TIMES Version 2.9 User Note

User Control Switches in TIMES

Antti Lehtilä, VTT

(Updated: Oct 31, 2008)

Contents

1 Overview

This document describes the GAMS control variables that are available in TIMES as control switches that can be set by the user. As a rule, all the switches to be used in a specific TIMES model run should be set in the model run file (*.run). However, as the TIMES user shells in most cases automatically take care of inserting the proper switches into the run file, the user normally does not have to modify the run file at all. The switches are set in the highly user-friendly GUI interface of the user shell, which uses a run file template and inserts all run-specific switches correctly into the run file of each model run.

2 Basic run controls

2.1 RUN_NAME

The use of the RUN_NAME control variable is practically mandatory when running TIMES. By setting the RUN_NAME control variable, the user gives a name to the model run, which will be used when generating various output files and/or loading information from a previously generated file that has the same name. The control variable is used in the following way:

$SET RUN_NAME runname

Here the *runname* identifier is a string of letters, numbers and other characters (excluding spaces), such that the name complies with the rules for the base name of files under the operating system used.

The use of the RUN_NAME control variable for naming various files is explained in the TIMES documentation for each of the features that create or use such files.

2.2 SHELL

The SHELL control variable can be used to indicate the TIMES user shell that is being used for running TIMES models. This control variable is currently only needed when running TIMES under the ANSWER-TIMES shell, and is pre-set in the ANSWER-TIMES run template:

$SET SHELL ANSWER

3 Controls affecting equilibrium mode

3.1 TIMESED

The TIMESED control variable is one of the most important TIMES control variables. It has to be used whenever the full partial equilibrium features of TIMES are to be utilized. For running a baseline scenario to be subsequently used as the reference scenario for partial equilibrium analyses with elastic demands, the following setting should be used:

$SET TIMESED NO

This setting indicates that the user plans to use the resulting price levels from the current run as reference prices in subsequent runs with elastic demands. The setting causes the model generator to create the following (identical) two files from the Baseline run (the second file is a backup copy):

- Com_Bprice.gdx
- %RUN_NAME%_DP.gdx

For running any policy scenarios with elastic demands, using price levels from a previous run as reference prices, one must use the following setting:

$SET TIMESED YES

The reference price levels are read from a file named 'Com_Bprice.gdx ,' which is expected to reside in the current directory folder of the GAMS run. Therefore, the Baseline scenario using the setting $SET TIMESED NO has to be run before running the policy scenarios, or the correct 'Com_Bprice.gdx 'be otherwise restored from some backup copy.

If the user neither wishes the base prices to be written out, nor to run policy scenarios with elastic demands, she should not use any setting for the TIMESED control variable.

3.2 MACRO

The general equilibrium mode of TIMES can be activated by using the following setting:

$SET MACRO YES

For further details about the MACRO feature, see the TIMES-MACRO documentation available at the ETSAP site:
http://www.etsap.org/documentation.asp

4 Controls affecting objective function

4.1 OBJ

Starting from TIMES v2.5.7, the user can choose to use several alternative objective function formulations instead of the standard objective function. The alternative objective formulations can be activated in the following ways:

$SET OBJ AUTO (the default)
$SET OBJ MOD
$SET OBJ ALT
$SET OBJ LIN

The AUTO setting indicates that the user lets the model generator to automatically select the objective function among the standard formulation or the MOD 'alternative formulation according to the B(t) and E(t) parameters specified by the user. If those parameters comply with the assumptions used in the standard formulation, then the standard formulation is used. But if they do not comply with those assumptions, then the alternative formulation MOD 'is used.

The MOD/ALT/LIN settings indicate that the user unconditionally wants to use the alternative objective formulation MOD, ALT or LIN, respectively.

The MOD formulation is based on modified assumptions concerning period boundaries and investment spreads in Cases I.1.a and I.1.b only. The ALT formulation additionally uses modified capacity transfer coefficients that improve the independency of investment costs on period definitions. Finally, the LIN setting additionally assumes linear evolution of flows and activities between milestone years, but is otherwise similar to the ALT formulation.

