IBM Planning Analytics:

A Plug & Play Modeling Framework for the Creation & Maintenance of TM1 Cubes, Dimensions & Hierarchies

Prepared:
July 2015

Last Updated:
1/2020

Authors:

Sameer Gujar
Senior Managing Consultant,
TM1 Senior Consultant
IBM Data and AI Expert Labs
Mobile Phone: +1-847-858-2581
Email: sgujar@us.ibm.com

Andreas Kugelmeier
Executive Consultant, FOPM,
Planning Analytics Architect
IBM Data and AI Expert Labs
Mobile Phone: +1-215-384-7302
Email: kugelmeier@us.ibm.com

Notices & Disclaimers

Document Version History

Date	Version	Author	Description
02/07/2015	1.0		1st Version
06/07/2015	1.01		
10/19/2015	1.1	Andreas Kugelmeier	○ Add Functionality to define default zero-out parameters for each Cube to Data Source Mapping ○ Add Functionality to define default source view parameters for each Cube to Cube Data Source Mapping
11/23/2015	1.2	Andreas Kugelmeier	○ Misc. enhancements; addtl. documentation on mash-up functionality
11/27/2015	1.3	Andreas Kugelmeier	○ Add documentation on cube export and import processes
12/07/2015	1.35	Andreas Kugelmeier	○ Misc. minor documentation updates
02/23/2016	1.40	Andreas Kugelmeier	○ Add overwrite features to cube load process and (Master-)Dimension load process ○ Add SQL parameterization functionality to cube load process and (Master-)Dimension load process ○ Add ability to create compound leaf elements using parent and child names as defined in dimension data source mapping ○ Add ability to create Top Node to dimension and hierarchies
3/2017	1.45	Andreas Kugelmeier	○ Misc. minor edits
7/2018	2.0	Andreas Kugelmeier	○ V2.0: add documentation for Maintenance and Management Features pertaining to Planning Analytics Hierarchies (Hierarchy Containers)
1/2020	2.5	Andreas Kugelmeier	○ Updates to 2.4.1 du to enhancements (new versions) of process 'Manage Cube – Load or Update Data.pro'

Table of Contents

1. About this Document

This document describes a TM1 Framework for cube, dimension & hierarchy creation and maintenance. The framework was with the goal of allowing very rapid development and deployment of new or changed cubes (& dimensions) that rely on existing data feeds, without requiring actual code changes. The framework provides the capability to
- create, load & maintain dimensions, hierarchies & dimension element metadata and to
- create, load & maintain cubes
without requiring code / requiring TM1 development skills.

The TM1 objects needed to operate the framework (including sample dimensions, cubes, load files & configurations) can be found in section <TM1 Framework Objects>.

Note: The configuration models also contain metadata-elements & objects for "An IBM Cognos TM1 Plug&Play Framework for accelerated Configuration, Management & Maintenance of TM1 Security". This security model is documented and available separately. Its functionality is not discussed in this paper. Corresponding documentation is available separately

2. Cube Maintenance

The cube maintenance model allows the configuration of new and existing cubes (for existing cubes, the models provides the ability to configure cube & cell security[1]) by means of

a) **a TI process called 'Manage Cube - Create New Cube Element for Cube Configuration.pro'** to create a new dimension element in 'TM1 Cube.dim'. Alternatively, one may create the 'cube' manually via the dimension editor. The name of the new dimension element will be the name of the new cube.

b) **a lookup cube '}ElementAttributes_TM1 Cube.cub'**, allowing configuration of new cubes (naming, dimensionality, security regime)

c) **a TI process called 'Manage Cube - Create New Cube.pro'** to create a new cube according to the configuration in }ElementAttributes_TM1 Cube.cub.

d) **a TI process 'Manage Cubes – Apply Security.pro**' for applying security (a cube has to be created prior to applying security).

e) **a lookup cube '}ElementAttributes_TM1 Cube Data Source.cub'**, allowing configuration of cube data sources (cubes, files, SQL/ODBC)

f) **a mapping model 'TM1 Cube to Data Source Mapping.cub'** to map Cubes to Cube Data Sources

g) **a TI process 'Manage Cube - Load or Update Data.pro'** to update cubes with new data based on the mapping information in 'TM1 Cube to data Source Mapping.cub'.

[1] Please see 'TM1Asset_IntegratedSecurityMaintenanceAndManagementFramework.docx' for detailed information on security configuration

2.1 Creating a new cube

Cubes are created via TI process 'Manage Cube - Create New Cube.pro'. The TI-process creates the cube according to configuration metadata in }ElementAttributes_TM1 Cube.cub. The following dimension attributes ('}ElementAttributes_TM1 Cube.cub') are relevant when creating a new cube:

- **Is new Cube:**

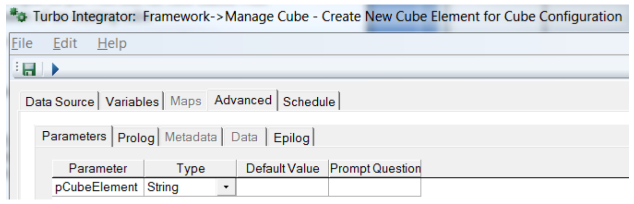
  automatically derived. Y => cube does not exit (yet); N => Cube exists
- **Cube Security**: please refer to the TM1 Security Configuration Framework documentation
- **Reference Security Dimension:** please refer to the TM1 Security Configuration Framework documentation
- **Reference Security Object:** please refer to the TM1 Security Configuration Framework documentation
- **Reference Security Element:** please refer to the TM1 Security Configuration Framework documentation
- **Cell Security Can be applied**: please refer to the TM1 Security Configuration Framework documentation
- **Cell Security is Applied**: please refer to the TM1 Security Configuration Framework documentation
- **Cell Security**: please refer to the TM1 Security Configuration Framework documentation
- **No of Dimensions**

  automatically derived & for existing cubes only – not for new cubes
- **Dimension N (1-16)**

  automatically derived for existing cubes / User input as per Pick List for new cubes: Pick the Nth dimension for a new cube / Nth dimension of an existing cube
- **Use Dimension N (1-16) for CellSecurity**: please refer to the TM1 Security Configuration Framework documentation

In the following Examples, we want to create a new cube called 'Operating Revenue & Expense New.cub':

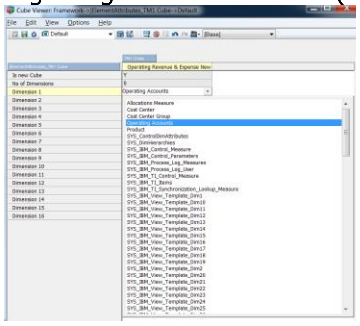
1) Create the corresponding element 'Operating Revenue & Expense New' (w/o the .cub extension) via process **'Manage Cube - Create New Cube Element for Cube Configuration.pro'**: Enter the new cube name (w/o the .cub extension) as the value for parameter pCubeElement and run the process:


Turbo Integrator: Framework->Manage Cube - Create New Cube Element for Cube Configuration
File
Edit
Help
Data Source
Variables
Maps
Advanced
Schedule
Parameters
Prolog
Metadata
Data
Epilog

Parameter	Type	Default Value	Prompt Question
pCubeElement	String		

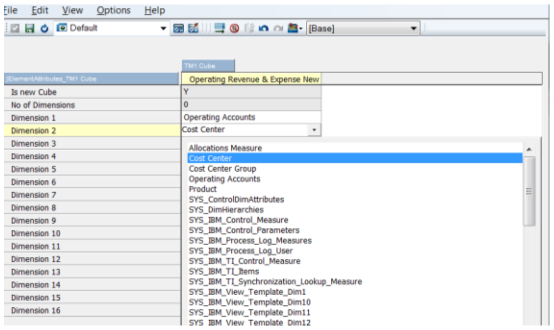
Dimension 'TM1 Cube.dim' will now have a new element called <CubeName> ('Operating Revenue & Expense New'). The cube may now be configured via }ElementAttributes_TM1 Cube.cub:

2) Attribute ‘Is new Cube’ has ‘Y’, indicating that ‘Operating Revenue & Expense New’ is a cube that does not yet exit. We do not have to configure security for this cube at this Time Period and will assign dimensions to the cube. Only existing dimensions can be added[2]. We can assign dimensions beginning with Dimension 1 (the first dimension). We will assign dimension Operating Account:


Cube Viewer: Framework->}ElementAttributes_TM1 Cube->Default
File Edit View Options Help
Default
[Base]
TM1 Cube
Allocations Measure
Cost Center
Cost Center Group
Operating Accounts
Product
SYS_ControlDimAttributes
SYS_DimHierarchies
SYS_IBM_Control_Measure
SYS_IBM_Control_Parameters
SYS_IBM_Process_Log_Measures
SYS_IBM_Process_Log_User
SYS_IBM_TI_Control_Measure
SYS_IBM_TI_Items
SYS_IBM_TI_Synchronization_Lookup_Measure
SYS_IBM_View_Template_Dim1
SYS_IBM_View_Template_Dim10
SYS_IBM_View_Template_Dim11
SYS_IBM_View_Template_Dim12
SYS_IBM_View_Template_Dim13
SYS_IBM_View_Template_Dim14
SYS_IBM_View_Template_Dim15
SYS_IBM_View_Template_Dim16
SYS_IBM_View_Template_Dim17
SYS_IBM_View_Template_Dim18
SYS_IBM_View_Template_Dim19
SYS_IBM_View_Template_Dim2
SYS_IBM_View_Template_Dim20
SYS_IBM_View_Template_Dim21
SYS_IBM_View_Template_Dim22
SYS_IBM_View_Template_Dim23
SYS_IBM_View_Template_Dim24
SYS_IBM_View_Template_Dim25

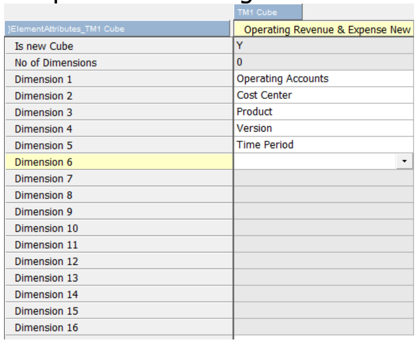
}ElementAttributes_TM1 Cube	Operating Revenue & Expense New
Is new Cube	Y
No of Dimensions	0
Dimension 1	Operating Accounts
Dimension 2	
Dimension 3	
Dimension 4	
Dimension 5	
Dimension 6	
Dimension 7	
Dimension 8	
Dimension 9	
Dimension 10	
Dimension 11	
Dimension 12	
Dimension 13	
Dimension 14	
Dimension 15	
Dimension 16	

Once a dimension has been selected, refresh the screen/query and you can now configure the 2nd dimension:


