Objects: https://ibm.box.com/s/rsq9uhshtu17xw0ber315ojfo8icbvaw

Allocations Partitioning Dimension Record Count - Dimension - Update Cube Record Count.pro

Process to aid in automating generation of partitioning rollups for the partitioning of parallel processing threads along a specific dimension, by record count of specified cubes. The process will

- Build target cube 'Allocations Partitioning Dimension Record Count - <Dimension>, with dimensions <PartitioningDimension>, <VersionDimension>, <TimePeriodDimension>, 'Allocation Cube Statistics Measure'
- Create Measure '<SourceCube> - Record Count' if it does not exist.
- process record count # into target cube by Partitioning Dimension Element, hence providing statistics on the distribution of data along elements of the specified dimension

The corresponding statistics then are used by the following process to create an even # of partitioning groups for parallel processing:

Allocations - Parallel Processing Partitioning Groupings - Update.pro

Process automates generation of partitioning rollups for the partitioning of parallel processing threads along a specific dimension, by record count of specified cubes. Uses statistics in 'Allocations Partitioning Dimension Record Count - <Dimension>.cub' (gathered by process 'Allocations Partitioning Dimension Record Count - Dimension - Update Cube Record Count.pro' as per above) to create thread-specific rollups that evenly partition data within a cube.

Partitioning rollups are created as follows:

1) Total Rollup named
'ProcessingThreadPartitioningGroupings' | <SourceCube>, and
2) immediate children (rollups) named
'Thread<ThreadNo><SourceCube>' & 'Remaining<SourceCube>'

For example: if the process is run for 8 threads and against record count statistics of cube 'Product Profitability', and if the Partitioning Dimension is to be 'Cost Pool', the process will process an alternate hierarchy in the cost pool dimension will rollups

1) 'ProcessingThreadPartitioningGroupingsProduct Profitability' and
2) immediate children (rollups) named
'Thread1ProductProfitability', 'Thread2ProductProfitability', …' Thread8ProductProfitability', 'RemainingProfitability'

The corresponding elements may then be used for parallel processing threads, for example by configuring 8 threads for parallel consolidation, each with DimensionA_Element 'ND;<ThreadRollup>', like 'ND;Thread1ProductProfitability', 'ND;Thread2ProductProfitability', …' ND;Thread8ProductProfitability'. (ND; = N-Level Descendants as per <**Error! Reference source not found.**>)