

---

**AIMMS Function Reference - Mathematical Program Suffices**

This file contains only one chapter of the book. For a free download of the complete book in pdf format, please visit [www.aimms.com](http://www.aimms.com)

# Mathematical Program Suffices

AIMMS mathematical programs support the following four collections of suffices.

The first group of suffices steers the solution process. These suffices are specified in the model before the solve statement and are used during the solution process.

- `.bratio`
- `.cutoff`
- `.domlim`
- `.iterlim`
- `.limrow`
- `.nodlim`
- `.optca`
- `.optcr`
- `.reslim`
- `.tolinfrep`
- `.workspace`

The second group of suffices contain information obtained during and at the end of the solution process. these suffices can be accessed after the solve statement.

- `.SolverStatus`
- `.ProgramStatus`
- `.SolverCalls`
- `.objective`
- `.incumbent`
- `.LinearObjective`
- `.GenTime`
- `.SolutionTime`
- `.Iterations`
- `.NumberOfConstraints`
- `.NumberOfNonzeros`
- `.NumberOfVariables`
- `.NumberOfInfeasibilities`
- `.SumOfInfeasibilities`

The third group of suffices control which AIMMS procedure should be called during the solution process and whether this calling should take place.

- `.CallbackProcedure`
- `.CallbackIterations`
- `.CallbackStatusChange`
- `.CallbackNewIncumbent`
- `.CallbackReturnStatus`
- `.CallbackAddCut`
- `.CallbackAOA`

The fourth group of suffices are obsolete ones. They are only retained in order not to invalidate converted AIMMS 2 and GAMS models.

- `.solveopt`
- `.prioropt`
- `.scaleopt`
- `.optfile`
- `.solprint`
- `.sysout`
- `.numnlins`
- `.numlnz`
- `.domusd`
- `.nodusd`
- `.integer1`
- `.integer2`
- `.integer3`
- `.integer4`
- `.integer5`
- `.real1`
- `.real2`
- `.real3`
- `.real4`
- `.real5`
- `.line`
- `.limcol`

---

**.bratio****Definition:**

The `.bratio` suffix controls the basis acceptance test. When specified it overrides the option `accept_basis`.

**Datatype:**

The value of the `.bratio` suffix is numeric.

**Remarks:**

- The suffix `.bratio` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.cutoff****Definition:**

The `.cutoff` suffix can be specified when solving mixed integer programs. When specified it overrides the option `cutoff`.

**Datatype:**

The value of the `.cutoff` suffix is numeric.

**Remarks:**

- The suffix `.cutoff` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.domlim****Definition:**

When the number of domain violations during the optimization of a non-linear program exceeds the value of the suffix `.domlim` the solution process is stopped. When specified this suffix overrides the option `maximal_number_of_domain_errors`.

**Datatype:**

The value of the `.domlim` suffix is numeric.

**Remarks:**

- The suffix `.domlim` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.iterlim****Definition:**

The `.iterlim` suffix limits the number of iterations that can be used to solve the mathematical program. When specified this suffix overrides the option `iteration_limit`.

**Datatype:**

The value of the `.iterlim` suffix is numeric.

**Remarks:**

- The suffix `.iterlim` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.limrow****Definition:**

The `.limrow` suffix limits the number of rows printed in the constraint listing per symbolic constraint. When specified it overrides the option `Number_of_rows_per_constraint_in_listing`.

**Datatype:**

The value of the `.limrow` suffix is numeric.

**Remarks:**

- The suffix `.limrow` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.nodlim****Definition:**

The `.nodlim` controls the maximum number of nodes created during the Branch and Bound process. When specified it overrides the option `maximal_number_of_nodes`.

**Datatype:**

The value of the `.nodlim` suffix is numeric.

**Remarks:**

- The suffix `.nodlim` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.optca****Definition:**

When specified, the solution process stops if the solver can guarantee that the current best solution is within the value of suffix `optca` of the global optimum. This is only valid for mixed integer programming models including mixed integer quadratic problems. When specified the suffix `.optca` overrides the option `MIP_Absolute_Optimality_Tolerance`.

**Datatype:**

The value of the `.optca` suffix is numeric.

