



# Mapping Applications for Reporting in Hyperion Planning 11.1.2.x

Jonathan Johnson  
MindStream Analytics





## Agenda

---

- About MindStream Analytics
- Objectives of the Presentation
- Overview - Map Reporting Application
- Defining Mappings & POV
- Validation Rules
- Pushing Data
- When to use ASO vs. BSO
- Planning 11.1.2.3 Updates
- Q&A



## About MindStream Analytics

---

- Mission is to deliver premier consulting and managed services to clients by enhancing technology and aligning resources
- Oracle Platinum Partner with Hyperion Specialization with focused practices in Planning, Consolidations, OBIEE, Infrastructure, EPM Strategy & Training, and Hyperion Managed Services
- Our people average over 9 years of experience in the BI or BPM space
- Visit us at [www.mindstreamanalytics.com](http://www.mindstreamanalytics.com)





# Introduction

---

## Jonathan Johnson

MindStream Analytics

- Practice Leader, Planning & Analysis
- Hyperion Planning Certified
- Certified Public Accountant

**Email:** [jjohnson@mindstreamanalytics.com](mailto:jjohnson@mindstreamanalytics.com)

**Office:** 800-497-0151 ext. 711

**Cell:** 469-688-4151



# Objectives of the Presentation

---

Attendees will learn about:

- What is Mapping Applications for Reporting in Planning?
- Design considerations
- Creating a mapping and pushing data
- When to use ASO vs. BSO

**Disclaimer:** The information and commentary will be based on versions 11.1.2 – 11.1.2.2. Screenshots are based on 11.1.2.2. The core functionality covered today is largely the same between 11.1.2 – 11.1.2.2; exceptions will be pointed out. 11.1.2.3 (overview at the end), due to its significant new features does not necessarily apply to all of the information presented today.



# Mapping Reporting Applications

---

## What is it?:

Hyperion Planning now allows for a web-based interface to facilitate the mapping and loading of data from a Hyperion Planning application into a target reporting application.

First Release: 11.1.2

Source Application: Hyperion Planning

## Target Application(s):

Essbase Block Storage Option (BSO)

Essbase Aggregate Storage Option (ASO)



## Design Considerations

---

- Use Cases:

- Need to push data or a portion of data within Hyperion Planning to one or more reporting applications.
- Want the ability to report on Planning Smart Lists as standard dimensions.
  - Smart Lists are converted into regular dimensions in the reporting application.

- Unsupported Functionality:

- User Variables
- Attribute Dimensions
- ASO Databases enabled for “Duplicate Members Allowed”



## What it **CAN** do...

---

- Dimension to Dimension Mapping
- Smart List to Dimension Mapping
- Supports BSO & ASO as Reporting Applications
- Synchronization of Smart List entries from Reporting Dimension members  
(**Reporting -> Planning**)
- Push Data from Planning to Reporting (Workspace)
- Automation of pushing data via the 'PushData' utility (**11.1.2.1 +**)
- Substitution Variable support within Mapping Definitions
- Ability to create multiple mappings within a single Planning application





## What it **CAN'T** do...

---

- It does NOT create a reporting application for you
- It does not push metadata
- Doesn't support User-Defined Variables
- Attribute dimensions and attribute member selections aren't supported
- Doesn't support ASO cubes with "Duplicate Members Allowed"