File Edit View Options Help
Default
[Base]
TM1 Cube
Allocations Measure
Cost Center
Cost Center Group
Operating Accounts
Product
SYS_ControlDimAttributes
SYS_DimHierarchies
SYS_IBM_Control_Measure
SYS_IBM_Control_Parameters
SYS_IBM_Process_Log_Measures
SYS_IBM_Process_Log_User
SYS_IBM_TI_Control_Measure
SYS_IBM_TI_Items
SYS_IBM_TI_Synchronization_Lookup_Measure
SYS_IBM_View_Template_Dim1
SYS_IBM_View_Template_Dim10
SYS_IBM_View_Template_Dim11
SYS_IBM_View_Template_Dim12

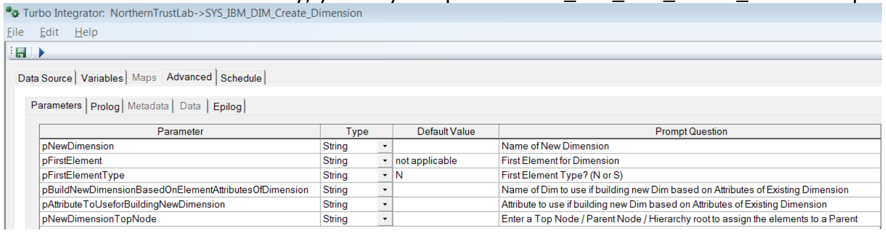
}ElementAttributes_TM1 Cube	Operating Revenue & Expense New
Is new Cube	Y
No of Dimensions	0
Dimension 1	Operating Accounts
Dimension 2	Cost Center
Dimension 3	
Dimension 4	
Dimension 5	
Dimension 6	
Dimension 7	
Dimension 8	
Dimension 9	
Dimension 10	
Dimension 11	
Dimension 12	
Dimension 13	
Dimension 14	
Dimension 15	
Dimension 16	

3) Complete cube configuration as shown in the following screenshot:



}ElementAttributes_TM1 Cube	Operating Revenue & Expense New
Is new Cube	Y
No of Dimensions	0
Dimension 1	Operating Accounts
Dimension 2	Cost Center
Dimension 3	Product
Dimension 4	Version
Dimension 5	Time Period
Dimension 6	
Dimension 7	
Dimension 8	
Dimension 9	
Dimension 10	
Dimension 11	
Dimension 12	
Dimension 13	
Dimension 14	
Dimension 15	
Dimension 16	

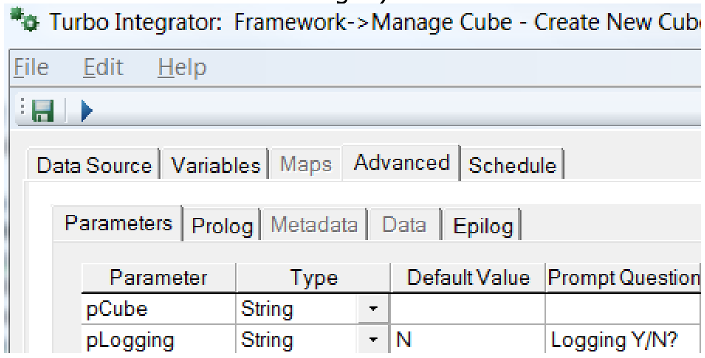
[2] To build a dimension on the fly, you may use process ‘SYS_IBM_DIM_Create_Dimension.pro’:


Turbo Integrator: NorthernTrustLab->SYS_IBM_DIM_Create_Dimension
File Edit Help
Data Source | Variables | Maps | Advanced | Schedule
Parameters | Prolog | Metadata | Data | Epilog

Parameter	Type	Default Value	Prompt Question
pNewDimension	String		Name of New Dimension
pFirstElement	String	not applicable	First Element for Dimension
pFirstElementType	String	N	First Element Type? (N or S)
pBuildNewDimensionBasedOnElementAttributesOfDimension	String		Name of Dim to use if building new Dim based on Attributes of Existing Dimension
pAttributeToUseforBuildingNewDimension	String		Attribute to use if building new Dim based on Attributes of Existing Dimension
pNewDimensionTopNode	String		Enter a Top Node / Parent Node / Hierarchy root to assign the elements to a Parent

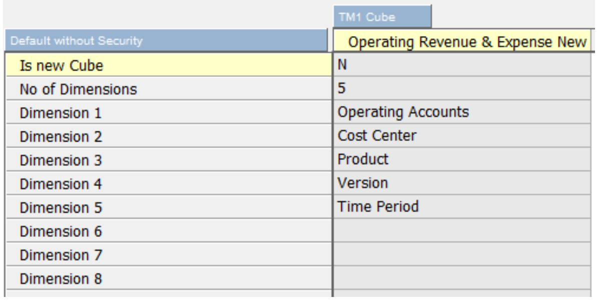
After the dimension was created, refresh the view for creating a new cube and the dimension will now show up in the drop-down menu.

4) Once all dimensions have been set, run process **'Manage Cube - Create New Cube.pro'** to create the cube by entering the cube name 'Operating Revenue & Expense New' as the value for parameter pCube. Then, refresh the query against }ElementAttributes_TM1 Cube.cub: 'Is New Cube' will now display 'N' and the dimensions of the cube will be grayed out (for existing cubes, the dimensions will be listed but cannot be changed):



Parameter	Type	Default Value	Prompt Question
pCube	String		
pLogging	String	N	Logging Y/N?

5) After running the process, refresh the view from (3) above. It will now show:



Default without Security	Operating Revenue & Expense New
Is new Cube	N
No of Dimensions	5
Dimension 1	Operating Accounts
Dimension 2	Cost Center
Dimension 3	Product
Dimension 4	Version
Dimension 5	Time Period
Dimension 6	
Dimension 7	
Dimension 8	

2.2 Cube Data Source Maintenance

To load data into a cube (or to load dimension master- and meta-data), the framework leverages data-source related metadata in '}ElementAttributes_TM1 Cube Data Source.cub'. The data source maintenance model allows the configuration of new and existing data sources by means of

a) **a TI process called 'Manage Cube Data Source - Create New Cube Data Source Element for Cube Data Source Configuration.pro'**
to create a new dimension element in 'TM1 Cube Data Source.dim'. Alternatively, one may create the 'data source' manually via the dimension editor. The name of the new dimension element will be the name of the new data source.

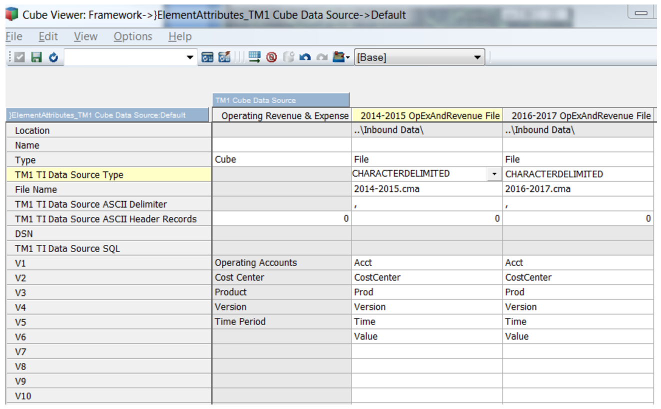
b) **a lookup cube '}ElementAttributes_TM1 Cube data Source.cub'**, allowing configuration of data sources (type, names, locations, SQL etc.)

Data sources configured via the model can then be used for loading/updating TM1 cubes. Attributes / Configuration parameters in '}ElementAttributes_TM1 Cube data Source.cub':

- **Type**: 'File', 'ODBC', 'Cube'

For data source type = File:

- **TM1 TI Data Source Type** (only for data source Type 'File'): CHARACTERDELIMITED or POSITIONDELIMITED


Cube Viewer: Framework->}ElementAttributes_TM1 Cube Data Source->Default

}ElementAttributes_TM1 Cube Data Source:Default	Operating Revenue & Expense	2014-2015 OpExAndRevenue File	2016-2017 OpExAndRevenue File
Location		..\Inbound Data\	..\Inbound Data\
Name			
Type	Cube	File	File
TM1 TI Data Source Type		CHARACTERDELIMITED	CHARACTERDELIMITED
File Name		2014-2015.cma	2016-2017.cma
TM1 TI Data Source ASCII Delimiter		,	,
TM1 TI Data Source ASCII Header Records	0	0	0
DSN			
TM1 TI Data Source SQL			
V1	Operating Accounts	Acct	Acct
V2	Cost Center	CostCenter	CostCenter
V3	Product	Prod	Prod
V4	Version	Version	Version
V5	Time Period	Time	Time
V6		Value	Value
V7			
V8			
V9			
V10			

- **Location** (only for data source Type 'File'): inbound data directory as per control cube 'SYS_IBM_Control.cub' = location where csv/txt/cma types will be dropped.

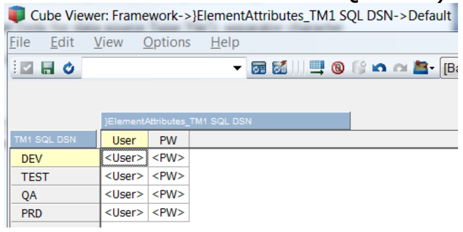
- **File Name** (only for data source Type 'File'): name of load file (note: the name of the load file could be determined automatically based on current period or other parameters.)

- **TM1 TI Data Source ASCII Delimiter** (only for data source Type 'File'): separator character

- **TM1 TI Data Source ASCII Header Records** (only for data source Type 'File'): # of header records

For data source type = ODBC:

- **DSN** (only for data source Type 'ODBC'): the **D**ata **S**ource **N**ame to use for the SQL Query. You may choose any of the DSNs available in dimension 'TM1 SQL DSN'. Create a new DSN (= Element in dimension 'TM1 SQL DSN') if needed:


Cube Viewer: Framework->}ElementAttributes_TM1 SQL DSN->Default
File Edit View Options Help

}ElementAttributes_TM1 SQL DSN

TM1 SQL DSN	User	PW
DEV	<User>	<PW>
TEST	<User>	<PW>
QA	<User>	<PW>
PRD	<User>	<PW>

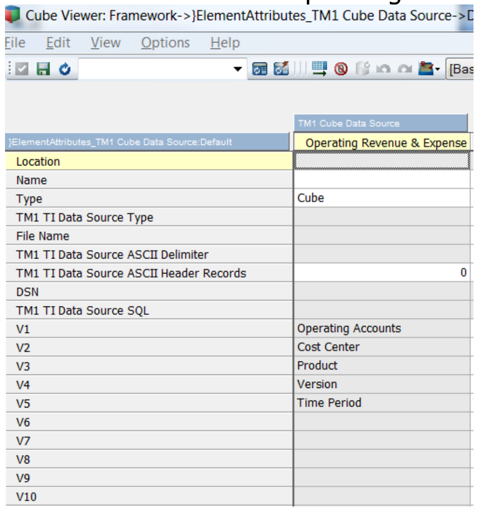
note that TM1 will use the values for SQL DSN attributes 'User' & 'PW' for the login to the Database associated with the DSN!

