptions Class Reference

The GamsOptions class manages GAMS options (sometimes also called GAMS parameters since they correspond to the command line parameters of the GAMS executable) for a GamsJob and GamsModelInstance.

Inheritance diagram for gams.options.GamsOptions:



Properties

- all_model_types property(fset=set_all_model_types)

  Set solver for all model types.
- lp property(get_lp, set_lp)

  Default lp solver.
- mip property(get_mip, set_mip)

  Default mip solver.
- rmip property(get_rmip, set_rmip)

  Default rmip solver.
- nlp property(get_nlp, set_nlp)

  Default nlp solver.
- mcp property(get_mcp, set_mcp)

  Default mcp solver.
- mpec property(get_mpec, set_mpec)

Default mpec solver.

- rmpec property(get_rmpec, set_rmpec)

  Default rmpec solver.
- cns property(get_cns, set_cns)

  Default cns solver.
- dnlp property(get_dnlp, set_dnlp)

  Default dnlp solver.
- rminlp property(get_rminlp, set_rminlp)

  Default rminlp solver.
- minlp property(get_minlp, set_minlp)

  Default minlp solver.
- qcp property(get_qcp, set_qcp)

  Default qcp solver.
- miqcp property(get_miqcp, set_miqcp)

  Default miqcp solver.
- rmiqcp property(get_rmiqcp, set_rmiqcp)

  Default rmiqcp solver.
- emp property(get_emp, set_emp)

  Default emp solver.
- action property(get_action, set_action)

  Gams processing requests.
- putdir property(get_putdir, set_putdir)

  Put file directory.
- griddir property(get_griddir, set_griddir)

  Grid file directory.
- stepsum property(get_stepsum, set_stepsum)

  Summary of computing resources used by job steps.
- dformat property(get_dformat, set_dformat)

  Date format.
- tformat property(get_tformat, set_tformat)

  Time format.
- tabin property(get_tabin, set_tabin)

  Tab spacing.
- case property(get_case, set_case)

  Output case option.
- license property(get_license, set_license)

  Use alternative license file.
- profile property(get_profile, set_profile)

  Execution profiling.
- libincdir property(get_libincdir, set_libincdir)

  LibInclude directory.
- sysincdir property(get_sysincdir, set_sysincdir)

  SysInclude directory.
- user1 property(get_user1, set_user1)

  User string N.
- user2 property(get_user2, set_user2)

  User string N.
- user3 property(get_user3, set_user3)

  User string N.
- user4 property(get_user4, set_user4)

  User string N.

- user5 property(get_user5, set_user5)

  User string N.
- forcework property(get_forcework, set_forcework)

  Force newer GAMS systems to translate and read save files generated by older systems.
- cerr property(get_cerr, set_cerr)

  Compile time error limit.
- optfile property(get_optfile, set_optfile)

  Default option file.
- opt property(get_opt, set_opt)

  Fold constant $ expressions.
- stringchk property(get_stringchk, set_stringchk)

  String substitution options.
- optdir property(get_optdir, set_optdir)

  Option file directory.
- execerr property(get_execerr, set_execerr)

  Execution time error limit.
- charset property(get_charset, set_charset)

  Character set flag.
- keep property(get_keep, set_keep)

  Do not delete scratch files.
- iterlim property(get_iterlim, set_iterlim)

  Iteration limit solver default.
- domlim property(get_domlim, set_domlim)

  Domain violation limit solver default.
- reslim property(get_reslim, set_reslim)

  Resource (CPU) solver default limit.
- optcr property(get_optcr, set_optcr)

  Relative Optimality criterion solver default.
- optca property(get_optca, set_optca)

  Absolute Optimality criterion solver default.
- sysout property(get_sysout, set_sysout)

  Solver Status file reporting option.
- solprint property(get_solprint, set_solprint)

  Solution report print option.
- bratio property(get_bratio, set_bratio)

  Basis acceptance threshold.
- holdfixed property(get_holdfixed, set_holdfixed)

  Treat fixed variables as constants.
- nodlim property(get_nodlim, set_nodlim)

  Node limit in branch and bound tree.
- workfactor property(get_workfactor, set_workfactor)

  Work space multiplier for some solvers.
- workspace property(get_workspace, set_workspace)

  Work space for some solvers in MB.
- forlim property(get_forlim, set_forlim)

  Gams looping limit.
- seed property(get_seed, set_seed)

  Random number seed.
- execmode property(get_execmode, set_execmode)

  Limits on external programs that are allowed to be executed.
- interactivesolver property(get_interactivesolver, set_interactivesolver)

Allow solver to interact via command line.

