

KAPSARC's GAMS Style Guide

Summary: GAMS Style Rules

1. File Names
2. Identifiers
3. Indentation
4. Spacing
5. Commenting Guidelines

Notation and Naming

File Names:

File names should be descriptive and end in *.gms*.

Identifiers:

- Sets Names, Scalars & Parameters
 - Each sector has a unique identifier used in the integration list
 - Sector: Starts with the sector abbreviation followed by the description.
 - Global: Descriptive name for shared and static coefficients
- Function & Variable Names
 - Primal: starts with the sector abbreviation, followed by the associated sector abbreviation if applicable, and then the description.
 - Dual: starts with capital “D” followed by the associated primal function/variable.
e.g. **WAEL**cons, **RF**transcapbal
- Linking Variables
Linking variable starts in the sector abbreviation followed by the linking sector abbreviation and then a description of the variable
e.g. **ELRF**consump

Syntax

Indentation:

The main purpose of indentation is to identify the start and the end of a block of code. Use four spaces to indent the code, never use tabs. Block of code is considered to be inside the declaration of sets, parameters, scalar, variables, and equations or definition of an equation. Make sure that the declaration/definition name (equation, variables ...) and the closing semicolon are in separate lines without any indentation, and then the block of code is indented. Also, the content of loops, summation, and “if statements” are considered blocks of code.

e.g.

```
set
  t final model run time period /t1*t3/
;
equation
  objective          minimizing cost
;
objective..
  objval=e=
  sum(t,
    (Imports(t)+Construct(t)+Opandmaint(t))*discfact(t))
;
```

Spacing:

Don't use spaces before and after ternary operators such as (= + - * / < >). Also, don't use spaces to separate arguments inside parentheses, arguments should only be separated by commas without spaces.

e.g.

```
t(trun)=yes;  
  
RFqualityconup.scale(RFqlim,RFcf,prop,trun,r)=1e-1;
```

Commenting Guidelines:

There are two types of comments:

- Block comments for functions.

e.g.

```
*****  
* Abstract: (The purpose of the function)  
* Precondition: (What needed to be done in order to run the function)  
* Postcondition: (What is the results of the function after the call)  
*****
```

- Inline comment: All inline comments should only describe special methods that are done in the implementation, using the following syntax e.g:

```
table WAdemval(time,r) Water demand in Billions m3 (2013 MOWE)  
      west  sout  cent  east  
t1 0.831 0.093 0.339 0.332 /*MMm3*/  
;
```