Understand Transaction Semantics in Essbase

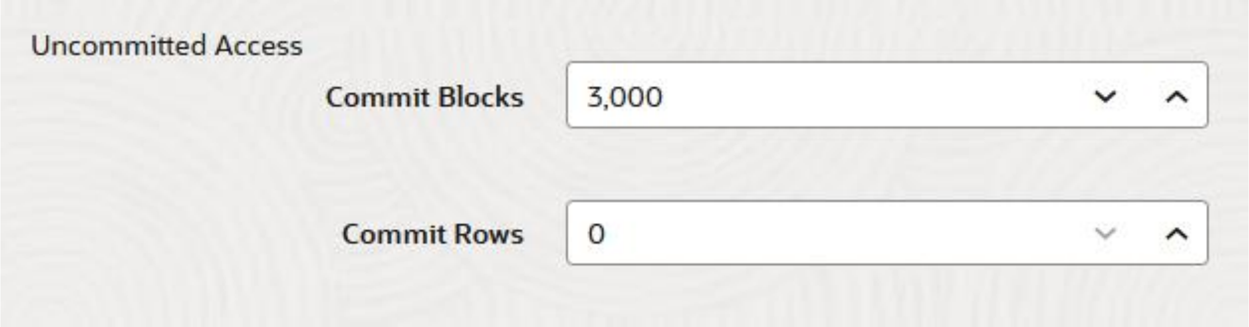
When a cube is in read/write mode, Essbase considers every update request to the server (such as a data load, a calculation, or a statement in a calculation script) as a transaction.

Each transaction has a current state: active, committed, or aborted. When data is committed, it is taken from server memory and written to the cube on disk.

The Commit Blocks/Commit Rows options in the Transactions tab of the cube inspector settings indicate the frequency at which Essbase commits data blocks or rows.

Essbase allows transactions to hold read/write locks on a block-by-block basis; Essbase releases a block after it is updated but does not commit blocks until the transaction completes or until a set limit (a “synchronization point”) has been reached.

You control when Essbase performs an explicit commit operation by specifying the following synchronization point parameters:



Uncommitted Access

Commit Blocks 3,000

Commit Rows 0

- **Commit Blocks** (number of blocks modified before a synchronization point occurs). Essbase commits after the specified number of blocks has been reached. This frequency may adjust dynamically during a calculation.

If you set Commit Blocks to 0, the synchronization point occurs at the end of the transaction.

- **Commit Rows** (number of rows to load before a synchronization point occurs). The default is 0, which means that the synchronization point occurs at the end of the data load.

If either Commit Blocks or Commit Rows has a nonzero value, a synchronization point occurs when the first threshold is reached. For example, if Commit Blocks is 10 but

Commit Rows is 0 and you load data, a synchronization point occurs after 10 blocks are updated. If Commit Blocks is 5 and Commit Rows is 5 and you load data, a synchronization point occurs after 5 rows are loaded or 5 blocks are updated, whichever happens first.

If Essbase Server runs on Oracle Exalytics In-Memory machine, uncommitted access settings are not applicable. The commit happens at the end of the command or request. Any changes you make to the Commit Blocks or Commit Rows settings are ignored.

If Essbase Server runs on a Windows independent deployment, Commit Blocks is set to 3000 by default.

If a user-defined threshold is exceeded during an operation, Essbase creates a synchronization point to commit the data processed to that point. Essbase creates as many synchronization points as are necessary to complete the operation.

Essbase analyzes the value of Commit Blocks and Commit Rows during its analysis of feasibility for parallel calculation use. If Essbase finds the values set too low, it automatically increases them.

Essbase retains redundant data to enforce transactional semantics. Allow disk space for double the size of the database to accommodate redundant data, particularly if both Commit Blocks and Commit Rows are set to 0.

To set commit blocks and commit rows

1. Navigate to **Transactions** in the Essbase web interface.

In the Redwood Interface:

   a. On the Applications page, open the application and then open the database (cube).

   b. Click **Customization**, click **Settings**, and click **Transactions**.

In the Classic Web Interface:

   c. On the Applications page, expand the application.

   d. From the **Actions** menu, to the right of the cube name, click **Settings**, then click **Transactions**.

2. Make your selections and click **Save**.