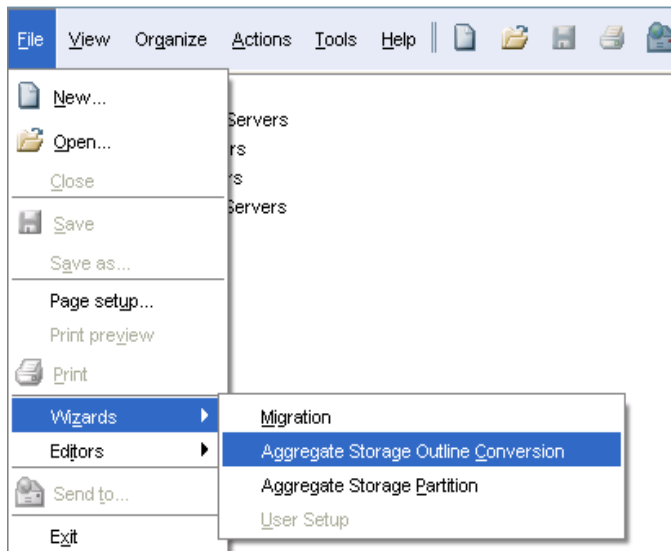
## Create and Define Mappings

### STEP 1: Create Target Reporting Application (BSO or ASO)

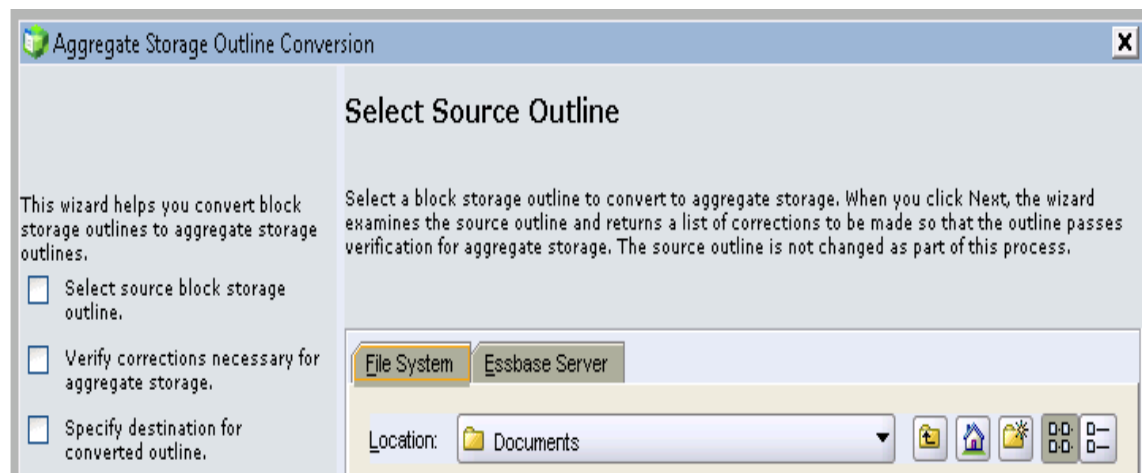


- EAS->Create new BSO or ASO

### ORACLE Essbase Administration Services



- EAS->Convert existing BSO to ASO Wizard



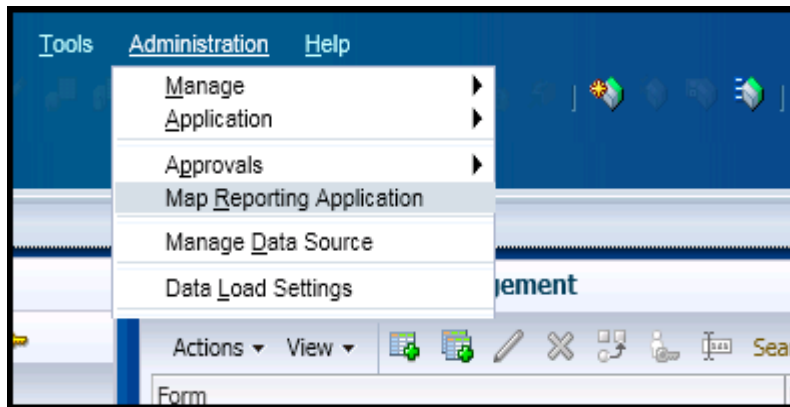
# Create and Define Mappings

## STEP 1: Create Target Reporting Application (BSO or ASO)

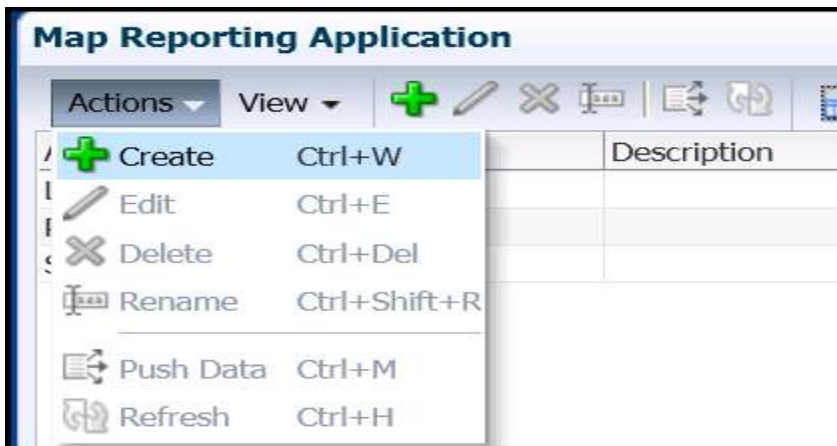
| Outline   | Properties | Text List Manager | Mod | Outline   | Properties | Query Hints | Text List Manager | Modifications |
|---|------------|-------------------|-----|---|------------|-------------|-------------------|---------------|
| <ul style="list-style-type: none"> <li>[-] Outline: Capex (Active Alias Table: Default)                             <ul style="list-style-type: none"> <li>[-] Account Accounts &lt;5&gt; (Never Share)</li> <li>[-] HSP_Rates &lt;3&gt; (Never Share)</li> <li>[-] Period Time &lt;7&gt; (Never Share)</li> <li>[-] Year &lt;2&gt; (Never Share)</li> <li>[-] Scenario &lt;10&gt; (Never Share)</li> <li>[-] Version &lt;15&gt; (Never Share)</li> <li>[-] Currency &lt;2&gt; (Never Share)</li> <li>[-] Entity Country &lt;3&gt; (Never Share)</li> <li>[-] Asset Detail &lt;3&gt; (Never Share)</li> <li>[-] Asset Class &lt;4&gt; (Never Share)</li> <li>[-] Project &lt;1&gt; (Never Share)</li> </ul> </li> </ul> |            |                   |     | <ul style="list-style-type: none"> <li>[-] Outline: ProjMap (Active Alias Table: Default)                             <ul style="list-style-type: none"> <li>[-] Account Accounts Dynamic Compression &lt;5&gt; (Label Only)</li> <li>[-] Period Time Stored # Default # &lt;7&gt;</li> <li>[-] Year Stored # Default # &lt;2&gt;</li> <li>[-] Scenario Stored # Default # &lt;10&gt;</li> <li>[-] Version Stored # Default # &lt;15&gt;</li> <li>[-] Currency Stored # Default # &lt;2&gt;</li> <li>[-] Entity Stored # Default # &lt;3&gt; (Never Share)</li> <li>[-] Asset Detail Stored # Default # &lt;3&gt;</li> <li>[-] Asset Class Stored # Default # &lt;2&gt;</li> <li>[-] Project Element Multiple Hierarchies Enabled &lt;3&gt; (Label Only)</li> <li>[-] Project Dynamic &lt;1&gt; (Label Only)</li> <li>[-] LABEL_PHYS_LOCATION Stored # Default # &lt;3&gt; (Alias: PhysicalLocation)</li> </ul> </li> </ul> |            |             |                   |               |
| <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p><b>Planning Source</b></p> </div>  |            |                   |     | <div style="border: 1px solid black; padding: 10px; width: fit-content; margin: 0 auto;"> <p><b>Reporting Target</b></p> </div>   |            |             |                   |               |

## Create and Define Mappings

### STEP 2: Administration->Map Reporting Application



### STEP 3: Create New Mapping



## Create and Define Mappings

### STEP 4: Select Applications

- Mapping Name
- Select Source Plan Type
- Select Target Reporting Application (Add Server if Necessary)

**Map Reporting Application**

Create Mapping

Select Applications | Map Dimensions | Point of View

**Details**

\* Application Mapping Name: PFPRPT

Description:

**Source Application**

Source Application: PFPSAMP

\* Plan Type: Select Plan Type

**Reporting Application**

| Essbase Server: Application: Database | Application Type |
|---------------------------------------|------------------|
| Default:bidemo                        |                  |

Save As Save Back Next Cancel Exit

## Create and Define Mappings

### STEP 5: Map Dimensions

#### Not Linked:

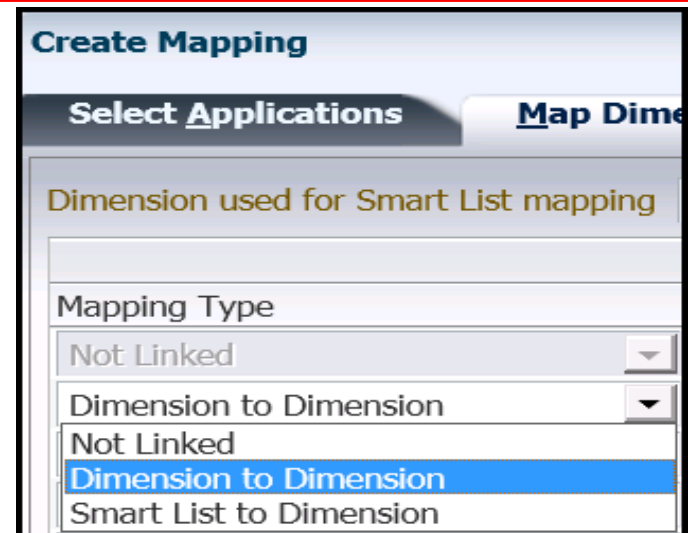
No direct mapping – defaults to POV definition

#### Dimension to Dimension:

Dimensions common in both source and target will automatically map and default to Lvl0Descendants(“Dimension”). Mapped members need to exist in both source and target when data is pushed.