- plicense property(get_plicense, set_plicense)

  Privacy license file name.
- warnings property(get_warnings, set_warnings)

  Number of warnings permitted before a run terminates.
- timer property(get_timer, set_timer)

  Instruction timer threshold in milli seconds.
- filecase property(get_filecase, set_filecase)

  Casing of new file names (put, gdx, ref etc.)
- savepoint property(get_savepoint, set_savepoint)

  Save solver point in GDX file.
- solvelink property(get_solvelink, set_solvelink)

  Solver link options 0 save 1 script 2 module.
- sys11 property(get_sys11, set_sys11)

  Dynamic resorting if indices in assignment/data statements are not in natural order.
- sys12 property(get_sys12, set_sys12)

  Pass model with generation errors to solver.
- zerores property(get_zerores, set_zerores)

  The results of certain operations will be set to zero if abs(result) .le.
- zeroresrep property(get_zeroresrep, set_zeroresrep)

  Report underflow as a warning when abs(results) .le.
- heaplimit property(get_heaplimit, set_heaplimit)

  Maximum Heap size allowed in MB.
- gdxcompress property(get_gdxcompress, set_gdxcompress)

  Compression of generated gdx file.
- gdxconvert property(get_gdxconvert, set_gdxconvert)

  Version of gdx files generated (for backward compatibility)
- etlim property(get_etlim, set_etlim)

  Elapsed time limit in seconds.
- profiletol property(get_profiletol, set_profiletol)

  Minimum time a statement must use to appear in profile generated output.
- profilefile property(get_profilefile, set_profilefile)

  Write profile information to this file.
- maxprocdir property(get_maxprocdir, set_maxprocdir)

  Maximum number of 225∗ process directories.
- integer1 property(get_integer1, set_integer1)

  Integer communication cell N.
- integer2 property(get_integer2, set_integer2)

  Integer communication cell N.
- integer3 property(get_integer3, set_integer3)

  Integer communication cell N.
- integer4 property(get_integer4, set_integer4)

  Integer communication cell N.
- integer5 property(get_integer5, set_integer5)

  Integer communication cell N.
- gridscript property(get_gridscript, set_gridscript)

  Grid submission script.
- threads property(get_threads, set_threads)

  Number of threads to be used by a solver.

4.26.1 Detailed Description

The GamsOptions class manages GAMS options (sometimes also called GAMS parameters since they correspond to the command line parameters of the GAMS executable) for a GamsJob and GamsModelInstance.

There are integer (e.g. nodlim), double (e.g. reslim), and string (e.g. putdir) valued options. There are also a few list options (defines to set string macros inside GAMS and idir provide multiple search paths for include files) and a power option to set a solver for all suitable model types (all_model_types).

Some options known from other interfaces to GAMS that are of limited use or could even create problematic situations in the Python environment are not settable through the GamsOptions class.

For some options (e.g. case) other GAMS interfaces use numeric values (e.g. 0,1) while the GamsOptions class has enumerated types with proper names (e.g. MixedCase, UpperCase).

4.26.2 Property Documentation

4.26.2.1 gams.options.GamsOptions.zerores property(get_zerores, set_zerores) [static]

The results of certain operations will be set to zero if abs(result) .le.

ZeroRes

4.26.2.2 gams.options.GamsOptions.zeroresrep property(get_zeroresrep, set_zeroresrep) [static]

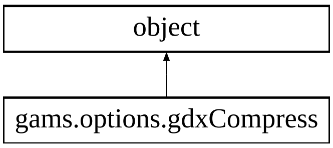
Report underflow as a warning when abs(results) .le.

ZeroRes and result set to zero

4.27 gams.options.gdxCompress Class Reference

Compression of generated gdx file.

Inheritance diagram for gams.options.gdxCompress:



Static Public Attributes

- int DoNotCompressGDX 0

  do not compress gdx files
- int CompressGDX 1

  compress gdx files

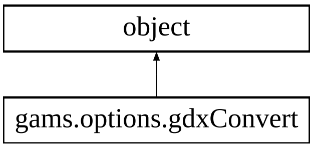
4.27.1 Detailed Description

Compression of generated gdx file.