**Remarks:**

- The suffix `.optca` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.optcr****Definition:**

When specified the solution procedure stops if the solver can guarantee that the current best solution is within suffix `.optcr` of the global optimum. This is only valid for mixed integer programming models including mixed integer quadratic problems. The `.optcr` suffix controls the append mode of the file. When specified the suffix `.optcr` overwrites the option `MIP_Relative_Optimality_Tolerance`

**Datatype:**

The value of the `.optcr` suffix is numeric.

**Remarks:**

- The suffix `.optcr` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.reslim****Definition:**

When specified, the solution process stops after `.reslim` seconds. When specified it overrides the option `time_limit`.

**Datatype:**

The value of the `.reslim` suffix is numeric.

**Remarks:**

- The suffix `.optcr` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.tolinfrep****Definition:**

When specified, the suffix `.tolinfrep` is the tolerance on row feasibility when computing the values of the suffices `.NumberOfInfeasibilities` and `.SumOfInfeasibilities`. When specified the option `.tolinfrep` overrides the option `bound_tolerance`.

**Datatype:**

The value of the `.tolinfrep` suffix is numeric.

**Remarks:**

- The suffix `.tolinfrep` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.workspace****Definition:**

The `.workspace` suffix controls the amount of workspace to be used by the solver in Mb. When specified it overrides the option `workspace`.

**Datatype:**

The value of the `.workspace` suffix is numeric.

**Remarks:**

- The suffix `.workspace` is initialized to NA. AIMMS considers it specified when its value is not equal to NA.

---

**.SolverStatus****Definition:**

The mathematical program suffix `.SolverStatus` suffix contains the solver status at the end of the solve statement.

**Datatype:**

The value of the `.SolverStatus` suffix is element and its range is `AllSolutionStates`.

**Remarks:**

- The related GAMS and AIMMS 2 name is `.SolveStat` but that value is a numeric code.
- The `.SolverStatus` suffix is also mentioned in Table [15.3](#) of the Language Reference.

---

**.ProgramStatus****Definition:**

The mathematical program suffix `.ProgramStatus` contains the status of the mathematical program at the end of the solve.

**Datatype:**

The value of the `.ProgramStatus` suffix is an element in the set `AllSolutionStates`.

**Remarks:**

- The related GAMS and AIMMS 2 name is `.modelstat` but that value is a numeric code.
- The `.ProgramStatus` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.SolverCalls****Definition:**

The mathematical program suffix `.SolverCalls` contains the number of times the mathematical program has been solved.

**Datatype:**

The value of the `.SolverCalls` suffix is an integer.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.number`.
- The `.SolverCalls` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.objective****Definition:**

The mathematical program suffix `.objective` suffix contains the value of the objective at the end of the solve.

**Datatype:**

The value of the `.objective` suffix is numeric. When the solve is not successful or infeasible the value of the `.objective` is NA.

**Remarks:**

- The equivalent GAMS and AIMMS 2 name is `.objval`.
- The `.objective` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.Incumbent****Definition:**

The `.Incumbent` suffix contains the current best solution during the solution process of AOA [22](#), MIP, MIQP and MIQCP problems.

**Datatype:**

The value of the `.Incumbent` suffix is numeric.

**Remarks:**

- The `.Incumbent` suffix is also mentioned in [Table 15.3](#) of the Language Reference.

---

**.LinearObjective****Definition:**

The `.LinearObjective` contains the objective value of the relaxation of the current node in the branch and bound process being investigated.

**Datatype:**

The value of the `.LinearObjective` suffix is numeric.

**Remarks:**

- There are two equivalent GAMS and AIMMS 2 names: `.objest` and `.bestest`.

---

## **.GenTime**

### **Definition:**

The mathematical program suffix `.GenTime` contains the time required to generate the mathematical program.

### **Datatype:**

The value of the `.GenTime` suffix is numeric and in wallclock seconds.

### **Remarks:**

- The suffix `.GenTime` has unit [second] iff (1) this unit has been declared, and (2) the option `solution_time_has_unit_seconds` is set to on. In all other cases the suffix has no unit.
- The equivalent GAMS and AIMMS 2 name is `.resgen`.
- The `.GenTime` suffix is also mentioned in Table [15.3](#).

---

**.SolutionTime****Definition:**

The mathematical program suffix `.SolutionTime` contains the time required to solve the mathematical program.

**Datatype:**

The value of the `.SolutionTime` suffix is numeric.