#### Smart List to Dimension:

Displays the available Smart Lists. After selecting a Smart List, the associated account members are displayed. If there is only one member, it is automatically selected.



## Create and Define Mappings

### STEP 5: Map Dimensions

Select Applications | **Map Dimensions** | Point of View

Dimension used for Smart List mapping: Account

| Planning Application - (PFPSAMP) |                             |   |  | Reporting Application - (M |
|----------------------------------|-----------------------------|---|--|----------------------------|
| Mapping Type                     | Dimension / Smart List Name | Member Selection                            |  | Dimension Name             |
| Dimension to Dimension           | Account                     | ILv0Descendants(Accounts (Capital Expense)) |  | Account                    |
| Dimension to Dimension           | Asset Class                 | ILv0Descendants(Total Fixed Assets)         |  | Asset Class                |
| Dimension to Dimension           | Asset Detail                | ILv0Descendants(Total All)                  |  | Asset Detail               |
| Not Linked                       | Select Dimension            |   |  | Currency                   |
| Dimension to Dimension           | Entity                      | ILv0Descendants(Entity)                     |  | Entity                     |
| Smart List to Dimension          | Physical Location           | Physical Location                           |  | LABEL_PHYS_LOCATION        |
| Dimension to Dimension           | Period                      | BegBalance,ILv0Descendants(Year Total)      |  | Period                     |
| Dimension to Dimension           | Project                     | ILv0Descendants(Project)                    |  | Project                    |
| Not Linked                       | Select Dimension            |   |  | Project Element            |
| Dimension to Dimension           | Scenario                    | ILv0Descendants(Scenario)                   |  | Scenario                   |
| Dimension to Dimension           | Version                     | ILv0Descendants(Version)                    |  | Version                    |
| Dimension to Dimension           | Year                        | No Year,FY11,FY12,FY13,FY14                 |  | Year                       |

Save As Save Back Next Cancel Finish



## Create and Define Mappings

### STEP 6: Point of View (POV)

| Select Applications                                   |  | Map Dimensions     |  | Point of View |   |
|---|--|--------------------|--|---------------|---|
| <b>Point of View Dimension(s) - PFPSAMP</b>           |  |                    |  |               |   |
| Dimensions  |  | Member             |  |               |   |
| Currency  |  | Local              |  |               |   |
|   |  |                    |  |               |   |
| <b>Reporting Point of View Dimension(s) - MapTest</b> |  |                    |  |               |   |
| Dimensions  |  | Member             |  |               |   |
| Currency  |  | Local              |  |               |   |
| Project Element                                       |  | No Project Element |  |               | ← |

- POV displays the dimensions that are not mapped or that are missing from either application
- Can use member selection or type directly in

*Project Element ('Not Linked' in prior screen) only exists in Reporting; and therefore everything can be set to map to "No Project Element"*