- **TM1 TI Data Source SQL** (only for data source Type 'ODBC'): The SQL to run. Note: if the SQL is to be parameterized, we recommend to add a corresponding number of measures/attributes to the dimension (like 'TM1 TI Data Source SQL Pt.I', 'TM1 TI Data Source SQL Pt.II',… as well as parameter attributes like 'SQL Parameter I', 'SQL Parameter II' and then to concatenate the SQL within **TM1 TI Data Source SQL** per cube rule.)

For all data source types:

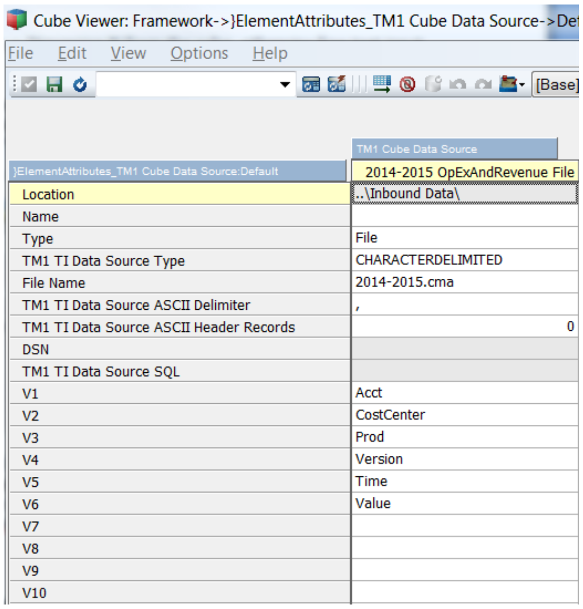
- **V1-50 (Column1-50)**: if data source type = cube, then = Dimension N from the cube, otherwise free text input. For cubes: V1-16, for files or SQL: V1-50. Examples:

  - Cube data source 'Operating Revenue & Expense' in the below screenshot: V1-16 represent the dimensions in cube 'Operating Revenue & Expense':


Cube Viewer: Framework->}ElementAttributes_TM1 Cube Data Source->D
File Edit View Options Help

}ElementAttributes_TM1 Cube Data Source:Default	TM1 Cube Data Source Operating Revenue & Expense
Location	
Name	
Type	Cube
TM1 TI Data Source Type	
File Name	
TM1 TI Data Source ASCII Delimiter	
TM1 TI Data Source ASCII Header Records	0
DSN	
TM1 TI Data Source SQL	
V1	Operating Accounts
V2	Cost Center
V3	Product
V4	Version
V5	Time Period
V6	
V7	
V8	
V9	
V10	

  - For ODBC and File data sources, V1-50 represent the columns & corresponding names in the data source. For each column, assign a name that you feel best represents the data in the corresponding data source column. You can assign any name, like in the below file data source '2014-2015 OpExAndRevenue File':


Cube Viewer: Framework->}ElementAttributes_TM1 Cube Data Source->De
File Edit View Options Help
[Base]

}ElementAttributes_TM1 Cube Data Source:Default	2014-2015 OpExAndRevenue File
Location	..\Inbound Data\
Name	
Type	File
TM1 TI Data Source Type	CHARACTERDELIMITED
File Name	2014-2015.cma
TM1 TI Data Source ASCII Delimiter	,
TM1 TI Data Source ASCII Header Records	0
DSN	
TM1 TI Data Source SQL	
V1	Acct
V2	CostCenter
V3	Prod
V4	Version
V5	Time
V6	Value
V7	
V8	
V9	
V10	

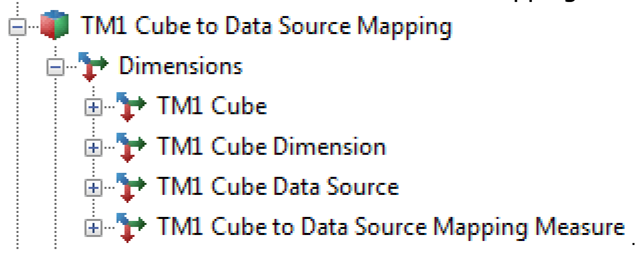
2.3 Cube to Data Source Mapping

2.3.1 Cube to Data Source Mapping model

2.3.1.1 Structure

The mapping model 'TM1 Cube to Data Source Mapping.cub' is used to map one or multiple data sources (configured in '}ElementAttributes_TM1 Data Source.cub') to a TM1 cube (configured in '}ElementAttributes_TM1 Cube.cub'). The mapping entries in 'TM1 Cube to Data Source Mapping.cub' are read and interpreted by the TI–process 'Manage Cube - Load or Update Data.pro' to load and update a cube.

Dimensions of 'TM1 Cube to Data Source Mapping.cub':



TM1 Cube to Data Source Mapping
- Dimensions
  - TM1 Cube
  - TM1 Cube Dimension
  - TM1 Cube Data Source
  - TM1 Cube to Data Source Mapping Measure

1) TM1 Cube: The cube for which the mapping is applied/configured, i.e. cubes configured in 'TM1 Cube.dim' & '}ElementAttributes_TM1 Cube.cub'.

2) TM1 Cube Dimension: a generic dimension with elements
   - All Dimensions
     - Value
     - Dimension 1
     - …
     - Dimension 16

   Purpose: configuration of data source to cube mapping for each dimension

3) TM1 Cube Data Source: the data source that is to be mapped to the cube, i.e. data sources configured in 'TM1 Cube Data Source.dim' & '}ElementAttributes_TM1 Cube Data Source.cub'.

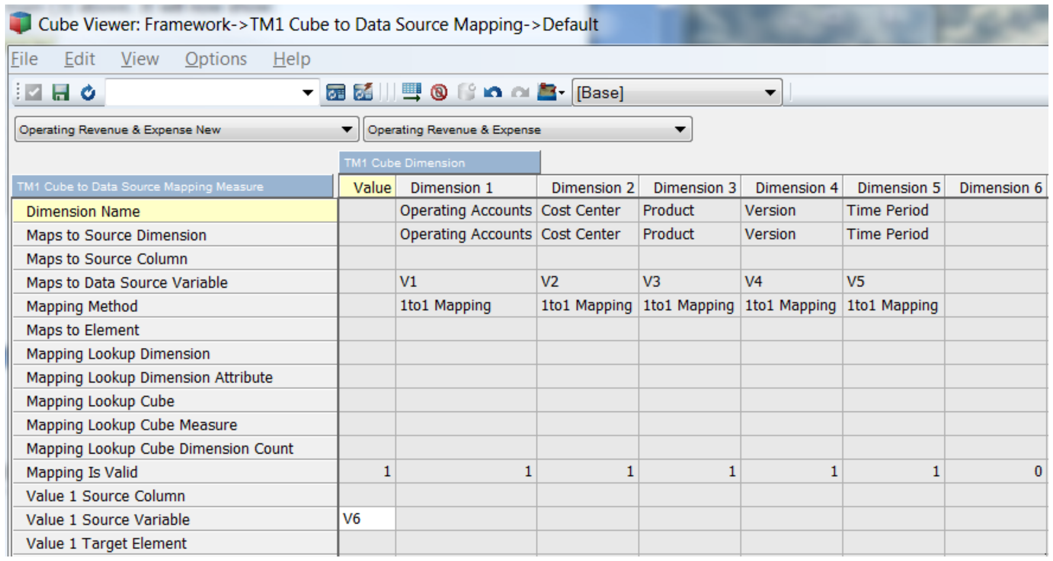
4) TM1 Cube to Data Source Mapping Measure (see below)

2.3.1.2 Data-Mapping

via TM1 Cube to Data Source Mapping Measure:

- **Dimension Name**: Name of the target cube dimension N; automatically derived.

- **Maps to Source Dimension** (only for data source = cube):
  If and where target cube and source cube use the same dimensions the mapping will be applied automatically:

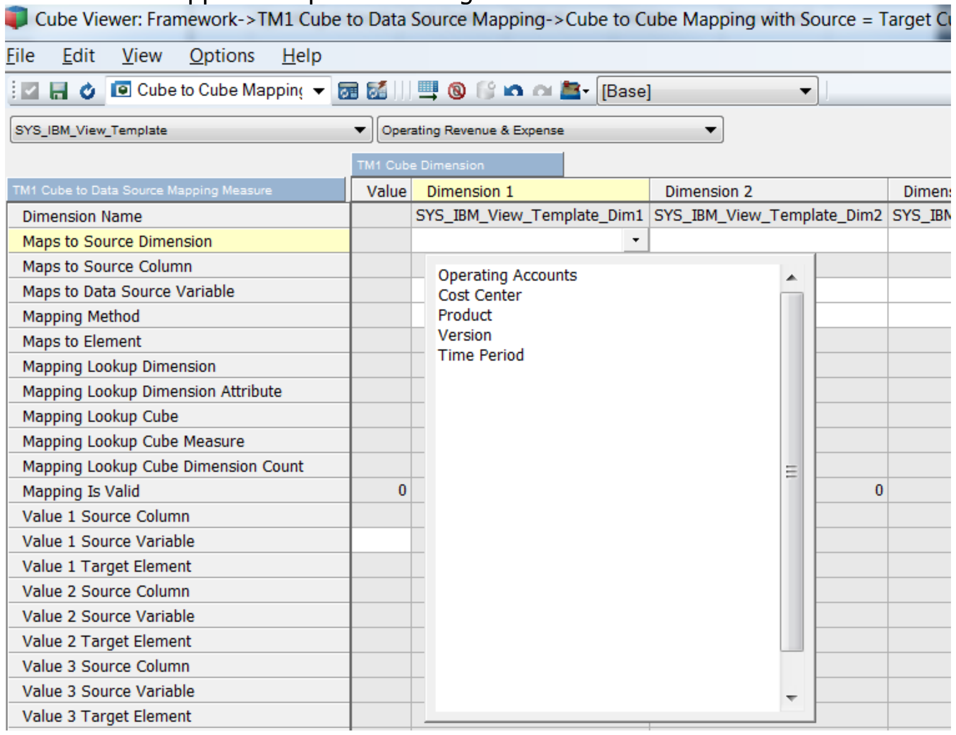

Cube Viewer: Framework->TM1 Cube to Data Source Mapping->Default

Operating Revenue & Expense New | Operating Revenue & Expense | [Base]

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period	
Maps to Source Dimension		Operating Accounts	Cost Center	Product	Version	Time Period	
Maps to Source Column							
Maps to Data Source Variable		V1	V2	V3	V4	V5	
Mapping Method		1to1 Mapping	1to1 Mapping	1to1 Mapping	1to1 Mapping	1to1 Mapping	
Maps to Element							
Mapping Lookup Dimension							
Mapping Lookup Dimension Attribute							
Mapping Lookup Cube							
Mapping Lookup Cube Measure							
Mapping Lookup Cube Dimension Count							
Mapping Is Valid	1	1	1	1	1	1	0
Value 1 Source Column							
Value 1 Source Variable	V6						
Value 1 Target Element							

In this case, all that needs to occur to complete the mapping is to assign the value 1 Source Variable which will be the Variable VN+1, with N = the # of dimensions.