In addition, the user can also use the following setting to ensure that the standard formulation is unconditionally used, even if the B(t) and E(t) parameters do not comply with the standard assumptions:

$SET OBJ STD

4.2 DAMAGE

The TIMES model generator supports the inclusion of so-called damage costs in the objective function. By default, if such damage costs have been defined in the model, they are also automatically included in the objective function. However, if the user wishes the damage costs to be included in the solution reporting only, the DAMAGE control variable can be set to NO. The valid settings of the control are the following:

$SET DAMAGE YES (=default)
$SET DAMAGE NO

4.3 MID_YEAR

In the standard objective formulation, both the investment payments and the operating cost payments are assumed to occur at the beginning of each year within the economic/technical lifetime of technologies. This also means that the so-called annuities of investment costs are calculated using the following formula, where r is the discount rate (see documentation, Part II):

$$CRF = (1-(1+r)^{-1})/(1-(1+r)^{-L})$$

According to this formula, the interest costs are zero if the lifetime L of the technology is only one year, because the payments are assumed to occur at the beginning of each year. This approach is often called as *beginning-of-year* discounting. However, it leads to an underestimation of the costs, because in reality the investments can be paid back only after getting some income from the investment. To avoid such underestimation, the following formula for annuities is perhaps more commonly used:

$$CRF = r/(1-(1+r)^{-L})$$

This second formula effectively assumes that the annual investment payments occur at the end of each year. This approach is often called as *end-of-year* discounting. As a good compromise between these two approaches, and highly recommended by many guidelines on good practices in cost evaluations[1], so-called *mid-year discounting* can additionally be used. Mid-year discounting can be activated in TIMES by the following setting:

$SET MID_YEAR YES

See Part II of the TIMES documentation for more information about mid-year discounting in TIMES (http://www.etsap.org/documentation.asp).

4.4 DISCSHIFT

As a generalization to the MID_YEAR setting, also the *end-of-year* discounting mentioned above can be used by using the DISCSHIFT control variable. The control variable should be set to correspond to the amount of time (in years) by which the discounting of continuous streams of payments should be shifted forward in time, with respect to the beginning of operation. Setting it to the value of 0.5 would be equal to the setting $SET MID_YEAR YES, and setting it to the value of 1.0 would be equal to end-of-year discounting, as follows:

$SET DISCSHIFT 1

[1] For example, by the U.S. government:
http://www.whitehouse.gov/omb/circulars/a094/a094.html

5 Stochastic and sensitivity analysis controls

5.1 STAGES

The stochastic mode of TIMES can be activated with the STAGES control variable, by using the following setting:

$SET STAGES YES

This setting is required for using the multi-stage stochastic programming features of TIMES. It can also be used for enabling sensitivity and tradeoff analysis features.

See the documentation on stochastic programming and tradeoff analysis in TIMES for more information on the use of this switch. The documentation is available at the ETSAP site: http://www.etsap.org/documentation.asp

5.2 SENSIS

Many useful sensitivity and tradeoff analysis features are available in TIMES, and they can be enables by activating the stochastic mode of TIMES (see above). However, such sensitivity and tradeoff analyses are often based on running the model in a series of cases that differ from each other in only in a few parameter values. In such cases the so-called warm start features can usually significantly speed up the model solution in the successive runs.

The use of the warm start facilities can be automatically enabled in sensitivity and tradeoff analysis by using the following setting instead of $SET STAGES YES:

$SET SENSIS YES

See the documentation on stochastic programming and tradeoff analysis in TIMES for more information on the use of this switch. The documentation is available at the ETSAP site: http://www.etsap.org/documentation.asp

6 Controls for stepped model solution

6.1 FIXBOH

The purpose of the FIXBOH option is to bind the first years of a model run to the same values determined during a previous optimization. The approach first requires that a reference case be run, and then by using FIXBOH the model generator sets fixed bounds for a subsequent run according to the solution values from the reference case up to the last milestone year less than or equal to the year specified by the FIXBOH control variable. The FIXBOH control has to be used together with the LPOINT control variable, in the following way:

```
$SET FIXBOH 2050
$SET LPOINT run_name
```

Here, the value of FIXBOH (2050) specifies the year, up to which the model solution will be fixed to the previous solution, and the value of LPOINT (run_name) specifies the name of the previous run, from which the previous solution is to be retrieved. Consequently, either a full GDX file or a GAMS "point file" (see section 8.1) from the previous run should be available. If no such GDX file is found, a compiler error is issued. The milestone years of the previous run must match those in the current run.