# Validation Rules & Constraints

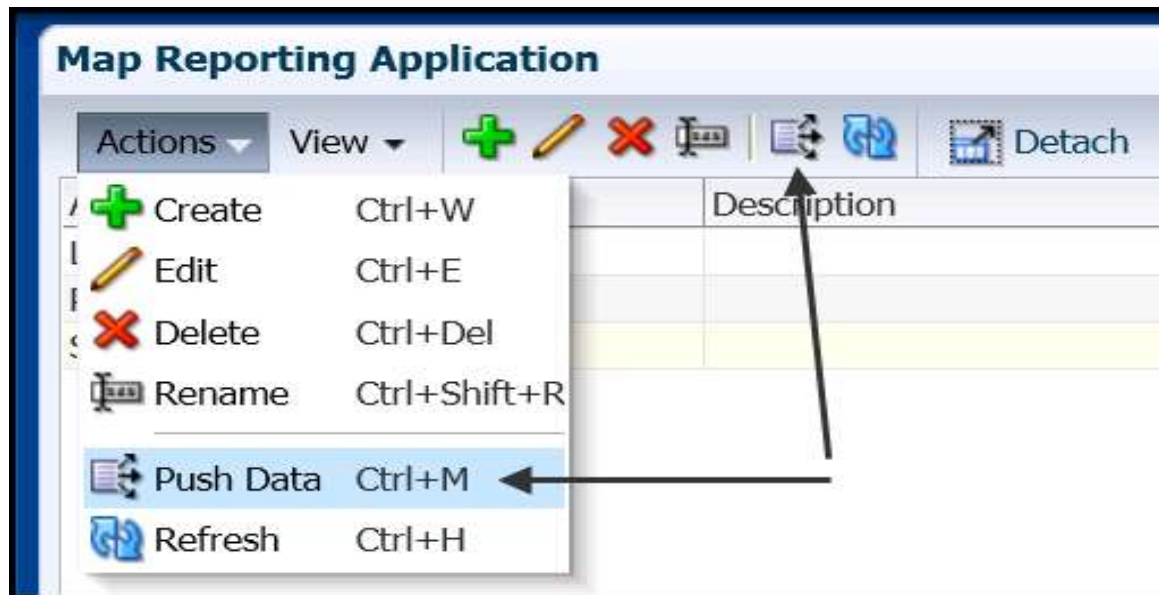
---

## Invalid Mapping Constraints:

- The reporting application is an ASO database, then the default members must be Level 0 members
  - Default members on BSO reporting databases can be any level/member
- Smart List to dimension mappings must map to Level 0 members
  - Also, the source plan type must contain a dense Account dimension with at least one member associated with a Smart List
- When defining mappings based on Descendants of (“Parent”), the (“Parent”) member must exist in the reporting applications

## Pushing Data

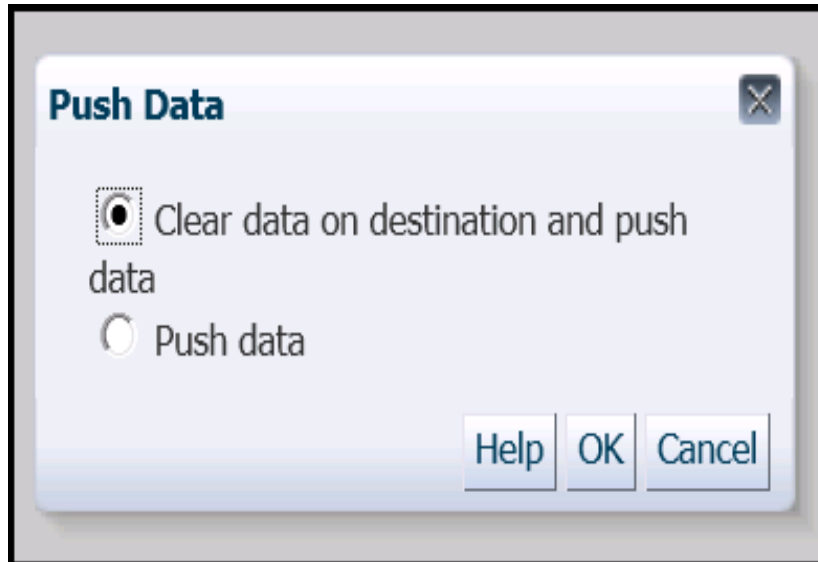
After you create and define your application mapping, you can push data to the target reporting application.



# Pushing Data

---

You will be given two options:



## Clear data on destination and push data:

- Clears data in the target reporting app before pushing data to it

## Push data:

- Pushes the data to the reporting application without first clearing the data in the target reporting application



## Pushing Data - Conditions

---

Some key conditions to point out based on the push data option you choose and the type of target reporting database you are pushing to:

### **'Clear data on destination and push data' – ASO Target:**

- Members with nonmatching names in the target are ignored
- When using member relationships, at least one member name in the source must match a member in the reporting application, otherwise this option errors out

### **'Clear data on destination and push data' – BSO Target:**

- When using member relationships, all member names in the source must match all member names in the reporting application
- Smart List to Dimension mappings require all Planning Smart List entries match all corresponding member names in the reporting application

# PushData Utility (11.1.2.1+)

New feature beginning in 11.1.2.1 is a command line utility allowing for the automation of pushing data to a reporting application:

**Utility directory:**

<MIDDLEWARE\_HOME>/user\_projects/epmsystem1/Planning/planning1

**Command Prompt syntax (from planning1 directory):**

PushData [-f:passwordFile] /U:username /A:sourceApplication /M:applicationMapping [/C]

| Parameter                     | Description  |
|-------------------------------|--|
| [-f:passwordFile]             | <b>Optional:</b> If an encrypted pw file is set up, use as the first parameter from the full file path and name specified in <i>passwordFile</i> |
| /U: <i>username</i>           | The name of the Planning administrator   |
| /A: <i>sourceApplication</i>  | The name of the source Planning application from which to push data  |
| /M: <i>applicationMapping</i> | The name of the application mapping that defines push data range   |
| [/C]                          | <b>Optional:</b> Clears the data range in the target reporting application before pushing data   |

