SAS®9 OLAP Cube Update - New Functionality in SAS 9.2

Janet Tierney
Senior Systems Developer
SAS OLAP Server R&D
Home remodel:
- you don’t want: tear down the whole house and rebuild from scratch
- you want: seamless integration, minimal disruption (time, noise, dirt) while you continue to live in you house

OLAP Cubes:
As you add data:
- you don’t want to rebuild from scratch
- you want to disrupt your application as little as possible, and use minimal time and resources.

SAS 9.2 will offer the ability of Incremental Cube Update.
You want to know about this now (more than half a year out):
- because it helps you with the design of your ETL processes
- you may be developing now, and have production rollout closer to SAS 9.2.
WHAT  Incremental Cube Update Is

HOW  You Do It

WHY  You Are Going to Like It!
Cube update defined in simplest terms is adding new data to an existing cube.
The great part is the original cube is undisturbed, and can even stay online while it is being updated.
What is it NOT?

- Update is NOT for major structural changes
- Cannot replace existing data
- Cannot delete data

Structural changes:
- it’s not really the same house any more. Those changes need to go through DEV/TEST – dependent client applications, Information Maps, Permission Conditions may also have to change and be tested.

SOME structural changes are allowed:
- change member property values, calculated members, adding new dimensional information (such as new products to a product dimension) without adding or deleting levels from a dimension.
**How To Do It?**

1. Prepare Input Data
2. Choose “In-place” or “Managed” Update
3. Run the update
Load NEW DATA

in order to
- add new facts (Feb sales, when Jan is already loaded in cube)
- correct values in existing cells

New data are
- new fact records (sales, quantities, etc.)
- and any new dimension records, if needed (new members such as customer names or products)
In-place Commit--the cube becomes available as soon as the update is finished.

OLD Cube Version
- Delete/Rename/Archive old cube version
- Switch-in new cube version

Managed Commit

OLD Cube Version:
- Quiesce (kick out existing query users)
  - PROC OLAPOPERATE
  - SAS Management Console OLAP Monitor
- Delete/Rename/Move (import/export to other repository)

NEW Cube Version:
- Switch in
  - Rename/Change Schema
    `PROC OLAP RENAME`
    `OLAP Cube Studio`
In-place Update: One step

"Sales Cube"

Jan

Feb

"Sales Cube"

In place update--one step. Resultant cube has same name as original.
Managed Update: Several steps

Next
1. Review updated cube
2. Delete old cube
3. Rename your new cube

Use this if you want more control over when your end users see the updated cube.

After the update, you have two instances of the cube: the two cube versions have different names, but the new cube shares data with the old cube.

Your Sales cube never goes offline until you are ready to do so!

When you are ready to replace the old version with the new, it’s a simple rename!
Here is how to do it in OLAP Cube Studio:
- Select your cube.
- Click the right mouse button.
- Select “Add Data...”.
- Specify whether you want to update the cube in-place, or give the updated version a new name and/or OLAP Schema.

- Optionally, specify the source table with the new data. There is no need to specify a new table if you have replace the data in the old table with the new data.

- Specify whether to scan the dimensions for new members and how to handle data that have no associated dimension members.
Select the dimension tables that contain the new members.
Add Code - OrinOBI

Select finish to submit the following code.

```sas
LIBNAME oligosm BASE
   "\software\sas\dev\test\oligosm\test\vars\dsm" ;
PROC OLAP
   COM   = OrinOBI
   DATA   = oligosm.SLHAALL
   ADLV   = OrinOBI
/;
META
   MM   = "OBI\mm\oligosm\";
   PM   = "OBI\pm\oligosm\"
END;
```

Save PROC OLAP Code
How to do It: Update Your Cube

PROC OLAP
  CUBE=mycube data=libname.newfacts
  ADD_DATA

  UPDATE_INPLACE    /* in place */
  or
  OUTCUBE=          /* managed */
;
RUN;
Why you’re going to like it

- Efficient
- Safe
- Flexible

Efficient: The update process does not require a cube rebuild

Safe: your current product cube is never touched

Flexible: you can implement a variety of deployment scenarios
This picture is a metaphor for “PROC OLAP MAN” adding new data to a cube. He may not be very flamboyant, but he is competent, efficient, and knows exactly what he is doing, getting the job done while the rest of the house is untouched and undisturbed.