

Essbase Database Restructures

Changes to an Essbase outline cause changes to the Essbase index and data files, regardless of the method (Essbase Administration Services, Hyperion Planning database refreshes, or from a script).

Changes that require restructuring the database are time-consuming (unless data is discarded before restructuring). Understanding the types of restructures and what causes them can help database owners more effectively manage the impacts to users.

TYPES OF RESTRICTURES

Essbase initiates an implicit restructure after an outline is changed, whether done with the outline editor, through an automated build, or some other fashion like a Hyperion Planning database refresh. The type of restructure that is performed depends on the type of changes made to the outline.

DENSE RESTRICTURE: If a member of a dense dimension is moved, deleted, or added, Essbase restructures the blocks in the data files and creates new data files. When Essbase restructures the data blocks, it regenerates the index automatically so that index entries point to the new data blocks. Empty blocks are not removed. Essbase marks all restructured blocks as dirty, so after a dense restructure you must recalculate the database. Dense restructuring, the most time-consuming of the restructures, can take a long time to complete for large databases.

SPARSE RESTRICTURE: If a member of a sparse dimension is moved, deleted, or added, Essbase restructures the index and creates new index files. Restructuring the index is relatively fast; the time required depends on the index size.

Sparse restructures are typically fast, but depend on the size

of the index file(s). Sparse restructures are faster than dense restructures.

OUTLINE ONLY: If a change affects only the database outline, Essbase does not restructure the index or data files. Member name changes, creation of aliases, and dynamic calculation formula changes are examples of changes that affect only the database outline.

Outline restructures are very quick and typically take seconds.

Explicit restructures occur when a user requests a restructure to occur. This can be done in Essbase Administration Services or via Maxl (and EssCmd for those of you who still use it) and forces a full restructure (see dense restructure above). It is worth noting that this also removes empty blocks.

CALCULATING IMPLICATIONS AFTER RESTRUCTURES

When a restructure occurs, every block that is impacted is tagged as dirty. If Intelligent Calculations are used in the environment, they don't provide any value when a dense restructure occurs as all blocks will be calculated. When member names or formulas are changed, the block is not tagged as dirty.

WHAT DICTATES THE RESTRUCTURE TYPE

The following outline changes will force a dense restructure, which is the most time-consuming restructure.

DENSE AND SPARSE

- Defining a regular dense dimension member as dynamic calc
- Defining a sparse dimension regular member as dynamic calc or dynamic calc and store
- Defining a dense dimension dynamic calc member as

regular member

- Adding, deleting, or moving dense dimension dynamic calc and store members
- Changing dense-sparse properties [Calc Required]
- Changing a label only property [Calc Required]
- Changing a shared member property [Calc Required]
- Changing the order of dimensions [Calc Required]

DENSE (DATA FILES)

- Deleting members from a dense dimension [Calc Required]
- Adding members to a dense dimension
- Defining a dense dynamic calc member as dynamic calc and store member

SPARSE (INDEX)

- Adding members to a sparse dimension
- Moving members (excluding shared members) in a sparse dimension
- Defining a dense dynamic calc member as dynamic calc and store
- Adding, deleting, or moving a sparse dimension dynamic calc member
- Adding, deleting, or moving a sparse dimension dynamic calc and store member
- Adding, deleting, or moving a dense dimension dynamic calc member
- Changing the order of two sparse dimensions

NO RESTRUCTURE OCCURS

- Deleting members of a sparse dimension [Calc Required]
- Deleting members of an attribute dimension
- Deleting shared members from a sparse or dense dimension [Calc Required]
- Adding members to an attribute dimension
- Adding shared members to a sparse or dense dimension
- Moving a member in an attribute dimension

- Renaming a member
- Changing a member formula [Calc Required]
- Defining a sparse dynamic calc member as dynamic calc and store member
- Defining a dense or sparse dynamic calc and store member as dynamic calc
- Defining a regular dense dimension member as dynamic calc and store
- Defining a sparse dimension dynamic calc and store member or dynamic calc member as regular member
- Defining a dense dimension dynamic calc and store member as regular member
- Changing properties other than dense-sparse, label, or shared [Calc Required]
- Changing the order of an attribute dimension
- Creating, deleting, clearing, renaming, or coping an alias table
- Importing an alias table
- Setting a member alias
- Changing the case-sensitive setting
- Naming a level or generation
- Creating, changing, or deleting a UDA

WHAT DOES THIS MEAN

Understanding this can help users and administrators manage applications to better meet the needs of all those involved. When designing an application, knowledge of this topic can be instrumental in the success of the application. Here are some things to keep in mind.

- When updating an outline or refreshing a planning application, it may be faster to export level 0 (or input level) data, clear the data, perform the update, and reload/aggregate the export when changes cause a dense restructure.
- For dimensions that are updated frequently, it may be

beneficial to define those dimensions as sparse. Changes to sparse dimensions typically require only restructures to the index file(s), which are much faster.

- If frequent changes are required, enabling incremental restructuring may make sense. Using this defers dense restructures. The Essbase restructure happens on a block by block basis, and occurs the first time the data block is used. The cost is that calculations will cause restructures for all the blocks included and the calculation performance will degrade.
- Setting the isolation level to committed access may increase memory and time requirements for database restructure. Consider setting the isolation level to uncommitted access before a database restructure.
- If multiple people have access to change the outline, outline logging may be useful. This can be turned on by adding `OUTLINECHANGELOG = TRUE` in the `essbase.cfg`.
- Monitoring progress of a restructure is possible when access to the server is granted. Both sparse and dense restructures create temporary files that mirror the index and data files. Data exists in the `.pag` files while indexes are stored in `.ind` files. As the restructure occurs, there are equivalent files for each (`pan` for data files and `inn` for index files). In total, the restructure should decrease the size of the `ind` and `pag` files, but the `pan` and `inn` files can be used for a general idea of the percent of completion.