## Smart List to Dimension Mapping

|                         |             | No Year     | No Year    | No Year    | No Year | No Year           |
|-------------------------|-------------|-------------|------------|------------|---------|-------------------|
|                         |             | Asset Units | Asset Rate | Basic Cost | Salvage | Physical Location |
| Machinery and Equipment | Line Item 1 | 2           | 6,000      | 13,800     | 600     | Location 1 ▼      |
| Machinery and Equipment | Line Item 2 | 5           | 20,000     | 115,000    | 2,000   | Location 1 ▼      |
| Laptop                  | Line Item 1 | 300         | 700        | 245,700    | 70      | Location 1 ▼      |
| Laptop                  | Line Item 2 | 400         | 650        | 299,000    | 50      | Location 1 ▼      |
| Desktop                 | Line Item 1 | 200         | 650        | 149,500    | 50      | Location 1 ▼      |
| Desktop                 | Line Item 2 | 300         | 750        | 258,750    | 100     | Location 1 ▼      |
| High End Server         | Line Item 1 | 2           | 50,000     | 117,000    | 0       | Location 2 ▼      |

Location as a Smart List

Location as a Dimension

|            |                         | No Year     | Plan                | Working | Local | USA | No Project |
|------------|-------------------------|-------------|---------------------|---------|-------|-----|------------|
|            |                         | BegBalance  |                     |         |       |     |            |
| Location 1 | Machinery and Equipment | Line Item 1 | Asset Units         | 2       |       |     |            |
|            |                         |             | Asset Rate          | 6,000   |       |     |            |
|            |                         |             | Basic Cost          | 13,800  |       |     |            |
|            |                         |             | Salvage             | 600     |       |     |            |
|            |                         |             | Capital Assumptions | 8       |       |     |            |
| Location 1 | Laptop                  | Line Item 1 | Asset Units         | 300     |       |     |            |
|            |                         |             | Asset Rate          | 700     |       |     |            |
|            |                         |             | Basic Cost          | 245,700 |       |     |            |
|            |                         |             | Salvage             | 70      |       |     |            |
|            |                         |             | Capital Assumptions | 5       |       |     |            |
| Location 1 | Desktop                 | Line Item 1 | Asset Units         | 200     |       |     |            |
|            |                         |             | Asset Rate          | 650     |       |     |            |
|            |                         |             | Basic Cost          | 149,500 |       |     |            |
|            |                         |             | Salvage             | 50      |       |     |            |
|            |                         |             | Capital Assumptions | 5       |       |     |            |
| Location 2 | High End Server         | Line Item 1 | Asset Units         | 2       |       |     |            |
|            |                         |             | Asset Rate          | 50,000  |       |     |            |
|            |                         |             | Basic Cost          | 117,000 |       |     |            |
|            |                         |             | Salvage             | -       |       |     |            |
|            |                         |             | Capital Assumptions | 5       |       |     |            |



# Smart List Synchronization

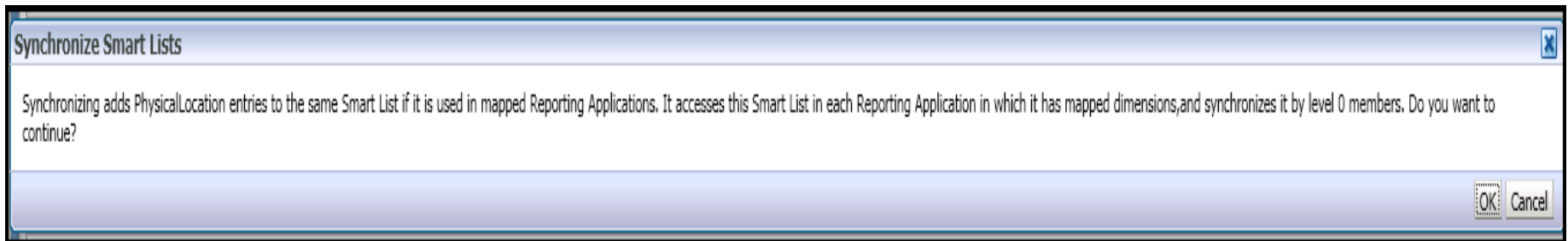
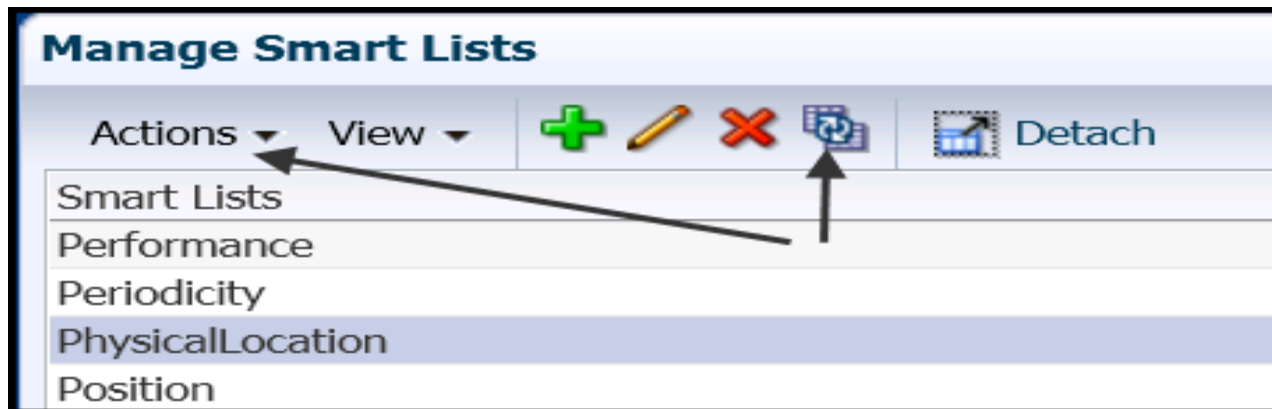
---

