TIMES Version 4.4 User Note  
Dynamic Timeslice Configurations  
Experimental TIMES Feature

# Introduction

The original design of TIMES assumes that within each region, the definition of the timeslice tree applies to all model periods, such that one cannot employ using different timeslice resolutions in different periods. In fact, allowing dynamically changing timeslice trees would tend to make both the model pre-processing and equation formulations substantially more complex, and therefore this original design decision on a static timeslice tree may be considered well justified.

However, a TIMES user has presented the need for being able to use parametric higher time-slice resolution in the near term in the TIMES models, and lower time slice resolution in the longer term as demand and generation profiles become more uncertain. For example, using hourly resolution in the short term, and day/night/peak resolution 40 years (EOH) into the future. According to the user, the main idea would be to maximize the use of the temporal-technical functional richness of TIMES in the short-medium term without providing false accuracy in the longer term. However, one might also think of using higher time-slice resolution in some later model periods, in order to better capture the effects of the increasing contribution from variable renewable generation and the role of energy storage in certain future periods.

Because of the potential usefulness of employing such dynamic timeslice configurations, there is now an experimental "light-weight" implementation available for dynamic timeslice trees, briefly summarized here.

# Design approach

## Overview

In its general interpretation, the proposed functionality might imply the need of making all of the following input attributes period-dependent in TIMES:

* Timeslice mappings to timeslice levels (TS\_GROUP)
* Timeslice mappings to parent timeslices (TS\_MAP)
* Year fractions (G\_YRFR), with interpolation over periods
* Process and commodity timeslice levels (PRC\_TSL, COM\_TSL)

Implementing the functionality in such a very general way would thus require both introducing several new input attributes (old attributes would need to be retained for backwards compatibility), and most likely also large parts of the GAMS code for TIMES to be redesigned and re-written. A substantial number of internal attributes related to the timeslice hierarchy would also have to be defined to be period-dependent, and the changes would widely affect major parts of the code, including all the levelizing routines for timeslice-specific data parameters.

Nonetheless, in simplified terms the suggested functionality might be characterized to have only the following minimum requirements:

* One should be able to use different timeslice trees in different periods of the model horizon; or more generally, one should be able define several different timeslice trees, and to select which one of them is to be used for each period.
* The user should have the possibility of, and responsibility for, defining any sub-annual timeslice-specific data parameters for each of the timeslice trees; at minimum these data would include at least the year fractions and demand commodity fractions for each of the timeslice trees.

Assuming these simple requirements to be considered sufficient for addressing the basic need, the implementation of the new feature appeared feasible within the current GAMS code, subject to certain limitations concerning the allowable timeslice trees discussed in more detail below. However, as it stands, the implementation should be considered only experimental, until some user experience has been gained on its usefulness.

## Limitations

The light-weight design and implementation of the dynamic timeslice trees supports defining a different timeslice tree for any subsets of model periods. At the extreme, the timeslice tree could be defined to be different in each and every period. However, the experimental implementation has the following main limitations:

1. All the timeslice tree configurations must be included in the "master" timeslice tree, such that each of the distinct timeslice configurations to be assigned to periods comprises a sub-tree (subset of full branches) of the master tree. A "full branch" refers to any timeslice branch directly below the ANNUAL timeslice, including all the nodes below it.
2. All the different timeslice configurations would have to include the same timeslice levels (e.g. ANNUAL / SEASON / DAYNITE).
3. The user would need to define any timeslice-specific data for all of the timeslice configurations (in other words, there is no automatic aggregation or "inter­polation" of timeslice-specific data between the timeslice configurations).
4. Period-dynamic UC constraints on non-ANNUAL timeslices cannot be supported.

# User input specifications

To enable the dynamic timeslice tree feature, the following switch is required:

$SET DYNTS YES

The following input data are required needed for defining a dynamic timeslice tree:

* *ALL\_TS* – the master set of all timeslices; this set should include all the timeslices in all of the different timeslice configurations*.*
* *TS\_GROUP*(r,tslvl,s) – the set defining the timeslice level of each timeslice ***s***.
* *TS\_MAP*(r,s,ts) – the set defining the parent timeslice **s** for each timeslice **ts**.
* *TS\_OFF*(r,ts,y1,y2) – set defining year ranges, indicating that the timeslice ***ts*** and all finer timeslices below it are not to be used for those model periods (milestone years) falling within the year range. *TS\_OFF* should thus be defined only for the timeslices immediately below the ANNUAL timeslice, i.e. typically for the SEASONs.
* *G\_YRFR*(r,s) – the parameter for the timeslice fractions; these must sum up to 1 over the finest timeslices of every different timeslice configuration used.
* *COM\_FR*(r,t,c,s) – the parameter for the commodity fractions; these must also sum up to 1 over the timeslices of every different timeslice configuration.

The following input data are optional for defining a dynamic timeslice tree:

* *TS\_CYCLE*(r,ts) – the parameter defining the lengths of the timeslice cycles under timeslice ***ts*** (in days). With this optional parameter, one would also be able to use cycles of different lengths in the dynamic timeslice tree. By default, the value of *TS\_CYCLE* is 365 for the ANNUAL timeslice (because normally there is a single seasonal cycle under the ANNUAL timeslice, which should cover the full year), 7 for any non-ANNUAL timeslices above the WEEKLY timeslices (one week), and 1 for the any non-ANNUAL timeslices above DAYNITE timeslices (one day).