If and where the source cube is <> the target cube and where source cube dimensions <> target cube dimensions, a picklist will allow the selection of a source cube dimension that is to be mapped to a particular target cube dimension:


Cube Viewer: Framework->TM1 Cube to Data Source Mapping->Cube to Cube Mapping with Source = Target C

Cube to Cube Mappin | [Base]

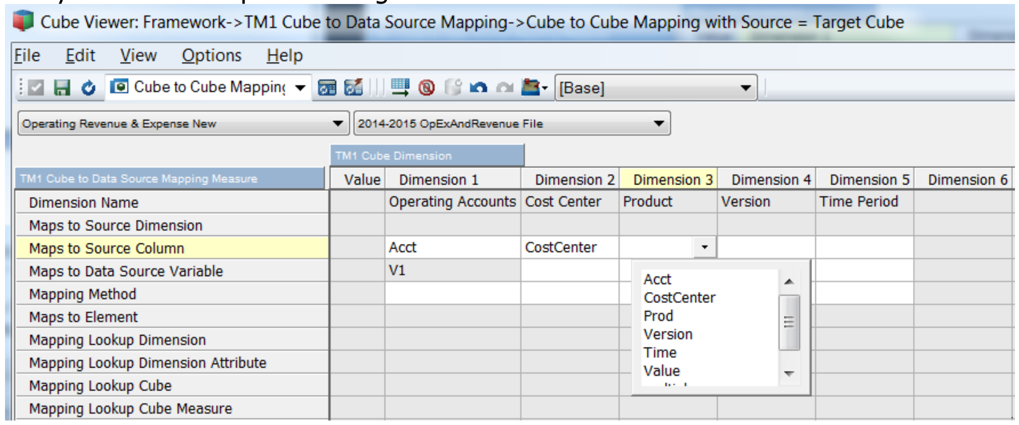
SYS_IBM_View_Template | Operating Revenue & Expense

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimen
Dimension Name		SYS_IBM_View_Template_Dim1	SYS_IBM_View_Template_Dim2	SYS_IBN
Maps to Source Dimension				
Maps to Source Column				
Maps to Data Source Variable				
Mapping Method				
Maps to Element				
Mapping Lookup Dimension				
Mapping Lookup Dimension Attribute				
Mapping Lookup Cube				
Mapping Lookup Cube Measure				
Mapping Lookup Cube Dimension Count				
Mapping Is Valid	0		0	
Value 1 Source Column				
Value 1 Source Variable				
Value 1 Target Element				
Value 2 Source Column				
Value 2 Source Variable				
Value 2 Target Element				
Value 3 Source Column				
Value 3 Source Variable				
Value 3 Target Element				

Operating Accounts
Cost Center
Product
Version
Time Period

If the source data is to be mapped to a 'default element', 'Maps to Source Dimension' may be left empty.

- **<u>Maps to Source Column</u>** (only for data source = ODBC or File):
  Pick the data source column name (from the 'Cube Data Source' configuration model) that you want to map to the target dimension:



Cube Viewer: Framework->TM1 Cube to Data Source Mapping->Cube to Cube Mapping with Source = Target Cube

File Edit View Options Help

Cube to Cube Mapping [Base]

Operating Revenue & Expense New | 2014-2015 OpExAndRevenue File

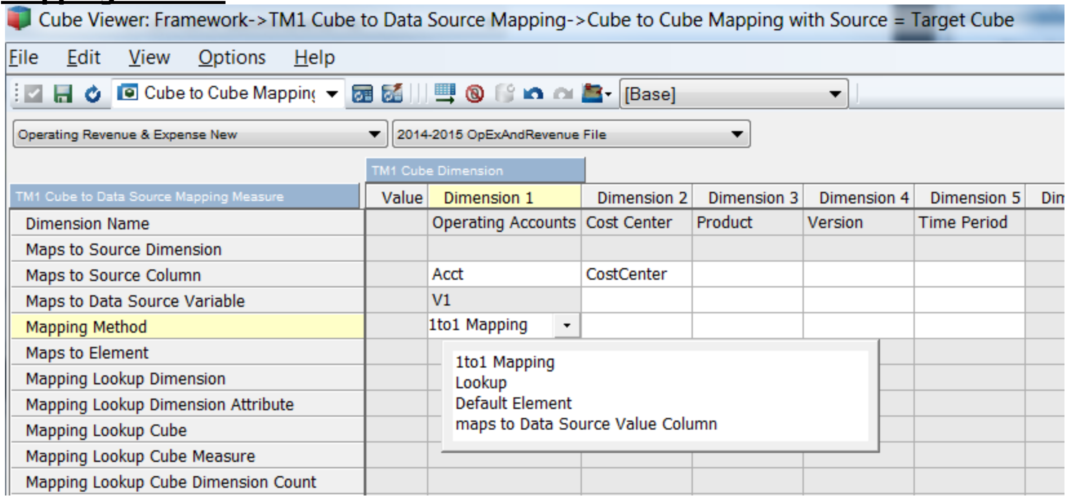
TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period	
Maps to Source Dimension							
Maps to Source Column		Acct	CostCenter				
Maps to Data Source Variable		V1					
Mapping Method							
Maps to Element							
Mapping Lookup Dimension							
Mapping Lookup Dimension Attribute							
Mapping Lookup Cube							
Mapping Lookup Cube Measure							

Acct
CostCenter
Prod
Version
Time
Value

If the source data is to be mapped to a 'default element', 'Maps to Source Column' may be left empty.

- **<u>Maps to Data Source Variable</u>**: automatically derived where applicable. Used as a lookup for the data load process. For cube data sources: VN for Dimension N; for ODBC and File Data Sources: VN for Column N according to data source configuration in '}ElementAttributes_TM1 Cube Data Source.cub'.

- **<u>Mapping Method</u>**:



Cube Viewer: Framework->TM1 Cube to Data Source Mapping->Cube to Cube Mapping with Source = Target Cube

File Edit View Options Help

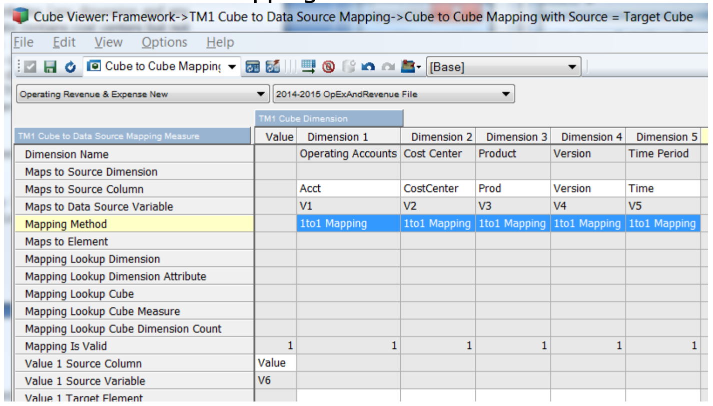
Cube to Cube Mapping [Base]

Operating Revenue & Expense New | 2014-2015 OpExAndRevenue File

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Din
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period	
Maps to Source Dimension							
Maps to Source Column		Acct	CostCenter				
Maps to Data Source Variable		V1					
Mapping Method		1to1 Mapping					
Maps to Element							
Mapping Lookup Dimension							
Mapping Lookup Dimension Attribute							
Mapping Lookup Cube							
Mapping Lookup Cube Measure							
Mapping Lookup Cube Dimension Count							

1to1 Mapping
Lookup
Default Element
maps to Data Source Value Column

- **<u>1to1 Mapping</u>**: Elements from data source directly correspond to elements in target dimension, i.e. are mapped 1 to 1. Note that if the data source is a cube and where the target and source cube dimensions are identical, the mapping method will be defaulted to '1to1 mapping' as in:



Cube Viewer: Framework->TM1 Cube to Data Source Mapping->Cube to Cube Mapping with Source = Target Cube

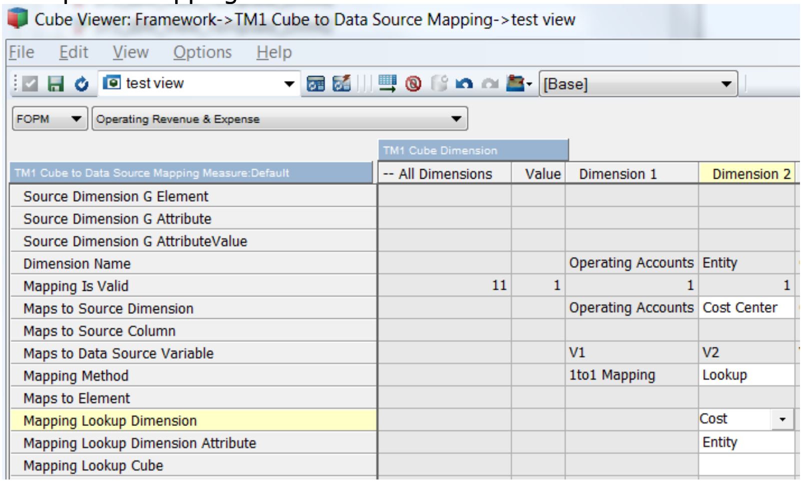
Cube to Cube Mapping · [Base] · Operating Revenue & Expense New · 2014-2015 OpExAndRevenue File

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period
Maps to Source Dimension						
Maps to Source Column		Acct	CostCenter	Prod	Version	Time
Maps to Data Source Variable		V1	V2	V3	V4	V5
Mapping Method		1to1 Mapping	1to1 Mapping	1to1 Mapping	1to1 Mapping	1to1 Mapping
Maps to Element						
Mapping Lookup Dimension						
Mapping Lookup Dimension Attribute						
Mapping Lookup Cube						
Mapping Lookup Cube Measure						
Mapping Lookup Cube Dimension Count						
Mapping Is Valid	1	1	1	1	1	1
Value 1 Source Column	Value					
Value 1 Source Variable	V6					
Value 1 Target Element						

- **<u>Lookup</u>**: mapping via a lookup using

  - **<u>Dimension attributes</u>** (any dimension and any attribute of that dimension). Example: your source data contains cost centers but not geography information. The target cube needs geography as a dimension. Geography is an attribute of Cost Center. => You can use the 'Lookup' mapping method to tell the load process to map Cost Center data to Geography via the Cost Center attribute 'Geo.', i.e. for each source record, TM1 will look up the Geography for the Cost Center from the source record via an element attribute lookup and then process the data to the geography.
    'Mapping Lookup Dimension' (Picklist) will allow using any TM1 dimension as a lookup and 'Mapping Lookup Dimension Attribute' will allow selecting any attribute from that dimension as the lookup attribute (the value of which TM1 shall use to determine the data target element(s)).