Member names in the reporting application can be synchronized to the Smart Lists in the source Planning application:

- This will update Smart List entries in Planning with the member names of the reporting cube dimension(s) that have been identified for 'Smart List to Dimension' mapping
- During synchronization, values from reporting applications in all existing mappings are appended after the last Smart List item in the appropriate Smart list
- If a Smart List is mapped to two dimensions, all members from the first mapping are inserted first, and then members from the second mapping are inserted. If a member already exists in a Smart List, it is not added again
- Smart List members in the Planning Smart Lists **are not deleted**, even if the corresponding dimension members on the reporting application are deleted



## Smart List Synchronization







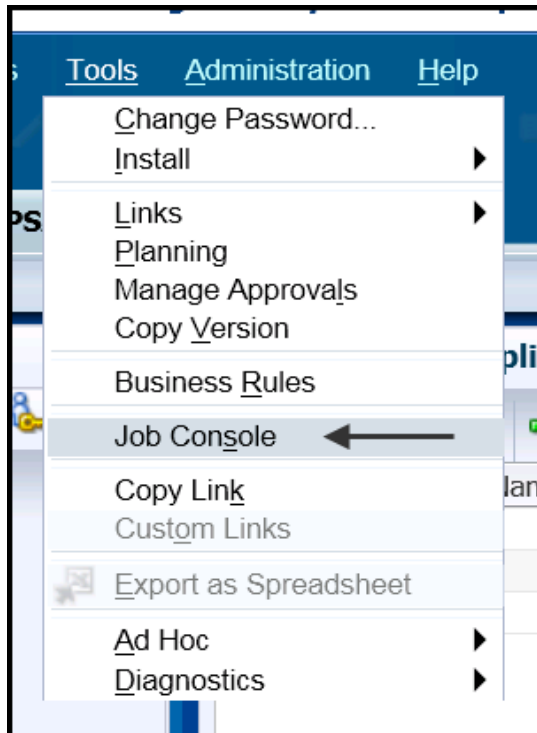
## Behind the Scenes...

---

So what is really happening under the covers?...

- DATAEXPORT is executed based on members defined in mappings
- The exported output (via .csv) file is sent to temporary directory
- This .csv file is transformed to accommodate the mappings, ultimately resulting in a .txt file to be loaded
- If the “Clear data on destination and push data” option was used, then BSO targets are cleared with the CLEARDATA command, and ASO targets are cleared via a physical ‘clear data in region’ command

## Behind the Scenes...



The Job Console will provide useful information for troubleshooting and understanding how everything is working:

**Tools -> Job Console**

The Job Console interface includes the following search filters:

- Start Time:
- End Time:
- Job Type:
- Status:
- Job Name:
- User Name:

| Job ID | Job Type  | Job Name    | User Name | Start Time      | End Time        | Run Status |
|--------|-----------|-------------|-----------|-----------------|-----------------|------------|
| 383    | Push Data | LocationMap | demoadmin | 6/18/13 9:31 PM | 6/18/13 9:31 PM | Completed  |
| 384    | Push Data | LocationMap | demoadmin | 6/18/13 9:39 PM | 6/18/13 9:40 PM | Completed  |