**Remarks:**

- The suffix `.SolutionTime` has unit [second] iff (1) this unit has been declared, and (2) the option `solution_time_has_unit_seconds` is set to on. In all other cases the suffix has no unit.
- The GAMS and AIMMS 2 equivalent name is `.resusd`.
- The `.SolutionTime` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.Iterations****Definition:**

The mathematical program suffix `.Iterations` contains the number of iterations executed by the solver.

**Datatype:**

The value of the `.Iterations` suffix is an integer.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.itrusd`.
- The `.Iterations` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.NumberOfConstraints****Definition:**

The mathematical program suffix `.NumberOfConstraints` contains the number of individual constraints in the generated mathematical program.

**Datatype:**

The value of the `.NumberOfConstraints` suffix is an integer.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.numequ`.
- The `.NumberOfConstraints` suffix is also mentioned in Table [15.3](#) of the Language Reference.

---

**.NumberOfNonzeros****Definition:**

The mathematical program suffix `.NumberOfNonzeros` contains the number of nonzeros in the generated mathematical program.

**Datatype:**

The value of the `.NumberOfNonzeros` suffix is an integer.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.numnz`.
- The `.NumberOfNonzeros` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.NumberOfVariables****Definition:**

The mathematical program suffix `.NumberOfVariables` contains the number of individual variables in the generated mathematical program.

**Datatype:**

The value of the `.NumberOfVariables` suffix is an integer.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.numvar`.
- The `.NumberOfVariables` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.NumberOfInfeasibilities****Definition:**

The mathematical program suffix `.NumberOfInfeasibilities` contains the number of individual constraints that are infeasible at the end of the solve.

**Datatype:**

The value of the `.NumberOfInfeasibilities` suffix is an integer.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.numinfes`.
- The `.NumberOfInfeasibilities` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.SumOfInfeasibilities****Definition:**

The `.SumOfInfeasibilities` contains the sum of the infeasibilities at the end of a solve.

**Datatype:**

The value of the `.SumOfInfeasibilities` suffix is numeric.

**Remarks:**

- The GAMS and AIMMS 2 equivalent name is `.suminfes`.
- The `.SumOfInfeasibilities` suffix is also mentioned in Table 15.3 of the Language Reference.

---

**.CallbackProcedure****Definition:**

The suffix `.CallbackProcedure` contains the name of the AIMMS procedure to be called for every suffix `.CallbackIterations` iterations executed.

**Datatype:**

The value of the `.CallbackProcedure` suffix is an element in the set of All-Procedures and the default is the empty element ''.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.

---

**.CallbackIterations****Definition:**

The suffix `.CallbackIterations` states after how many iterations the AIMMS procedure in the suffix `.CallbackProcedure` should be called.

**Datatype:**

The value of the `.CallbackIterations` suffix is numeric and the default is 0. When the value of this suffix is 0, the callback procedure in the suffix `.CallbackProcedure` is not called.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.

---

**.CallbackStatusChange****Definition:**

The mathematical program suffix `.CallbackStatusChange` contains the name of the AIMMS procedure to be called upon a status change of the generated mathematical program during the solution process.

**Datatype:**

The value of the `.CallbackStatusChange` suffix is an element in the set of `AllProcedures` and the default is the empty element `''`.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.

---

**.CallbackNewIncumbent****Definition:**

The mathematical program suffix `.CallbackNewIncumbent` contains the name of the AIMMS procedure to be called when a new incumbent is found during the solution process.

**Datatype:**

The value of the `.CallbackNewIncumbent` suffix is an element in the set of `AllProcedures` and the default is the empty element `''`.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.

---

**.CallbackReturnStatus****Definition:**

The mathematical program suffix `.CallbackReturnStatus` controls the continuation of the solution process. It can be set from within one of the callback procedures.

**Datatype:**

The value of the `.CallbackReturnStatus` suffix is an element in the set `ContinueAbort`.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.

---

**.CallbackAOA****Definition:**

The mathematical program suffix `.CallbackAOA` contains the name of the AIMMS procedure to be called by the AOA open solver.

**Datatype:**

The value of the `.CallbackAOA` suffix is an element in the set of `AllProcedures` and the default is the empty element `''`.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.

---

**.CallbackAddCut****Definition:**

The mathematical program suffix `.CallbackAddCut` contains the name of the AIMMS procedure to be called to add additional cuts.

**Datatype:**

The value of the `.CallbackAddCut` suffix is an element in the set of `AllProcedures` and the default is the empty element `''`.

**Remarks:**

- See also Section [15.2](#) of the Language Reference.