

# AMPL

---

# AMPL

---

The TOMLAB optimization environment fully support problems formulated in AMPL. TOMLAB reads AMPL problems as *nl* -files, and stores them in the standard Prob structure depending on the problem type. The problem type is automatically detected by TOMLAB when assigning an AMPL problem. The options that TOMLAB will select from are:

- LP - Linear Programming.
- MILP - Mixed Integer Linear Programming.
- QP - Quadratic Programming.
- MIQP - Mixed Integer Quadratic Programming.
- NLP - Nonlinear Programming.
- MINLP - Mixed Integer Nonlinear Programming.

## Introduction

Using interfaces between a modeling language and TOMLAB is of great benefit and improve the possibilities for analysis of a given problem. A full TOMLAB interface to the AMPL modeling language had been developed. AMPL was chosen, as a Matlab interface written in C existed.

## AMPL problems

AMPL users define their problem in ASCII files, using a text editor like Notepad. The naming convention is the problem name and various extensions, e.g. rosenbr.mod (contains the model) and rosenbr.dat (contains the data) for the Rosenbrock banana function. As the model and the data can be separated the model may be reused with several different data sets and parameter settings. This separation of model and data in AMPL is optional. It is assumed that the user has sufficient knowledge in using AMPL in this manual.

## Generating TOMLAB input files

TOMLAB uses the *nl* -file format, but may use several other files to give the user a full problem description when optimizing the problems in TOMLAB. The *nl* -file is the only required file, but it is recommended that the user include all files, as the AMPL Presolve phase may have eliminated constraints and primary variables. The *nl* -files should always be generated with Presolve active, as this is the AMPL default.

## Running AMPL

If the search path to the AMPL directory has been specified AMPL may be run from any directory. If this is not the case, the user need to specify the full path to AMPL in a shell prompts to run it.

Please refer to the AMPL manual: AMPL: A Modeling Language for Mathematical Programming. The first chapter of the book is available at the AMPL homepage; [www.ampl.com](http://www.ampl.com)<sup>[1]</sup>.

## Generating files

The input files can be generated either step-by-step from a dos prompt, or in batch mode. The batch mode is usually the preferred format.

Example of generating an *nl* -file from a dos prompt:

```
C:\> ampl
ampl: model ex1.mod;
ampl: data ex1.dat;
ampl: write gex1;
```

or simpler:

```
C:\> ampl -ogex1 ex1.mod ex1.dat;
```

The examples generate an *nl* -file in ASCII format. Example of generating an *nl* -file in batch mode:

```
C:\> ampl ex2.run
```

The *run* -file contains the generation commands. This example generates two different *nl* -files and also includes the generation of auxiliary files that are needed to fully describe the problem in TOMLAB.

Contents of the *run* -file:

```
model ex1.mod;
data ex1.dat;
option auxfiles 'acfrsu'
write gex1;
reset;
model ex2.mod;
data ex2.dat;
option auxfiles 'cfrsu'
write gex2;
```

The following table describes the auxiliary files that can be created. The *adj* -file is ignored by TOMLAB, as the value is explicitly stated in the *nl* -file.

Letter	File Name	Description
a	.adj	Constant in objective. This may have been eliminated by Presolve.
c	.col	AMPL names of variables.
f	.fix	AMPL variables fixed Presolve, including the value.
r	.row	AMPL names of the constraints.
s	.slc	AMPL names for eliminated constraints. These constraints can never be binding.
u	.unv	AMPL variables not used in problem.

The generated files have to be located in the same folder. The full file path has to be specified when running TOMLAB.

## Running TOMLAB

The user must call *amplAssign* in order to run an AMPL problem in TOMLAB. *amplAssign* will determine which problem type it is and create a problem with the correct structure.

The following command lines are required to solve an AMPL problem using TOMLAB /MINOS as a solver.

*amplAssign* may write the problem type to the Matlab Command Window.

```
Prob = amplAssign('ex1');
Result = tomRun('minos', Prob, 1);
```

Type 'help amplAssign' in the command window after starting TOMLAB, to get a full description of all the options.

## Generating a sol-file

A *sol*-file is readable by AMPL. This file can be generated by setting the third input parameter, *solFile*, to 1 when using *amplAssign*. The file will be generated with the same name as the *nl*-file, but with extension *.sol*. In order to import the solution to AMPL, run the following commands:

```
C:\> ampl
ampl: model ex1.mod;
ampl: data ex1.dat;
ampl: solution ex1.sol;
```

The problem name, solver, the exit text from the solver and the result will be displayed immediately.

All AMPL commands to view your results can now be executed. Enter 'display x;', to view the result for the variable named x in the original AMPL *mod*-file.

## Additional information

The AMPL field in the Prob structure provides some useful information for the user. There are several fields that hold AMPL specific information. The *sparseFlag* is set from TOMLAB in *amplAssign*; all others are from the AMPL problem formulation.

Variable	Description
sparseFlag	1 = The problem will be created with sparse matrices. 0 - dense. objType -1 = Maximize, 1 = Minimize.
objConst	Constant in the objective function. n con Number of constraints.
nzo	Number of non zeros in the objective gradient. solFile 1 = A .sol file will be written.
nlin	Indices for nonlinear constraints.
problemType	'LP', 'MILP', 'QP', 'MIQP', 'NLP', or 'MINLP'. vblNames AMPL variable names for x.
nlconstrNames	AMPL names for the nonlinear constraints. lconstrNames AMPL names for the linear constraints.
elimConstr	AMPL names for eliminated constraints.

## References

[1] <http://www.ampl.com/>

---

# Article Sources and Contributors

**AMPL** *Source:* <http://tomwiki.com/index.php?oldid=1196> *Contributors:* Elias