    Example of mapping via Cost Center attribute:

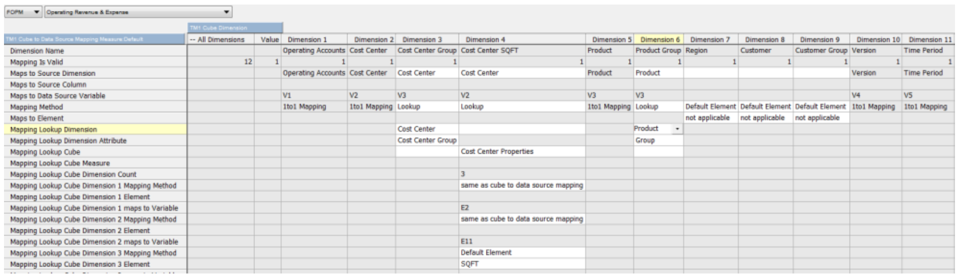


Cube Viewer: Framework->TM1 Cube to Data Source Mapping->test view

test view · [Base] · FOPM · Operating Revenue & Expense

TM1 Cube to Data Source Mapping Measure:Default	-- All Dimensions	Value	Dimension 1	Dimension 2
Source Dimension G Element				
Source Dimension G Attribute				
Source Dimension G AttributeValue				
Dimension Name			Operating Accounts	Entity
Mapping Is Valid	11	1	1	1
Maps to Source Dimension			Operating Accounts	Cost Center
Maps to Source Column				
Maps to Data Source Variable			V1	V2
Mapping Method			1to1 Mapping	Lookup
Maps to Element				
Mapping Lookup Dimension				Cost
Mapping Lookup Dimension Attribute				Entity
Mapping Lookup Cube				

    Example of mapping via Product Dimension Attribute:

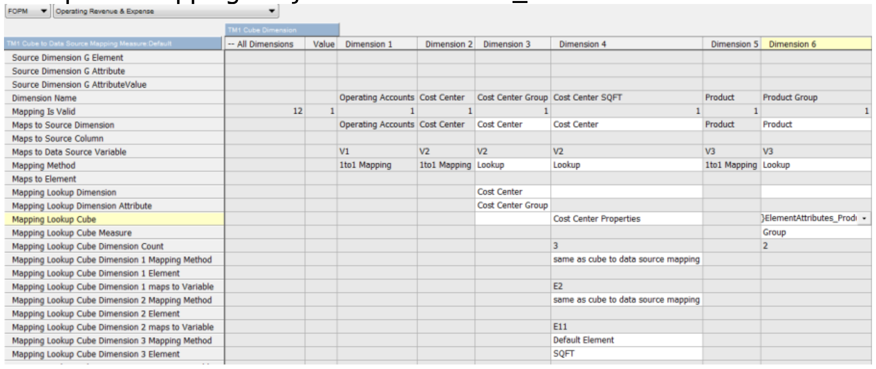


FOPM | Operating Revenue & Expense

TM1 Cube Dimension

TM1 Cube to Data Source Mapping Measure:Default	-- All Dimensions	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6	Dimension 7	Dimension 8	Dimension 9	Dimension 10	Dimension 11
Dimension Name			Operating Accounts	Cost Center	Cost Center Group	Cost Center SQFT	Product	Product Group	Region	Customer	Customer Group	Version	Time Period
Mapping Is Valid	12	1	1	1	1	1	1	1	1	1	1	1	1
Maps to Source Dimension			Operating Accounts	Cost Center	Cost Center	Cost Center	Product	Product				Version	Time Period
Maps to Source Column													
Maps to Data Source Variable			V1	V2	V3	V2	V3	V3				V4	V5
Mapping Method			1to1 Mapping	1to1 Mapping	Lookup	Lookup	1to1 Mapping	Lookup	Default Element	Default Element	Default Element	1to1 Mapping	1to1 Mapping
Maps to Element									not applicable	not applicable	not applicable		
Mapping Lookup Dimension					Cost Center			Product					
Mapping Lookup Dimension Attribute					Cost Center Group			Group					
Mapping Lookup Cube						Cost Center Properties							
Mapping Lookup Cube Measure													
Mapping Lookup Cube Dimension Count						3							
Mapping Lookup Cube Dimension 1 Mapping Method						same as cube to data source mapping							
Mapping Lookup Cube Dimension 1 Element													
Mapping Lookup Cube Dimension 1 maps to Variable						E2							
Mapping Lookup Cube Dimension 2 Mapping Method						same as cube to data source mapping							
Mapping Lookup Cube Dimension 2 Element													
Mapping Lookup Cube Dimension 2 maps to Variable						E11							
Mapping Lookup Cube Dimension 3 Mapping Method						Default Element							
Mapping Lookup Cube Dimension 3 Element						SQFT							

- **<u>2-dimensional Lookup Cubes</u>** (same as for dimension attributes, but not limited to attribute cubes).
  Example of mapping via }ElementAttributes_Product:

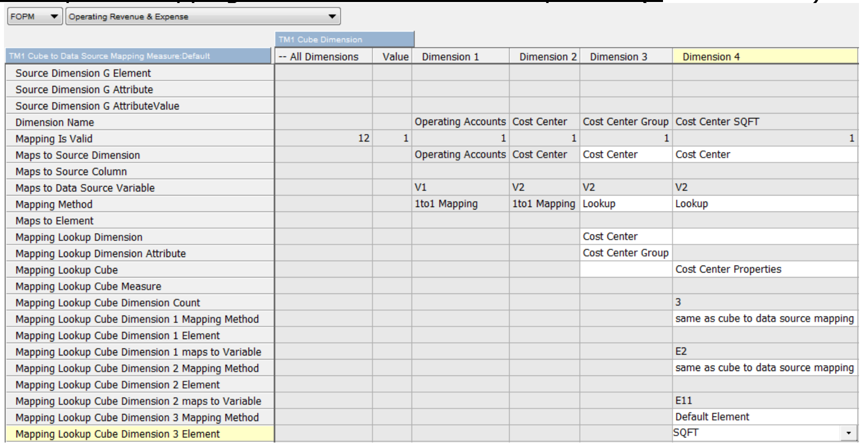


FOPM | Operating Revenue & Expense

TM1 Cube Dimension

TM1 Cube to Data Source Mapping Measure:Default	-- All Dimensions	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5	Dimension 6
Source Dimension G Element								
Source Dimension G Attribute								
Source Dimension G AttributeValue								
Dimension Name			Operating Accounts	Cost Center	Cost Center Group	Cost Center SQFT	Product	Product Group
Mapping Is Valid	12	1	1	1	1	1	1	1
Maps to Source Dimension			Operating Accounts	Cost Center	Cost Center	Cost Center	Product	Product
Maps to Source Column								
Maps to Data Source Variable			V1	V2	V2	V2	V3	V3
Mapping Method			1to1 Mapping	1to1 Mapping	Lookup	Lookup	1to1 Mapping	Lookup
Maps to Element								
Mapping Lookup Dimension					Cost Center			
Mapping Lookup Dimension Attribute					Cost Center Group			
Mapping Lookup Cube						Cost Center Properties		}ElementAttributes_Prod
Mapping Lookup Cube Measure								Group
Mapping Lookup Cube Dimension Count						3		2
Mapping Lookup Cube Dimension 1 Mapping Method						same as cube to data source mapping		
Mapping Lookup Cube Dimension 1 Element								
Mapping Lookup Cube Dimension 1 maps to Variable						E2		
Mapping Lookup Cube Dimension 2 Mapping Method						same as cube to data source mapping		
Mapping Lookup Cube Dimension 2 Element								
Mapping Lookup Cube Dimension 2 maps to Variable						E11		
Mapping Lookup Cube Dimension 3 Mapping Method						Default Element		
Mapping Lookup Cube Dimension 3 Element						SQFT		

- **<u>Multi-dimensional Lookup Cubes</u>** <u>(up to 12 Dimensions)</u>. Example: Example: Client Financials are to be merged with Risk data in such that the client data is to be assigned to a new dimension containing asset (instrument) risk categories (derived from risk data). It follows that the framework shall allow mapping the customer data to the risk category of the instrument for each month. In other words: Risk data is to be loaded first, then the client financials are to be mashed with the Financials via a lookup or each client record and associated instrument against the risk data, then assigning the instrument risk category to the client record. Because the risk data is instrument and time based (at a minimum) we have to look up the risk category against (i) instrument, (ii) time & (iii) risk metric. Consequently, a lookup against a cube with more than two dimensions is needed.
  <u>Example of mapping via cube 'Cost Center Properties' (3</u>-dimensional):

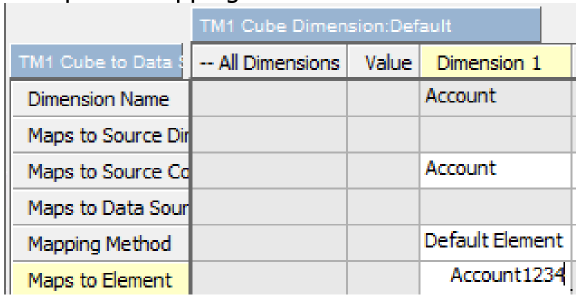


FOPM | Operating Revenue & Expense

TM1 Cube Dimension

TM1 Cube to Data Source Mapping Measure:Default	-- All Dimensions	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4
Source Dimension G Element						
Source Dimension G Attribute						
Source Dimension G AttributeValue						
Dimension Name			Operating Accounts	Cost Center	Cost Center Group	Cost Center SQFT
Mapping Is Valid	12	1	1	1	1	1
Maps to Source Dimension			Operating Accounts	Cost Center	Cost Center	Cost Center
Maps to Source Column						
Maps to Data Source Variable			V1	V2	V2	V2
Mapping Method			1to1 Mapping	1to1 Mapping	Lookup	Lookup
Maps to Element						
Mapping Lookup Dimension					Cost Center	
Mapping Lookup Dimension Attribute					Cost Center Group	
Mapping Lookup Cube						Cost Center Properties
Mapping Lookup Cube Measure						
Mapping Lookup Cube Dimension Count						3
Mapping Lookup Cube Dimension 1 Mapping Method						same as cube to data source mapping
Mapping Lookup Cube Dimension 1 Element						
Mapping Lookup Cube Dimension 1 maps to Variable						E2
Mapping Lookup Cube Dimension 2 Mapping Method						same as cube to data source mapping
Mapping Lookup Cube Dimension 2 Element						
Mapping Lookup Cube Dimension 2 maps to Variable						E11
Mapping Lookup Cube Dimension 3 Mapping Method						Default Element
Mapping Lookup Cube Dimension 3 Element						SQFT