4.28 gams.options.gdxConvert Class Reference

Version of gdx files generated (for backward compatibility)

Inheritance diagram for gams.options.gdxConvert:



Static Public Attributes

- string Version5 "v5"

  version 5 gdx file, does not support compression
- string Version6 "v6"

  version 6 gdx file
- string Version7 "v7"

  version 7 gdx file

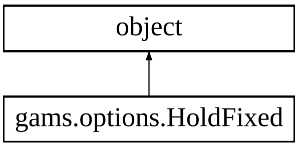
4.28.1 Detailed Description

Version of gdx files generated (for backward compatibility)

4.29 gams.options.HoldFixed Class Reference

Treat fixed variables as constants.

Inheritance diagram for gams.options.HoldFixed:



Static Public Attributes

- int FixedVarsNotTreatedAsConstants 0

  fixed variables are not treated as constants
- int FixedVarsTreatedAsConstants 1

  fixed variables are treated as constants

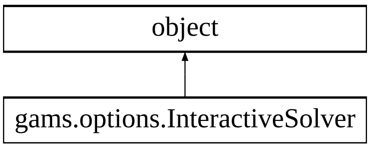
4.29.1 Detailed Description

Treat fixed variables as constants.

4.30 gams.options.InteractiveSolver Class Reference

Allow solver to interact via command line.

Inheritance diagram for gams.options.InteractiveSolver:



Static Public Attributes

- int NoInteraction 0

  Interaction with solvelink 0 is not supported.
- int AllowInteraction 1

  Interaction with solvelink 0 is supported.

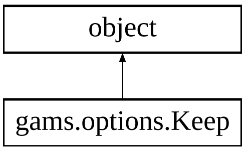
4.30.1 Detailed Description

Allow solver to interact via command line.

4.31 gams.options.Keep Class Reference

Do not delete scratch files.

Inheritance diagram for gams.options.Keep:



Static Public Attributes

- int DeleteAllFiles 0

  delete all files
- int KeepAllTempFiles 1

  keep all temporary files

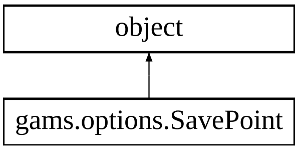
4.31.1 Detailed Description

Do not delete scratch files.

4.32 gams.options.SavePoint Class Reference

Save solver point in GDX file.

Inheritance diagram for gams.options.SavePoint:



Static Public Attributes

- int NoPointFile 0

  no point gdx file is to be saved
- int LastSolvePointFile 1

  a point gdx file from the last solve is to be saved
- int EverySolvePointFile 2

  a point gdx file from every solve is to be saved

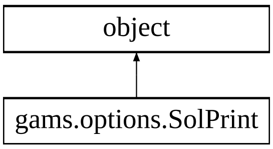
4.32.1 Detailed Description

Save solver point in GDX file.

4.33 gams.options.SolPrint Class Reference

Solution report print option.

Inheritance diagram for gams.options.SolPrint:



Static Public Attributes

- int RemoveSolLstFollowingSolves 0

  remove solution listings following solves
- int IncludeSolLstFollowingSolves 1

  include solution listings following solves
- int SuppressAllSolInfo 2

  suppress all solution information

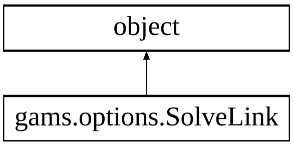
4.33.1 Detailed Description

Solution report print option.

4.34 gams.options.SolveLink Class Reference

Solver link options 0 save 1 script 2 module.

Inheritance diagram for gams.options.SolveLink:



Static Public Attributes

- int ChainScript 0

  GAMS operates as it has for years.
- int CallScript 1

  solver is called from a shell and GAMS remains open
- int CallModule 2

  solver is called with a spawn (if possible) or a shell (if spawn is not possible) and GAMS remains open
- int AsyncGrid 3

  GAMS starts the solution and continues in a Grid computing environment.
- int AsyncSimulate 4

  GAMS starts the solution and wait (same submission process as 3) in a Grid computing environment.
- int LoadLibrary 5

  the problem is passed to the solver in core without use of temporary files

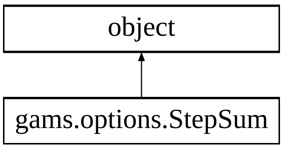
4.34.1 Detailed Description

Solver link options 0 save 1 script 2 module.

4.35 gams.options.StepSum Class Reference

Summary of computing resources used by job steps.