As a generalization to the basic scheme described above, the user can also request fixing to the previous solution different amounts of first years according to region. The region-specific years, up to which the model solution will be fixed, can be specified by using the TIMES REG_FIXT(reg) parameter. The FIXBOH control variable is in this case treated as a default value for REG_FIXT.

Example: Assume that you would like to analyze the 15-region ETSAP TIAM model with some shocks after the year 2030, and you are interested in differences in the model solution only in regions that have notable gas or LNG trade with the EU. Therefore, you would like to fix the regions AUS, CAN, CHI, IND, JPN, MEX, ODA and SKO completely to the previous solution, and all other regions to the previous solution up to 2030.

In the RUN file you should specify the control switches described above:

```
$SET FIXBOH 2030
$SET LPOINT run_name
```

In a model DD file you should include the values for the REG_FIXT parameter:

```
PARAMETER REG_FIXT /
AUS   2200, CAN   2200, CHI   2200, IND   2200
JPN   2200, MEX   2200, ODA   2200, SKO   2200
/;
```

6.2 TIMESTEP

The purpose of the TIMESTEP option is to run the model in a stepwise manner with increasing model horizon and limited foresight. The TIMESTEP control variable specifies the amount of years that should be optimized in each solution step. The total model horizon will be solved by successive steps, so that in each step the periods to be optimized are advanced further in the future, and all periods before them are fixed to the solution of the previous step. Figure 1 illustrates the step-wise solution approach.

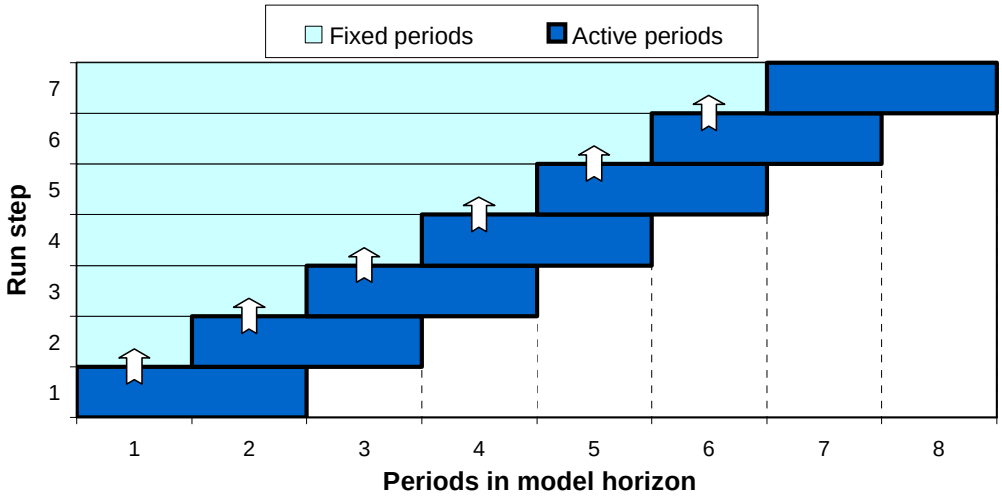

Figure 1. Sequence of optimized periods in the stepped TIMES solution approach. Each run includes also the fixed solution of all earlier periods.

The amount of overlapping years between successive steps is by default half of the active step length (the value of TIMESTEP), but it can be controlled by the user by using the TIMES G_OVERLAP parameter. Consequently, the specifications that can be used to control a stepped TIMES solution are the following:

```
$SET TIMESTEP 20                    (specified in the run file)
PARAMETER G_OVERLAP / 10  /;        (specified in a DD file)
```

In this example, the TIMESTEP control variable specifies the active step length of each successive solution step (20 years), and the G_OVERLAP parameter specifies the amount of years, by which the successive steps should overlap (10 years).

Because the time periods used in the model may be variable and may not always exactly match with the step-length and overlap, the actual active step-lengths and overlaps may somewhat differ from the values specified. At each step the model generator tries to make a best match between the remaining available periods and the prescribed step length. However, at each step at least one of the previously solved periods is fixed, and at least one remaining new period is taken into the active optimization in the current step.