- **<u>Default Element</u>**: use to map all of the source data to a specific element in the dimension. If Mapping Method = 'Default Element', the measure 'Maps to Element' will be available for input. Enter any valid N-Level Element from the target dimension. When mapping to a 'default element', 'Maps to Source Dimension' & 'Maps to Source Column' may be left empty.
  Example of mapping of all source data to account 'Account12324':

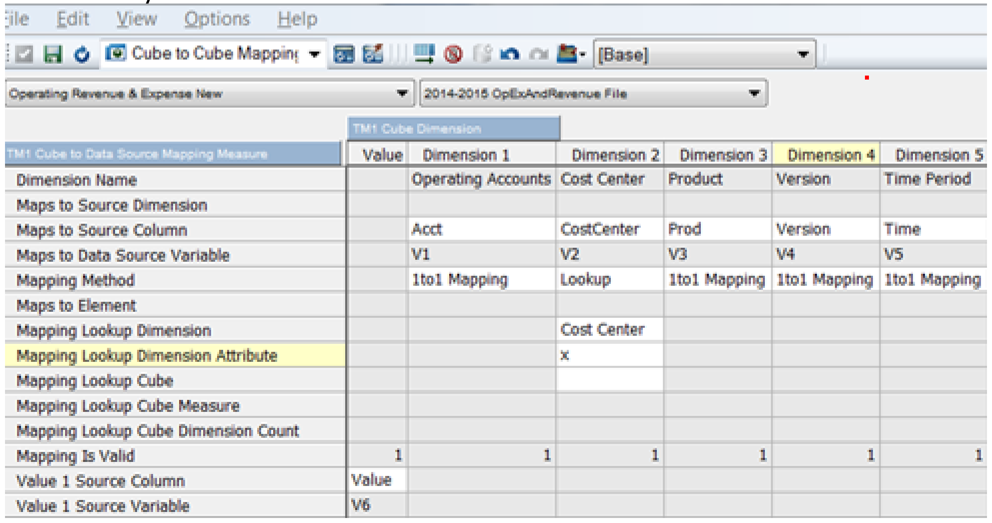


TM1 Cube to Data S	-- All Dimensions	Value	Dimension 1
Dimension Name			Account
Maps to Source Dir			
Maps to Source Co			Account
Maps to Data Sour			
Mapping Method			Default Element
Maps to Element			Account1234

- **<u>maps to Data Source Value Column</u>**: used to indicate that an entire fact data column is to be mapped to a specific element in the target dimension. See below on configuration for 'Value X Target Element' for more information on this type of source to target mapping.

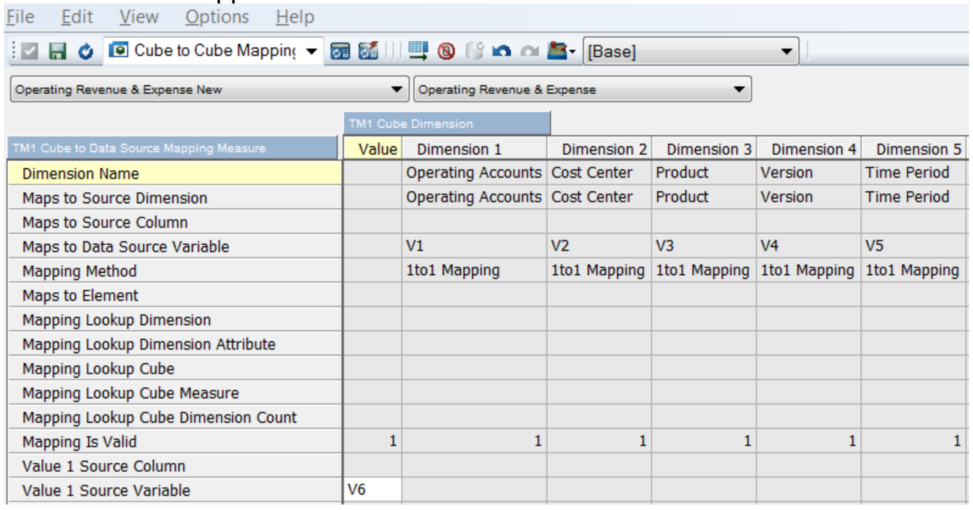
○ **<u>Mapping Is Valid</u>**: will indicate completeness of the mapping between target cube and data source. All target dimensions need to be mapped & at least one 'Value X Source Column' (if data source = ODBC or File) or 'Value X Source Variable' (for data source = cube) needs to be defined:

○ **<u>Value X Source Column</u>** (only applicable if data source = ODBC or File): pick the 1st data source column that contains numeric fact data. In the following example, the source column 'Value' is mapped to Value 1 and the corresponding variable V6 is determined automatically:


File Edit View Options Help
Cube to Cube Mapping
[Base]
Operating Revenue & Expense New
2014-2016 OpExAndRevenue File
TM1 Cube Dimension

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period
Maps to Source Dimension						
Maps to Source Column		Acct	CostCenter	Prod	Version	Time
Maps to Data Source Variable		V1	V2	V3	V4	V5
Mapping Method		1to1 Mapping	Lookup	1to1 Mapping	1to1 Mapping	1to1 Mapping
Maps to Element						
Mapping Lookup Dimension			Cost Center			
Mapping Lookup Dimension Attribute			x			
Mapping Lookup Cube						
Mapping Lookup Cube Measure						
Mapping Lookup Cube Dimension Count						
Mapping Is Valid	1	1	1	1	1	1
Value 1 Source Column	Value					
Value 1 Source Variable	V6					

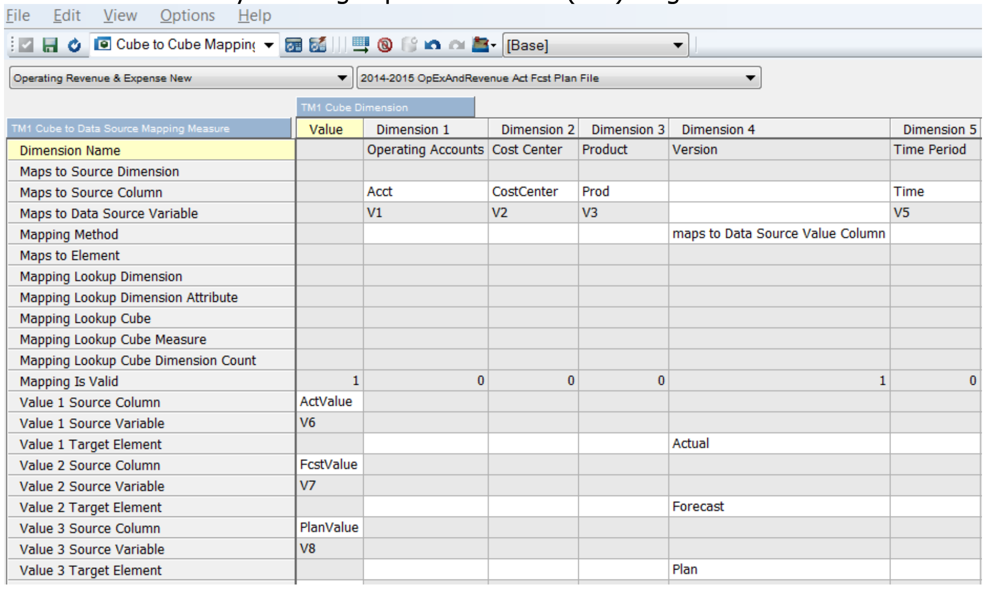
○ **<u>Value X Source Variable (with X = 1-6)</u>**: automatically determined based on 'Value X Source Column' entry if data source = ODBC or File. For Data Source = Cube, set to VN+1, with N = # of source cube dimensions. In other words: if the source cube has 9 dimensions, set Value X Source Variable to V10. In the following example, the source Variable 'V6' is mapped to Value:



File Edit View Options Help
Cube to Cube Mapping
[Base]
Operating Revenue & Expense New
Operating Revenue & Expense
TM1 Cube Dimension

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period
Maps to Source Dimension		Operating Accounts	Cost Center	Product	Version	Time Period
Maps to Source Column						
Maps to Data Source Variable		V1	V2	V3	V4	V5
Mapping Method		1to1 Mapping	1to1 Mapping	1to1 Mapping	1to1 Mapping	1to1 Mapping
Maps to Element						
Mapping Lookup Dimension						
Mapping Lookup Dimension Attribute						
Mapping Lookup Cube						
Mapping Lookup Cube Measure						
Mapping Lookup Cube Dimension Count						
Mapping Is Valid	1	1	1	1	1	1
Value 1 Source Column						
Value 1 Source Variable	V6					

- Value X Target Element (only applicable if data source = ODBC or File): optional, The Value X Target Element can be set for each dimension 1-N. if specified, the facts from the Value X Source Variable will all be processed the Target Element specified. Example: if one column in the SQL or Source File has Actuals, one would set Actuals as the Value X Target Element, hence ensuring that Actuals are loaded against version dimension element Actuals. Let's assume a data source contains 3 separate columns for Act (ActAmount = V6 = Column 6), FCST (FCSTAmount= V7 = Column 7) and Budget (PlanAmount = V8 = Column 8) Facts. Now we can map the 3 different facts to the version dimension by defining separate Value X (1-3) target elements:



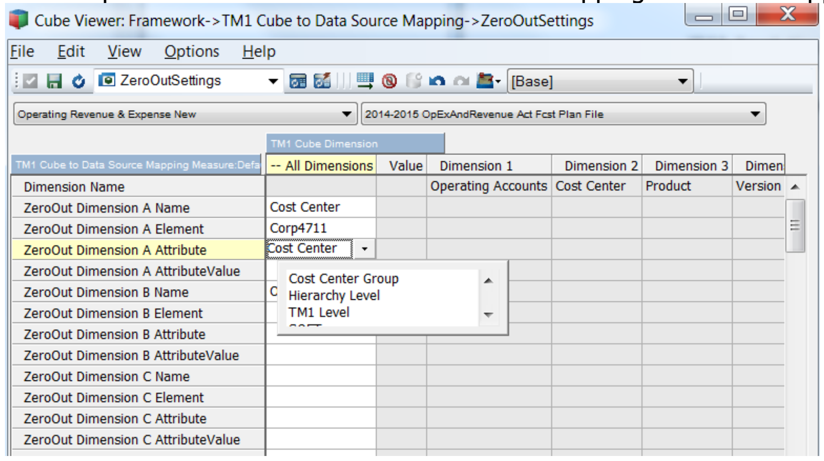
File Edit View Options Help
Cube to Cube Mapping
[Base]
Operating Revenue & Expense New
2014-2015 OpExAndRevenue Act Fcst Plan File
TM1 Cube Dimension

TM1 Cube to Data Source Mapping Measure	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period
Maps to Source Dimension						
Maps to Source Column		Acct	CostCenter	Prod		Time
Maps to Data Source Variable		V1	V2	V3		V5
Mapping Method					maps to Data Source Value Column	
Maps to Element						
Mapping Lookup Dimension						
Mapping Lookup Dimension Attribute						
Mapping Lookup Cube						
Mapping Lookup Cube Measure						
Mapping Lookup Cube Dimension Count						
Mapping Is Valid	1	0	0	0	1	0
Value 1 Source Column	ActValue					
Value 1 Source Variable	V6					
Value 1 Target Element					Actual	
Value 2 Source Column	FcstValue					
Value 2 Source Variable	V7					
Value 2 Target Element					Forecast	
Value 3 Source Column	PlanValue					
Value 3 Source Variable	V8					
Value 3 Target Element					Plan	

Note: once a 'Value X Target Element' is specified, mapping methods <> 'maps to Data Source Value Column' will be ignored, i.e. the 'Value X Target Element' will override other mappings.