Inheritance diagram for gams.options.StepSum:



Static Public Attributes

- int NoStepSummmary 0

  no step summmary
- int StepSummary 1

  step summary printed

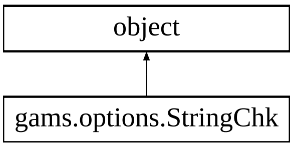
4.35.1 Detailed Description

Summary of computing resources used by job steps.

4.36 gams.options.StringChk Class Reference

String substitution options.

Inheritance diagram for gams.options.StringChk:



Static Public Attributes

- int NoError 0

  no substitution if symbol undefined and no error
- int Error 1

  error if symbol undefined
- int NoErrorRemoveSymbol 2

  remove entire symbol reference if undefined and no error

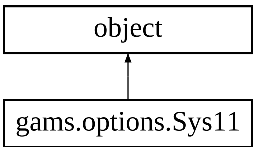
4.36.1 Detailed Description

String substitution options.

4.37 gams.options.Sys11 Class Reference

Dynamic resorting if indices in assignment/data statements are not in natural order.

Inheritance diagram for gams.options.Sys11:



Static Public Attributes

- int AutomaticOptimization 0

  Automatic optimization/restructuring of data.
- int NoOptimization 1

  no optimization
- int AlwaysOptimize 2

  always optimize/restructure

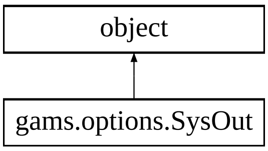
4.37.1 Detailed Description

Dynamic resorting if indices in assignment/data statements are not in natural order.

4.38 gams.options.SysOut Class Reference

Solver Status file reporting option.

Inheritance diagram for gams.options.SysOut:



Static Public Attributes

- int SuppressAdditionalSolverOutput 0

  suppress additional solver generated output
- int IncludeAdditionalSolverOutput 1

  include additional solver generated output

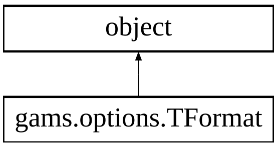
4.38.1 Detailed Description

Solver Status file reporting option.

4.39 gams.options.TFormat Class Reference

Time format.

Inheritance diagram for gams.options.TFormat:



Static Public Attributes

- int Colon 0

  time as hh:mm:ss
- int Dot 1

  time as hh.mm.ss

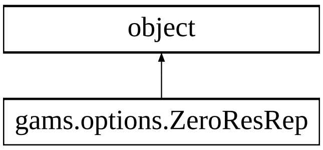
4.39.1 Detailed Description

Time format.

4.40 gams.options.ZeroResRep Class Reference

Report underflow as a warning when abs(results) .le.

Inheritance diagram for gams.options.ZeroResRep:



Static Public Attributes

- int NoWarning 0

  no warning when a rounding occurs because of ZeroRes
- int IssueWarning 1

  issue warnings whenever a rounding occurs because of ZeroRes

4.40.1 Detailed Description

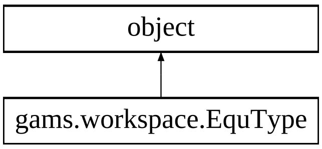
Report underflow as a warning when abs(results) .le.

ZeroRes and result set to zero

4.41 gams.workspace.EquType Class Reference

Equation subtype.

Inheritance diagram for gams.workspace.EquType:



Static Public Attributes

- int E 0

  Equality - =E=.
- int G 1

  Greater or equal than inequality - =G=.
- int L 2

  Less or equal than inequality - =L=.
- int N 3

  Non-binding equation - =N=.
- int X 4

  External equation - =X=.
- int C 5

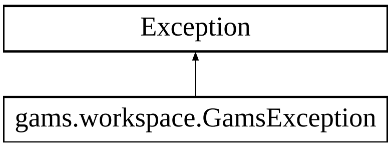
  Cone equation - =C=.

4.41.1 Detailed Description

Equation subtype.

4.42 gams.workspace.GamsException Class Reference

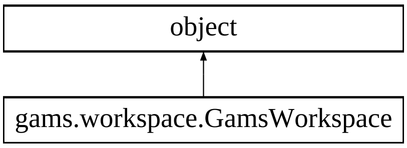
Inheritance diagram for gams.workspace.GamsException:



4.43 gams.workspace.GamsWorkspace Class Reference

The GamsWorkspace is the base class of the gams namespace.