7 Controls for activating TIMES extensions

7.1 ETL

The ETL control variable can be used to enable the Endogenous Technology Learning extension of TIMES, with the following setting:

$SET ETL YES

See Parts I–II of the TIMES documentation for more information on the use of ETL and this switch. The documentation is available at the ETSAP site: http://www.etsap.org/documentation.asp

7.2 CLI

The CLI control variable can be used to enable the Climate Module extension of TIMES, with the following setting:

$SET CLI YES

See Parts I–II of the TIMES documentation for more information on the use of the Climate Module and this switch. The documentation is available at the ETSAP site: http://www.etsap.org/documentation.asp

7.3 DSC

The DSC control variable can be used to enable the Discrete Capacity Investment extension of TIMES, with the following setting:

$SET DSC YES

See Parts I–II of the TIMES documentation for more information on the use of lumpy investments and this switch. The documentation is available at the ETSAP site: http://www.etsap.org/documentation.asp

7.4 VDA

The VDA control variable can be used to enable the VDA preprocessor extension of TIMES, with the following setting:

$SET VDA YES

See a separate documentation (TIMES-VDA.pdf) on the new attributes available under the VDA extension.

8 GAMS Savepoint / Loadpoint controls

Starting from TIMES 2.40, TIMES includes GAMS control variables that can be used to utilize the GAMS savepoint and loadpoint facilities. The savepoint facility makes it possible to save the basis information (levels and dual values of variables and equations) into a gdx file after model solution. The loadpoint facility makes it possible to load previously saved basis information from a gdx file and utilize it for a so-called warm start to speed-up model solution.

The GAMS control variables that can be used for the savepoint and loadpoint features in TIMES models are SPOINT and LPOINT. These control variables are *completely optional*, but can be set in the following ways if desired:

8.1 SPOINT

$SET SPOINT 1
This setting indicates that the final solution point from the model run should be saved in the file %RUN_NAME%_p.gdx, where %RUN_NAME% is the GAMS control variable that should always be set to contain the name of the current TIMES model run in the run file for the model.

$SET SPOINT 2
This setting indicates that the model generator should make an attempt to load the solution point from the file %RUN_NAME%_p.gdx, where %RUN_NAME% is the GAMS control variable that should always be set to contain the name of the current TIMES model run in the run file for the model. If the control variable LPOINT has additionally been set as well, this attempt will be made only if the loading from the file %LPOINT%_p.gdx fails.

$SET SPOINT 3
This setting combines both of the functionalities of the settings 1 and 2 described above.

$SET SPOINT YES
This setting is equivalent to $SET SPOINT 1

8.2 LPOINT

$SET LPOINT loadname
This setting indicates that the model generator should make an attempt to load the solution point from the file %LPOINT%_p.gdx. If the control variable SPOINT has additionally been set to 2 or 3, a subsequent attempt to load from %RUN_NAME%_p.gdx is also made if the loading from the file %LPOINT%_p.gdx fails.

9 Debugging controls

9.1 DEBUG

By using the DEBUG control, the user can request dumping out all user/system data structures into a file, and turn on extended quality assurance checks. The switch is activated by means of the following setting:

$SET DEBUG YES

9.2 DUMPSOL

By using the DUMPSOL control, the user can request dumping out selected solution results into a text file. The switch is activated by means of the following setting:

$SET DUMPSOL YES

9.3 SOLVE_NOW

If the user wishes to only check the input data and compile the source code, but not solve the model, the following control variable setting can be specified:

$SET SOLVE_NOW NO

9.4 XTQA

By using the XTQA control, the user can turn on extended quality assurance checks. The switch is activated by means of the following setting:

$SET XTQA YES

This setting is automatically enabled whenever $SET DEBUG YES is used.

10 Controls affecting solution reporting

10.1 SOLANS

The SOLANS control switch should be used whenever the solution reports that can be imported into the ANSWER-TIMES user shell are to be produced. The switch is enabled by using the following setting:

$SET SOLANS YES

10.2 SOLVEDA

The SOLVEDA control switch should be used whenever the solution reports that can be imported into the VEDA-BE back-end shell are to be produced. The switch is enabled by using the following setting:

$SET SOLVEDA YES

10.3 VEDAVDD

The VEDAVDD control switch should be used whenever the solution reports that can be imported into the VEDA-BE back-end shell are to be produced, and the GDX2VEDA utility is to be used for generating the VEDA-BE import files. The switch is enabled by using the following setting:

$SET VEDAVDD YES

10.4 RPT_FLOTS

The RPT_FLOTS switch can be used for controlling the timeslices that will be used for reporting the levels of the TIMES flow variables. By default, the timeslices of the original TIMES flow variables are used also for reporting. However, in many cases it may be more desirable to have all the flow levels reported at the commodity timeslices, or, for very large models, at the ANNUAL Timeslice only. The switch can be given the following values:

$SET RPT_FLOTS COM (use commodity timeslices)
$SET RPT_FLOTS ANNUAL (use the ANNUAL timeslice)

Setting any other value for RPT_FLOTS will result in the default reporting. The RPT_FLOTS setting has no effect on the reporting of marginal costs for flows.

