

# CONVERT

## Contents

---

<b>1</b>	<b>Introduction</b> . . . . .	<b>1</b>
<b>2</b>	<b>How to use CONVERT</b> . . . . .	<b>2</b>
<b>3</b>	<b>The GAMS Scalar Format</b> . . . . .	<b>2</b>
<b>4</b>	<b>User-Specified Options</b> . . . . .	<b>3</b>

---

## 1 Introduction

CONVERT is a utility which transforms a GAMS model instance into a scalar model where all confidential information has been removed or into formats used by other modeling and solution systems. CONVERT is designed to achieve the following goals:

- Permit users to convert a confidential model into GAMS scalar format so that any identifiable structure is removed. It can then be passed on to others for investigation without confidentiality being lost.
- A way of sharing GAMS test problems for use with other modeling systems or solvers.

CONVERT comes free of charge with any licensed GAMS system and can convert GAMS models into the following formats:

- AlphaECP
- AMPL
- AmplNLC
- BARON
- CoinFML
- CplexLP
- CplexMPS
- Dict
- FixedMPS
- GAMS (Scalar format)
- Jacobian
- LAGO
- LGO

- LindoMPI
- LINGO
- MINOPT
- NLP2MCP
- ViennaDag

For more information see the options section.

## 2 How to use CONVERT

CONVERT is run like any other GAMS solver. From the command line this is:

```
>> gams modelname modeltype=convert
```

where `modelname` is the GAMS model name and `modeltype` the solver indicator for a particular model type (e.g. LP, MIP, RMIP, QCP, MIQCP, RMIQCP, NLP, DNLP, CNS, MINLP, or MCP). CONVERT can also be specified via the option statement within the model itself before the solve statement:

```
option modeltype=convert;
```

## 3 The GAMS Scalar Format

By default, CONVERT generates a scalar GAMS model (`gams.gms`) from the input model. The scalar model exhibits the following characteristics:

- A model without sets or indexed parameters. It does not exhibit any of the advanced characteristics of modeling systems and is easily transformable.
- A model with a new set of individual variables, depicting each variable in the GAMS model as one of 3 types: positive, integer or binary. Each variable is numbered sequentially, i.e. all positive GAMS variables are mapped into  $n$  single variables `x1`, `x2`, ..., `xn`.
- A model with individual equations depicting each variable in the GAMS model. All equations are also numbered sequentially, that is equations `e1`, `e2`, ..., `em`.

Equation and variable bounds, as well as variable starting values are preserved from the original GAMS formulation.

As an example, suppose the user wishes to translate the GAMS Model Library model `trnsport.gms` into scalar format, One would run `gams trnsport.gms lp=convert`, which would generate the following scalar model `gams.gms`:

```
* LP written by GAMS Convert at 07/29/04 12:59:58
*
* Equation counts
*   Total      E        G        L        N        X        C
*     6         1        3        2        0        0        0
*
* Variable counts
*           x          b          i          s1s          s2s          sc          si
*   Total   cont  binary integer   sos1   sos2   scont   sint
```

```

*          7          7          0          0          0          0          0          0
* FX      0          0          0          0          0          0          0          0
*
* Nonzero counts
*   Total   const      NL      DLL
*       19     19       0       0
*
* Solve m using LP minimizing x7;

Variables  x1,x2,x3,x4,x5,x6,x7;
Positive Variables  x1,x2,x3,x4,x5,x6;
Equations  e1,e2,e3,e4,e5,e6;

e1..  - 0.225*x1 - 0.153*x2 - 0.162*x3 - 0.225*x4 - 0.162*x5 - 0.126*x6 + x7
      =E= 0;
e2..   x1 + x2 + x3 =L= 350;
e3..   x4 + x5 + x6 =L= 600;
e4..   x1 + x4 =G= 325;
e5..   x2 + x5 =G= 300;
e6..   x3 + x6 =G= 275;

* set non default bounds

* set non default levels

* set non default marginals

Model m / all /;
m.limrow=0; m.limcol=0;

Solve m using LP minimizing x7;

```

Note that the resulting scalar model does not contain any of the descriptive information about the data or the context of the constraints.

## 4 User-Specified Options

CONVERT options are passed on through option files. If you specify “<modelname>.optfile = 1;” before the SOLVE statement in your GAMS model. CONVERT will then look for and read an option file with the name *convert.opt* (see “Using Solver Specific Options” for general use of solver option files). The syntax for the CONVERT option file is

```
optname value
```

with one option on each line. For example,

```
AMPL
```

This option file would tell CONVERT to produce an AMPL input file. For file format options, the user can specify the filename for the file to be generated. For example, the option file entry

```
lingo myfile.lng
```

would generate a LINGO input file format called *myfile.lng*. Using the option *lingo* by itself, would produce the default output file for that option (*lingo.lng*).

All available options are listed in the following table.

Option	Description	Default
all	Generates all supported file formats.	
AlphaECP	Generates AlphaECP input file.	alpha.ecp
Ampl	Generates AMPL input file.	ampl.mod
AmplNLC	Generates Ampl NLC compatible file.	amplnlc.c
Baron	Generates BARON input file.	gams.bar
CoinFML	Generates Coin FML style XML.	coinfml.xml
ConeReform	Reformulation of cone =C= constraints to NLP format. 0: keep conic =C= format 1: convert conic constraints to NLP format	0
CplexLP	Generates CPLEX LP format input file.	cplex.lp
CplexMPS	Generates CPLEX MPS format input file.	cplex.mps
Dict	Convert to GAMS dictionary.	dict.txt
FileList	Generates file list of file formats generated.	file.txt
FixedMPS	Generates fixed format MPS file.	fixed.mps
Gams	Generates GAMS scalar model. This is the default conversion format used.	gams.gms
Jacobian	Writes GDX version of current point.	jacobian.gdx
GmsInsert	Inserts the line  \$if NOT '%gams.u1%' == '' \$include'%gams.u1%'  before the solve statement.	
help	Generates option summary.	
include <filename>	Start reading from a new file.	
Lago	Generates a partial Lago file.	lago.gms
Lgo	Generates an LGO FORTRAN file.	lgomain.for
LindoMPI	Generates Lindo MPI file.	lindo.mpi
Lingo	Generates Lingo input file.	lingo.lng
match	Force a complete match for all MCP variable / equation pairs.	
memo	Generates a memo file containing model statistics and files created .	memo.txt
Minopt	Generates Minopt input file.	minopt.dat
NLP2MCP	Generates GAMS scalar MCP model.	gamsmcp.gms
ObjVar	Name of objective variable. By default the objective variable is just named via index, for example x1.	
Reform	Force reformulations.	100
Terminate	Force GAMS to terminate after conversion.	
ViennaDag	Generates Vienna Dag input file.	vienna.dag