Inheritance diagram for gams.workspace.GamsWorkspace:



Public Member Functions

- def __init__

  constructor
- def gamslib

  Retrieves model from GAMS Model Library.
- def testlib

  Retrieves model from GAMS Test Library.
- def emplib

  Retrieves model from Extended Math Programming Library.
- def datalib

  Retrieves model from GAMS Data Utilities Library.
- def add_database

  Database creation.
- def add_database_from_gdx

  Database creation from an existing GDX file.
- def add_job_from_string

  Create GamsJob from string model source.
- def add_job_from_file

  Create GamsJob from model file.
- def add_options

  Create GamsOptions.
- def add_checkpoint

  Create GamsCheckpoint.

Public Attributes

- scratch_file_prefix

  A string used to prefix automatically generated files.

Properties

- working_directory property(get_working_directory)

  GAMS working directory, anchor for all file-based operations.
- system_directory property(get_system_directory)

  GAMS system directory.

4.43.1 Detailed Description

The GamsWorkspace is the base class of the gams namespace.

Most objects of the gams namespace (e.g. GamsDatabase and GamsJob) should be created by an "add" method of GamsWorkspace instead of using the constructors.

Unless a GAMS system directory is specified during construction of GamsWorkspace, GamsWorkspace determines the location of the GAMS installation automatically. This is a source of potential problems if more than one GAMS installation exist on the machine.

Furthermore, a working directory (the anchor into the file system) can be provided when constructing the GamsWorkspace instance. All file based operation inside a GAMS model should be relative to this location (e.g. $GDXIN and $include). There are options to add input search paths (e.g. IDir) and output path (e.g. PutDir) to specify other file system locations. If no working directory is supplied, GamsWorkspace creates a temporary folder and on instance destruction removes this temporary folder.

In a typical Python application a single instance of GamsWorkspace will suffice, since the class is thread-safe.

4.43.2 Constructor & Destructor Documentation

4.43.2.1 def gams.workspace.GamsWorkspace.__init__ (*self*, *working_directory* = `None`, *system_directory* = `None`, *debug* = `False`)

constructor

Parameters

working_directory	GAMS working directory, anchor for all file-based operations (determined automatically if omitted, in user's temporary folder)
system_directory	GAMS system directory (determined automatically if omitted)
debug	Debug Flag

4.43.3 Member Function Documentation

4.43.3.1 def gams.workspace.GamsWorkspace.add_checkpoint (*self*, *checkpoint_name* = `None`)

Create GamsCheckpoint.

Parameters

checkpoint_name	checkpoint_name Identifier of GamsCheckpoint or filename for existing checkpoint (determined automatically if omitted)

Returns

GamsCheckpoint instance

4.43.3.2 def gams.workspace.GamsWorkspace.add_database (*self*, *database_name* = None, *source_database* = None)

Database creation.

Parameters

database_name	Identifier of GamsDatabase (determined automatically if omitted)
source_database	Source GamsDatabase to initialize Database from (empty Database if omitted)

Returns

Instance of type GamsDatabase

4.43.3.3 def gams.workspace.GamsWorkspace.add_database_from_gdx (*self*, *gdx_file_name*, *database_name* = None)

Database creation from an existing GDX file.

Parameters

gdx_file_name	GDX File to initialize Database from
database_name	Identifier of GamsDatabase (determined automatically if omitted)

Returns

Instance of type GamsDatabase

4.43.3.4 def gams.workspace.GamsWorkspace.add_job_from_file (*self*, *file_name*, *checkpoint* = None, *job_name* = None)

Create GamsJob from model file.

Parameters

file_name	GAMS source file name
checkpoint	GamsCheckpoint to initialize GamsJob from
job_name	Job name (determined automatically if omitted)

Returns

GamsJob instance

4.43.3.5 def gams.workspace.GamsWorkspace.add_job_from_string (*self*, *gams_source*, *checkpoint* = None, *job_name* = None)

Create GamsJob from string model source.

Parameters

gams_source	GAMS model as string
checkpoint	GamsCheckpoint to initialize GamsJob from
job_name	Job name (determined automatically if omitted)

Returns

GamsJob instance

4.43.3.6 def gams.workspace.GamsWorkspace.add_options (*self*, *gams_options_from* = `None`)

Create GamsOptions.

