MooD International Support Portal Support Solution

**Aggregation Matrix Caching - An Introduction**

The aggregation cache stores the calculated results of specified Aggregation matrices. Values already calculated can be retrieved from the cache quickly without the need to recalculate each time the matrix is accessed (hit). Because of this, aggregation caching improves performance.

Cached results have a lifespan. This ranges from hourly to indefinitely. You need to determine which lifespan is best for each matrix. Base this decision on the likelihood of the results changing, and the possibility and danger of using outdated and hence incorrect values retrieved from the cache.

Once a set of cached results has expired, the next time the matrix is hit its result set is recalculated and recached, thereby repeating the caching cycle.

If caching is enabled, values are added to the cache as they are calculated for the first time. For example, if the matrix includes variables, some variable permutations will be calculated and added to the cache before others. It depends on use. However, there is a **Manage Aggregation Cache** synchronizer that you can use to populate and manage the cache.

Some matrices will not be suitable for caching. Of those that are, some will be suitable for indefinite caching, and some will require regular refreshes to counter the possibility of inaccuracy. MooD supports all of this, but you have to decide what is best for each matrix.

Smart Columns cannot be cached.

# Key points

MooD maintains a single aggregation cache. This cache stores individual result sets for cached Aggregation matrices.

Caching is set within each Aggregation matrix. There is a setting called **Caching** on the ribbon (**Home** tab, **Advanced** group). This setting does two things: it enables caching, and it sets the lifespan of the cached result set and hence the potential frequency of recaching.

By default, aggregation caching is disabled for each Aggregation matrix.

Once an Aggregation matrix allows caching, its results are cached the next time the matrix is hit by a model or included in a **Manage Aggregation Cache** synchronizer that creates the cache for that matrix.

Executing the matrix within its Designer tab does not populate the cache.

Cached result sets are valid until they expire, or until the matrix is next saved in Business Architect.

Cached results expire when there lifespan is exceeded or they are actively deleted by a **Manage Aggregation Cache** synchronizer. Matrices are recached when they are next hit and those results are then used until they too expire.

You can generate individual and global aggregation cache reports.

# The Caching frequency/lifespan setting

The **Caching** setting on the ribbon for an Aggregation matrix can be set to several lifespans/frequencies. Setting one of these enables caching. The settings are:

**Off**. Caching is not enabled for this Aggregation matrix. This is the default setting. Including a matrix set to **Off** in a **Manage Aggregation Cache** synchronizer will not override this. The synchronizer will fail for that matrix and report that the matrix is not cached.

**Infinite**. Cache and use the same set of results indefinitely. Indefinitely actually means until the matrix is changed (saving a matrix deletes its cached results) or until the cache is refreshed by a **Manage Aggregation Cache** synchronizer.

**Hourly**. Cache and use the results until the start of the next hour. For example, if the results are cached at 12:37pm, they will expire at 12:59:59. The key point is that the cache is not used for 60 minutes. It is only used for the remaining part of the hour within which it is created. Once the cache has expired, the next hit recreates it. The remaining settings work in the same way.

**Daily**. Cache the results until 23:59:59 that day.

**Weekly**. Cache the results until 23:59:59 on the Sunday of that week. **Monthly**. Cache the results until 23:59:59 on the last day of the current month. **Yearly**. Cache the results until 23:59:59 on the last day of the current year.

# The Manage Aggregation Cache synchronizer

This can do four things:

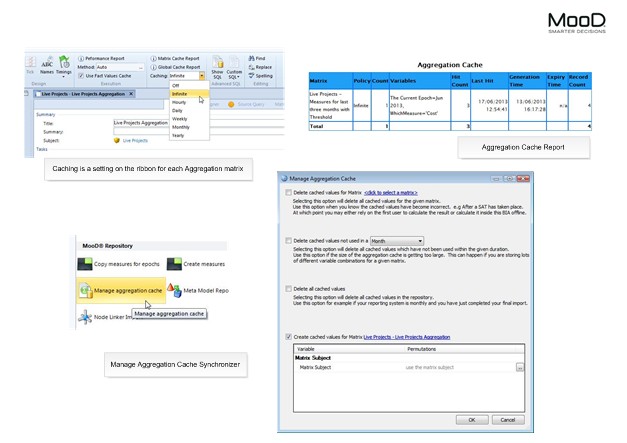
Populate the cache for a matrix. You can do this for all variable permutations or for selected variables (such as the most common ones). This allows you to bypasses the default caching mechanism. For example, for a matrix whose cache expires at midnight, you could use this synchronizer to schedule a recache at 1am, rather than wait for the first true hit of the day which would be slower due to the necessity to recache. Equally importantly, you can populate the cache with all variable permutations at once, rather than piecemeal as those permutations are requested (the ability to select the variables cached gives you a fine level of control over this).

Delete cached values for specific matrices.

Delete all cached values in the repository. This is Build 60+.

Delete cached values not used within a timeframe, for example, the last month.

***Readers Comments (if any) Images (if any)***



**Figure 1 AGC1 Aggregation Caching**