## Behind the Scenes...

---

```
Details for LocationMap [X]
Exporting data...

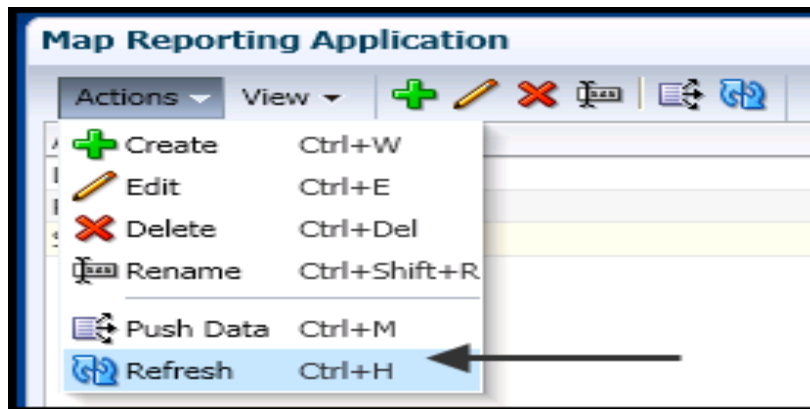
IMPORTING - AppName: MapTest2

TRANSFORMING: D:\ORCL\MW\user_projects\epmsystem1\tmp\export9193016743650624333.csv
Processed lines = 4984
Number of skipped records = 0

TRANSFORM elapsed time: 200
Data File: D:\ORCL\MW\user_projects\epmsystem1\tmp\importTrans3845086115428446262.txt
IMPORTING elapsed time: 262
TOTAL elapsed time: 1461
```

## Some items to consider

- Always “refresh” before Pushing Data:
  - This makes sure mappings recognize the most current reporting application outline



- Always use the Job Console:
  - Pushing data may complete successfully and show a completed status, but the status details may still reflect errors
  - Errors, missing members, and other useful information on transformations can help troubleshoot unexpected results in the reporting application

# Planning vs. Reporting – BSO vs. ASO

---





## What is ASO?

---

- Aggregate Storage Option (ASO) databases use a different kernel for storage than the traditional BSO database. The ASO architecture is not based on the BSO dense/sparse block-based structure, and is meant to enable rapid aggregation and dimensional scalability.
- Aggregate Storage Options when leveraged correctly with Hyperion Planning (BSO) applications can open doors to expanded functional and technical design options, as well as extensive gains in system performance and end-user experience.

**The key to “Planning vs. Reporting” is to create an atmosphere for success for each of the respective technologies – by letting them do what they’re best at doing.**



## ASO – The Good

---

- VERY FAST
- More dimensions, more members
- Large sparse datasets
- End-User experience is essentially seamless





## ASO – The Bad

---

- Much more “read-only” focus (with some exceptions)
- Not as flexible as BSO databases in terms of accommodating complex business logic and calculations
  - 11.1.2 introduced features that bridged the gap some, but BSO still excels overall at complex calculations and allocations
- No built-in variance reporting
- Exporting data is limited to Level 0 only and no columnar export
- Hyperion Planning only works against BSO (**pre-11.1.2.3**)





## BSO – The Good

---

- Naturally designed to accommodate reoccurring end-user input (write) at multiple levels of an organization
- Very good for complex calculations and allocations
  - Calculation Scripts, Business Rules, Prompts, etc.
- Great for Financial Applications



## BSO – The Bad

---

- # of Dimensions degrades performance exponentially
- Much less efficient with extremely sparse datasets
- Requires greater attention to database size over time
- Much more overhead in aggregation of data



## Overview – When to use ASO...

---

- Do not try to cram everything inside Hyperion Planning!
- Keep Planning lean, mean, and focused on what it does best – Planning, Budgeting, & Forecasting
- Large sparse datasets, organization-wide reporting of Actuals over many years, etc. should point towards a decision of leveraging ASO in combination with Planning (BSO).

# Hyperion Planning 11.1.2.3 Updates

---

- ASO Cubes in Planning!
  - Designate ASO Plan type during app creation or afterwards
  - Writeback (Level 0) and Planning web forms
- Dynamic Modeling: Creating Members “on the fly”
  - Module applications
  - Users can define members they need without a database refresh
- Flexible Custom Dimensions
  - Allows for different hierarchies across plan types
- Outline Load Utility
  - Web Interface - Import from file, Export Metadata to File, Export Data to File
- Metadata updates in excel via Smart View
- Enhanced Task Lists
  - Copy Version task
  - Enhanced dashboard
  - Excel/PDF Task reports



# Q & A

---

## Q & A!





## Questions

---

Thank You!

Jonathan Johnson – [JJohnson@MindStreamAnalytics.com](mailto:JJohnson@MindStreamAnalytics.com)

Mark Gregoire – [MGregoire@MindStreamAnalytics.com](mailto:MGregoire@MindStreamAnalytics.com)