Returns

GamsOptions instance

4.43.3.7 def gams.workspace.GamsWorkspace.datalib (*self*, *model*)

Retrieves model from GAMS Data Utilities Library.

Parameters

model	Model name

4.43.3.8 def gams.workspace.GamsWorkspace.emplib (*self*, *model*)

Retrieves model from Extended Math Programming Library.

Parameters

model	Model name

4.43.3.9 def gams.workspace.GamsWorkspace.gamslib (*self*, *model*)

Retrieves model from GAMS Model Library.

Parameters

model	Model name

4.43.3.10 def gams.workspace.GamsWorkspace.testlib (*self*, *model*)

Retrieves model from GAMS Test Library.

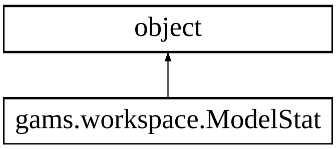
Parameters

model	Model name

4.44 gams.workspace.ModelStat Class Reference

Model Solution Status.

Inheritance diagram for gams.workspace.ModelStat:



Static Public Attributes

- int OptimalGlobal 1

  Optimal solution achieved.
- int OptimalLocal 2

  Local optimal solution achieved.
- int Unbounded 3

  Unbounded model found.
- int InfeasibleGlobal 4

  Infeasible model found.
- int InfeasibleLocal 5

  Locally infeasible model found.
- int InfeasibleIntermed 6

  Solver terminated early and model was still infeasible.
- int NonOptimalIntermed 7

  Solver terminated early and model was feasible but not yet optimal.
- int Integer 8

  Integer solution found.
- int NonIntegerIntermed 9

  Solver terminated early with a non integer solution found.
- int IntegerInfeasible 10

  No feasible integer solution could be found.
- int LicenseError 11

  Licensing problem.
- int ErrorUnknown 12

  Error - No cause known.
- int ErrorNoSolution 13

  Error - No solution attained.
- int NoSolutionReturned 14

  No solution returned.
- int SolvedUnique 15

  Unique solution in a CNS models.
- int Solved 16

  Feasible solution in a CNS models.
- int SolvedSingular 17

  Singular in a CNS models.
- int UnboundedNoSolution 18

  Unbounded - no solution.
- int InfeasibleNoSolution 19

  Infeasible - no solution.

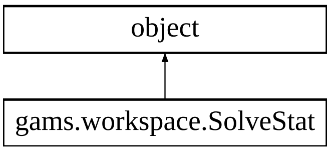
4.44.1 Detailed Description

Model Solution Status.

4.45 gams.workspace.SolveStat Class Reference

Solver termination condition.

Inheritance diagram for gams.workspace.SolveStat:



Static Public Attributes

- int Normal 1

  Normal termination.
- int Iteration 2

  Solver ran out of iterations.
- int Resource 3

  Solver exceeded time limit.
- int Solver 4

  Solver quit with a problem.
- int EvalError 5

  Solver quit with nonlinear term evaluation errors.
- int Capability 6

  Solver terminated because the model is beyond the solvers capabilities.
- int License 7

  Solver terminated with a license error.
- int User 8

  Solver terminated on users request (e.g.
- int SetupErr 9

  Solver terminated on setup error.
- int SolverErr 10

  Solver terminated with error.
- int InternalErr 11

  Solver terminated with error.
- int Skipped 12

  Solve skipped.
- int SystemErr 13

  Other error.

4.45.1 Detailed Description

Solver termination condition.

4.45.2 Member Data Documentation

4.45.2.1 int gams.workspace.SolveStat.User 8 [static]

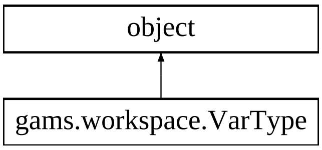
Solver terminated on users request (e.g.

Ctrl-C)

4.46 gams.workspace.VarType Class Reference

Variable subtype.

Inheritance diagram for gams.workspace.VarType:



Static Public Attributes

- int Unknown 0

  Unknown variable type.
- int Binary 1

  Binary variable.
- int Integer 2

  Integer Variable.
- int Positive 3

  Positive variable.
- int Negative 4

  Negative variable.
- int Free 5

  Free variable.
- int SOS1 6

  Special Ordered Set 1.
- int SOS2 7

  Special Ordered Set 2.
- int SemiCont 8

  Semi-continuous variable.
- int SemiInt 9

  Semi-integer variable.

4.46.1 Detailed Description

Variable subtype.

Index