In summary, the main difference between the data needed for a standard static timeslice tree and the data needed for a dynamic timeslice tree is the requirement of defining the set *TS\_OFF*(r,ts,y1,y2). The master timeslice tree is defined just like the single timeslice tree is defined in standard TIMES, but it now just includes more branches, and the TS\_OFF attribute defines which branches of the full timeslice tree are used in each model period. The year fractions are also defined in the same way as before, but the user would be responsible of ensuring that in each of the timeslice configurations the sum of the year fractions over the finest timeslices is equal to 1.

Likewise, the demand and any other commodity fractions are defined exactly in the same way as normally, but user is responsible of ensuring that in each timeslice configuration the sum of the commodity fractions over the finest level defined will be equal to 1, just like the user has to ensure that for the standard single configuration.

For any other timeslice-dependent data parameters, the user only needs to define all those parameters that have values defined at a **non-ANNUAL** level for all of the timeslice configurations. Usually, the amount of such other parameters defined at non-ANNUAL level is reasonably small, typically consisting mainly of some *NCAP\_AF* and *NCAP\_PKCNT* parameters for certain types of technologies (hydro, wind and solar), which are modelled with timeslice-differentiated availability factors. For example, in the ETSAP-TIAM model there are, in fact, no other timeslice-differentiated data parameters.

For example, if *NCAP\_AF* is originally defined on the SEASON level for some hydro power plants, they must be defined for the SEASON level timeslices of all timeslice configurations, when multiple configurations are used. Similarly, if *NCAP\_AF* is originally defined on the DAYNITE level for solar power plants, they would have to be defined for the DAYNITE level timeslices of all the timeslice configurations.

The dynamic timeslice tree feature has been tested with both simple test models and the ETSAP TIAM model. For the TIAM model, the tests were done by keeping the timeslice-specific model data identical to the original model, so that the objective function should remain unchanged. The results form the revised TIAM model were indeed found identical to the original model (excluding degeneracy), which confirmed that the feature was working as expected. However, substantially more experience would be needed before concluding about its usefulness.

# TIAM Test Example

Here we illustrate how the dynamic timeslice tree feature can be easily tested with TIAM model, by adding one additional DAYNITE timeslice (P=peak) under each season, for all periods beyond 2050. Because each timeslice configuration must consist of full branches, in this case we must actually define a complete second timeslice tree and then add all its branches to the full tree. Therefore, for the testing with TIAM we augment the global set of timeslices as follows:

Members of the SET ALL\_TS (the global set of timeslices):

|  |  |
| --- | --- |
| **Original members** | **New additional members** |
| S – Summer | Z – Summer |
| I – Intermediate | J – Intermediate |
| W – Winter | V – Winter |
| SD, SN – Summer DAYNITE cycle | ZD, ZP, ZN – Summer DAYNITE cycle |
| ID, IN – Intermediate DAYNITE cycle | JD, JP, JN – Intermediate DAYNITE cycle |
| WD, WN – Winter DAYNITE cycle | VD, VP, VN – Winter DAYNITE cycle |

Moreover, the sets *TS\_GROUP* and *TS\_MAP* will of course also have to be supplemented by the entries for the new timeslices, in the same way as for the original timeslices. For example, we, should define *TS\_MAP*(r,'Z',ts) for all the timeslices in the new summer DAYNITE cycle.

The core attribute for defining the dynamic timeslice tree is the set *TS\_OFF*. For the test, we would like to use the original tree for all periods up to 2055, and the new timeslice tree for periods starting from 2056. Therefore, we should define *TS\_OFF* for the season timeslices accordingly, as follows:

SET TS\_OFF /

(SET.R).Z.1990.2055

(SET.R).J.1990.2055

(SET.R).V.1990.2055

(SET.R).S.2056.2150

(SET.R).I.2056.2150

(SET.R).W.2056.2150 /;

We must also define *G\_YRFR*(r,s) for the new the new second timeslice tree, and for the test purpose we simply choose to define that the day (D) and peak (P) timeslices have both half of the *G\_YRFR* faction of the original day timeslice. Finally, we only need to supply the values for each of the following timeslice-specific attributes (because these were actually the only other timeslice-specific attributes used for non-ANNUAL timeslices in TIAM):

* COM\_FR(R,YEAR,C,TS)
* NCAP\_AF(R,YEAR,P,TS,BD)
* NCAP\_PKCNT(R,YEAR,P,TS)

For the testing purpose, we again set *COM\_FR* for the day (D) and peak (P) timeslices both to be half of the *COM\_FR* faction of the original day timeslice, so that we can expect identical results compared to the original model. And the *NCAP\_AF* and *NCAP\_PKCNT* can be just copied from the original values. With these specifications, the new model with the dynamic timeslice tree fully reproduces the original objective function, although the timeslice tree is now different in the later periods, and hence the test confirms the feature working as expected.