MooD International Support Portal Support Solution

**Aggregation Matrix Execution Methods - An Introduction**

Build 59+

Aggregation matrices can be executed in a number of ways called ***execution methods***. Each method uses different SQL constructs and each is suited to specific matrix design characteristics. Execution methods apply to SQL Server only.

For each Aggregation matrix, the **Execution** setting is on the ribbon (**Home** tab, **Advanced** group).

The default **Auto** setting lets MooD examine your matrix and choose what it thinks will give best performance. However, **Auto** may not always select the best method. Hence, you can manually select a specific method based on the characteristics of your matrix and the results of your own testing. The **Performance Report** (Build 60+) command on the ribbon generates a report that will help you.

You must test execution methods using data and variables that are truly representative of the deployed solution. Testing with small data sets will not give you credible results on which to base a decision.

You could manually select a method that in practice performs worse than **Auto**.

Click in the **Execution** box to display the **Execution Optimization** dialog box. This lets you see the different execution methods and choose the one to use. The different methods are:

**Auto**. MooD decides and chooses the best method. This is the default and will generally be accurate. However, you can manually select another method if testing shows that another is better.

**Temporary workings table**. Usually provides the best performance. Try this when you have large numbers of source records *and* a small number of levels.

**Cache source values**. Try this when you have one of the following. A complex source query that returns a small number of elements. If the source query is the slow part of your aggregation, this method is likely to be your best choice. Many levels – particularly visible (pinned) levels. Many facts.

**Temporary workings table (indexed)**. Try this when you have several levels visible.

**Temporary workings table (heavily indexed)**. Try this when you have several complex levels visible.

**Permanent workings table (for compatibility only)**. Use this when advised to by MooD International Support or if the other methods prove ineffectual. It is unlikely to be any better.

**Single statement (caution)**. You are highly unlikely to need this method. If your source query is the only level and doesn’t show totals, this can be fast. However, such aggregations are rare and using this method with typical aggregations could make your machine unresponsive (it times out after approximately 5 minutes).

**Best practice for execution methods**

MooD International Support Portal Support Solution

The current best advice is:

Make sure you have representative data/variables to test against. See how **Auto** performs first.

If required, compare **Auto**, **Temporary workings table** and **Cache source values**. One of these three is likely to be the best method. Use the **Performance Report** command on the ribbon (Build 60+).

If none of these gives acceptable performance, look at the usage guidelines for **Temporary workings table (indexed)** and **Temporary workings table (heavily indexed)** and test the more suitable.

The final two methods are unlikely to be better than any of the preceding methods, and the general advice is not to use them unless advised to do so. In particular, the last one (**Single statement**) can make your machine stop responding.

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



**Figure 1 AGEM1 Aggregation Matrix Execution Methods**