2.3.1.3 ZeroOut-Defaults

- <u>ZeroOut Dimension X Name, ZeroOut Dimension X Element, ZeroOut Dimension X Attribute, ZeroOut Dimension X AttributeValue</u>: optional, allowing defining of default zero-out parameters for each cube to data source mapping; parameters defined in the TI process 'Manage Cube - Load or Update Data.pro' (see below) will override the defaults, but wherever no parameter is defined in 'Manage Cube - Load or Update Data.pro', the default specified in 'TM1 Cube to Data Source Mapping.cub' will be applied':



Cube Viewer: Framework->TM1 Cube to Data Source Mapping->ZeroOutSettings

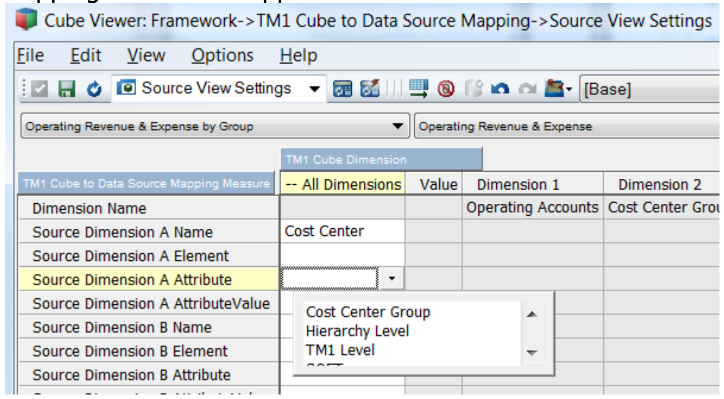
ZeroOutSettings | [Base]

Operating Revenue & Expense New | 2014-2015 OpExAndRevenue Act Fcst Plan File

TM1 Cube to Data Source Mapping Measure:Defa	-- All Dimensions	Value	Dimension 1	Dimension 2	Dimension 3	Dimen
Dimension Name			Operating Accounts	Cost Center	Product	Version
ZeroOut Dimension A Name	Cost Center					
ZeroOut Dimension A Element	Corp4711					
ZeroOut Dimension A Attribute	Cost Center					
ZeroOut Dimension A AttributeValue						
ZeroOut Dimension B Name	C					
ZeroOut Dimension B Element						
ZeroOut Dimension B Attribute						
ZeroOut Dimension B AttributeValue						
ZeroOut Dimension C Name						
ZeroOut Dimension C Element						
ZeroOut Dimension C Attribute						
ZeroOut Dimension C AttributeValue						

2.3.1.4 Cube Source View Defaults (for cube data sources)

- <u>Source Dimension X Name, Source Dimension X Element, Source Dimension X Attribute, Source Dimension X AttributeValue</u>: optional, allowing defining of default source view parameters for data sources of type 'Cube' & for each target cube to data source mapping; parameters defined in the TI process 'Manage Cube - Load or Update Data.pro' (see below) will override the defaults, but wherever no parameter is defined in 'Manage Cube - Load or Update Data.pro', the default specified in 'TM1 Cube to Data Source Mapping.cub' will be applied':



Cube Viewer: Framework->TM1 Cube to Data Source Mapping->Source View Settings

Source View Settings | [Base]

Operating Revenue & Expense by Group | Operating Revenue & Expense

TM1 Cube to Data Source Mapping Measure	-- All Dimensions	Value	Dimension 1	Dimension 2
Dimension Name			Operating Accounts	Cost Center Grou
Source Dimension A Name	Cost Center			
Source Dimension A Element				
Source Dimension A Attribute				
Source Dimension A AttributeValue				
Source Dimension B Name				
Source Dimension B Element				
Source Dimension B Attribute				

- Source View is C-level (Y/N): optional; if set to Y will allow creating a C-Level (consolidation) source view instead of the (default) leaf level view

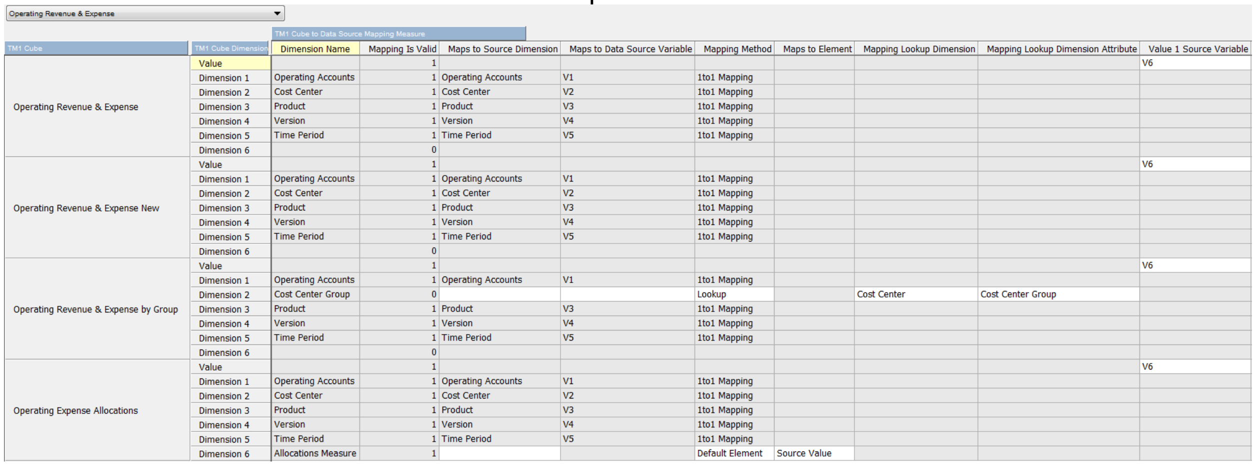
2.3.1.5 Other

- Increment or Put: optional; default if empty = Increment; allows to specify a default data write behavior for each mapping with Increment = increment values & Put = Overwrite with last record.

2.3.2 Cube to Data Source Mapping Examples

2.3.2.1 Cube to Cube Mapping

In the following mapping examples, most mappings are performed automatically because the source cube (‘Operating Revenue & Expense) dimensions correspond with the target cube dimensions. Note: the dimension order is irrelevant, i.e. (auto-) mapping is performed regardless of the dimension sort order. For the Allocations cube, we are mapping all data to one measure called ‘Source Value’. For the cube ‘Operating Revenue & Expense by Group’, we are mapping the data to the Cost Center Groups via the Cost Center dimension attribute ‘Cost Center Group’:

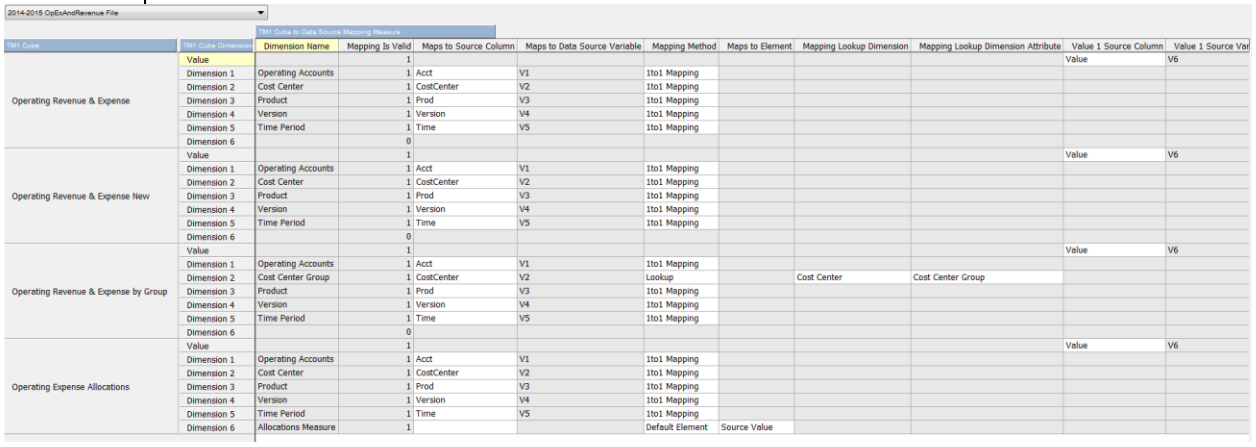


Operating Revenue & Expense

		TM1 Cube to Data Source Mapping Measure								
TM1 Cube	TM1 Cube Dimension	Dimension Name	Mapping Is Valid	Maps to Source Dimension	Maps to Data Source Variable	Mapping Method	Maps to Element	Mapping Lookup Dimension	Mapping Lookup Dimension Attribute	Value 1 Source Variable
Operating Revenue & Expense	Value		1							V6
	Dimension 1	Operating Accounts	1	Operating Accounts	V1	1to1 Mapping				
	Dimension 2	Cost Center	1	Cost Center	V2	1to1 Mapping				
	Dimension 3	Product	1	Product	V3	1to1 Mapping				
	Dimension 4	Version	1	Version	V4	1to1 Mapping				
	Dimension 5	Time Period	1	Time Period	V5	1to1 Mapping				
	Dimension 6		0							
Operating Revenue & Expense New	Value		1							V6
	Dimension 1	Operating Accounts	1	Operating Accounts	V1	1to1 Mapping				
	Dimension 2	Cost Center	1	Cost Center	V2	1to1 Mapping				
	Dimension 3	Product	1	Product	V3	1to1 Mapping				
	Dimension 4	Version	1	Version	V4	1to1 Mapping				
	Dimension 5	Time Period	1	Time Period	V5	1to1 Mapping				
	Dimension 6		0							
Operating Revenue & Expense by Group	Value		1							V6
	Dimension 1	Operating Accounts	1	Operating Accounts	V1	1to1 Mapping				
	Dimension 2	Cost Center Group	0			Lookup		Cost Center	Cost Center Group	
	Dimension 3	Product	1	Product	V3	1to1 Mapping				
	Dimension 4	Version	1	Version	V4	1to1 Mapping				
	Dimension 5	Time Period	1	Time Period	V5	1to1 Mapping				
	Dimension 6		0							
Operating Expense Allocations	Value		1							V6
	Dimension 1	Operating Accounts	1	Operating Accounts	V1	1to1 Mapping				
	Dimension 2	Cost Center	1	Cost Center	V2	1to1 Mapping				
	Dimension 3	Product	1	Product	V3	1to1 Mapping				
	Dimension 4	Version	1	Version	V4	1to1 Mapping				
	Dimension 5	Time Period	1	Time Period	V5	1to1 Mapping				
	Dimension 6	Allocations Measure	1			Default Element	Source Value			