11 Miscellaneous controls

11.1 BOTIME / EOTIME

The BOTIME and EOTIME controls can be used for adjusting the total available time span of years available in the model. All years related to the data and model must lie between BOTIME and EOTIME, inclusive. The default for BOTIME (Beginning of Time) is 1850 and the default for EOTIME (End of Time) is 2200. For example, the following settings change the BOTIME value to 1950 and EOTIME to 2250:

```
$SET BOTIME 1950
$SET EOTIME 2250
```

11.2 GDX_IREBND / GDX_IPRIC

These control flags can be used to import bounds and prices on exogenous imports/exports from a previous run, and thereby override any user-defined bounds/prices. Only bounds and prices for such imports and exports flows are imported, which were endogenous in the previous run but are exogenous in the current run. The controls can be used in the following way:

```
$SET GDX_IREBND boundfile
$SET GDX_IPRIC pricefile
```

The first setting tells the model generator to import the flow-levels of imports and exports from the file 'boundfile.gdx', and use these levels as fixed bounds on the imports and exports in the current run (if they are exogenous in the current run and were endogenous in the earlier run). The second setting tells the model generator to import the marginal prices of imports and exports from the file 'pricefile.gdx', and define these prices on the imports and exports in the current run (if they are exogenous in the current run and were endogenous in the earlier run).

The earlier run may also have different milestone years than the current run.

11.3 REDUCE

The REDUCE control can be used to enable or disable the optional reduction algorithm of the model generator. See Section 4 in PART III of the TIMES documentation for details. If the control variable is not used, the default is to make partial model reduction by eliminating unnecessary capacity variables and substituting emission flows only. The following explicit settings can be used:

$SET REDUCE YES
$SET REDUCE NO

The setting $SET REDUCE YES activates the full reduction algorithm, and the setting $SET REDUCE NO disables all optional model reduction features. Full model reduction can be very useful with large models to reduce memory and disk space consumption.

11.4 RELAX_PRC_CG

The RELAX_PRC_CG control can be used to relax the requirement that all genuine commodity groups that are used in process-related attributes have to be explicitly associated with the processes, using the set PRC_CG. When the setting is enabled in the following way, all PRC_CG definitions can be omitted in the model:

$SET RELAX_PRC_CG YES

11.5 VALIDATE

A greatly simplified formulation of the objective function and capacity constraints, emulating the MARKAL model generator, can be turned-on by the following control variable specification:

$SET VALIDATE YES

The use of this control variable is discouraged.

11.6 VAR_UC

The VAR_UC control variable can be used to enable or disable the explicit use of slack variables in user constraints. By default, no explicit slack variables are used and all the user constraints are either equalities or inequalities, depending on the bound type specified. However, if the slack variables are enabled, all the user constraints are defined as equality constraints, using bounds on the slack variables to define the actual type of the constraint. This can be useful for e.g. more efficient specification of ranges, and it is required when using the stochastic or sensitivity modes. The slacks are enabled by using the following setting:

$SET VAR_UC YES

11.7 VINTOPT

The VINTOPT control variable is currently for experimental use only.

The TIMES model generator supports technology vintaging for basically all processes. Normally any technology characteristics defined for a vintaged process describe the characteristics of new capacity installed in the year specified. However, in TIMES the same characteristics are by default used for all capacity installed for the period in question, which can lead to considerably more accelerated technology development, which depends on the lengths of periods. To avoid such distortions caused merely by period length definitions, the following settings can be used:

$SET VINTOPT 1
$SET VINTOPT 2

When the setting VINTOPT 1 is used, all vintaged characteristics of technologies are automatically adjusted so that the average characteristics of new capacity installed for each period correspond to the original data. When the setting VINTOPT 2 is used, all vintaged processes are modeled using a different approach, which preserves the average characteristics of new capacity installed for each period, as originally defined by the TIMES attributes.