2.3.2.2 File or SQL to Cube Mapping (one source value column)

In the following mapping examples, most mappings are 1to1 mappings because the source file data corresponds 1:1 with the target cube dimensions. Note: here too the dimension order and variable order are irrelevant, i.e. mapping is performed regardless of the dimension sort order, dimension 1 can be mapped to V1-N. We are mapping the source file to the target cube by determining which source file column maps to which target cube dimension. Furthermore - like above - for the Allocations cube, we are mapping all data to one measure called ‘Source Value’. For the cube ‘Operating Revenue & Expense by Group’, we are mapping the data to the Cost Center Groups via the Cost Center dimension attribute ‘Cost Center Group’:

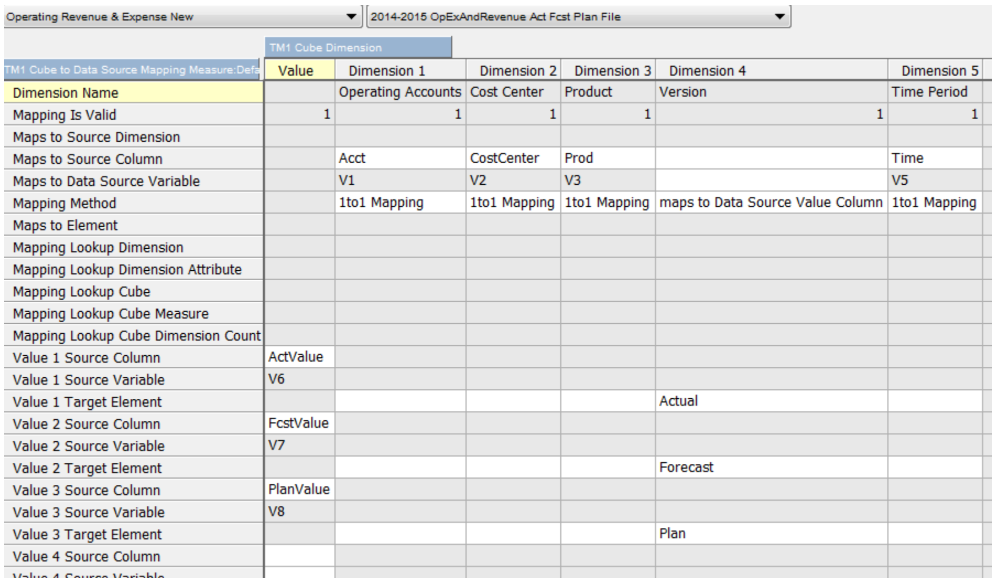


2014-2015 OpExAndRevenue File

		TM1 Cube to Data Source Mapping Measure									
TM1 Cube	TM1 Cube Dimension	Dimension Name	Mapping Is Valid	Maps to Source Column	Maps to Data Source Variable	Mapping Method	Maps to Element	Mapping Lookup Dimension	Mapping Lookup Dimension Attribute	Value 1 Source Column	Value 1 Source Var
Operating Revenue & Expense	Value		1							Value	V6
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping					
	Dimension 2	Cost Center	1	CostCenter	V2	1to1 Mapping					
	Dimension 3	Product	1	Prod	V3	1to1 Mapping					
	Dimension 4	Version	1	Version	V4	1to1 Mapping					
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping					
	Dimension 6		0								
Operating Revenue & Expense New	Value		1							Value	V6
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping					
	Dimension 2	Cost Center	1	CostCenter	V2	1to1 Mapping					
	Dimension 3	Product	1	Prod	V3	1to1 Mapping					
	Dimension 4	Version	1	Version	V4	1to1 Mapping					
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping					
	Dimension 6		0								
Operating Revenue & Expense by Group	Value		1							Value	V6
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping					
	Dimension 2	Cost Center Group	1	CostCenter	V2	Lookup		Cost Center	Cost Center Group		
	Dimension 3	Product	1	Prod	V3	1to1 Mapping					
	Dimension 4	Version	1	Version	V4	1to1 Mapping					
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping					
	Dimension 6		0								
Operating Expense Allocations	Value		1							Value	V6
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping					
	Dimension 2	Cost Center	1	CostCenter	V2	1to1 Mapping					
	Dimension 3	Product	1	Prod	V3	1to1 Mapping					
	Dimension 4	Version	1	Version	V4	1to1 Mapping					
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping					
	Dimension 6	Allocations Measure	1			Default Element	Source Value				

2.3.2.3 File or SQL to Cube Mapping (multiple source value columns)

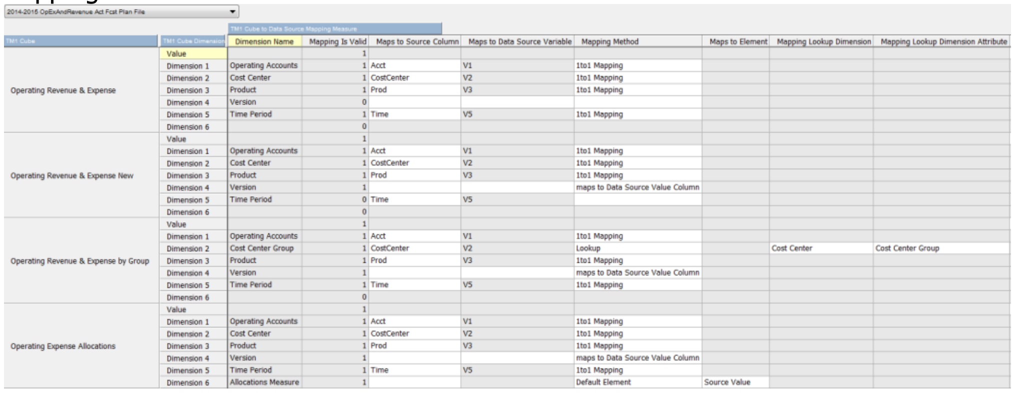
Except for dimension Version and for the value mapping, the mapping is as above. Our source file contains three columns with values for (1) Actuals, (2) Forecast & (3) Plan. We hence need to configure the mapping such that (A) all 3 columns are understood as 'value' columns and such that (B) the values are mapped/loaded against their corresponding elements in dimension Version:



Operating Revenue & Expense New | 2014-2015 OpExAndRevenue Act Fcst Plan File

TM1 Cube to Data Source Mapping Measure:Defa	Value	Dimension 1	Dimension 2	Dimension 3	Dimension 4	Dimension 5
Dimension Name		Operating Accounts	Cost Center	Product	Version	Time Period
Mapping Is Valid	1	1	1	1	1	1
Maps to Source Dimension						
Maps to Source Column		Acct	CostCenter	Prod		Time
Maps to Data Source Variable		V1	V2	V3		V5
Mapping Method		1to1 Mapping	1to1 Mapping	1to1 Mapping	maps to Data Source Value Column	1to1 Mapping
Maps to Element						
Mapping Lookup Dimension						
Mapping Lookup Dimension Attribute						
Mapping Lookup Cube						
Mapping Lookup Cube Measure						
Mapping Lookup Cube Dimension Count						
Value 1 Source Column	ActValue					
Value 1 Source Variable	V6					
Value 1 Target Element					Actual	
Value 2 Source Column	FcstValue					
Value 2 Source Variable	V7					
Value 2 Target Element					Forecast	
Value 3 Source Column	PlanValue					
Value 3 Source Variable	V8					
Value 3 Target Element					Plan	
Value 4 Source Column						

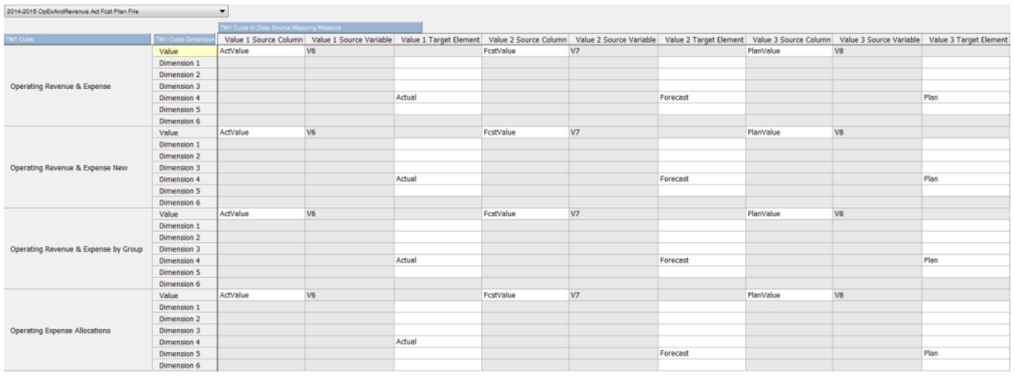
Mapping for all cubes:



2014-2015 OpExAndRevenue Act Fcst Plan File

TM1 Cube	TM1 Cube Dimension	Dimension Name	Mapping Is Valid	Maps to Source Column	Maps to Data Source Variable	Mapping Method	Maps to Element	Mapping Lookup Dimension	Mapping Lookup Dimension Attribute
Operating Revenue & Expense	Value		1						
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping			
	Dimension 2	Cost Center	1	CostCenter	V2	1to1 Mapping			
	Dimension 3	Product	1	Prod	V3	1to1 Mapping			
	Dimension 4	Version	0						
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping			
	Dimension 6		0						
Operating Revenue & Expense New	Value		1						
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping			
	Dimension 2	Cost Center	1	CostCenter	V2	1to1 Mapping			
	Dimension 3	Product	1	Prod	V3	1to1 Mapping			
	Dimension 4	Version	1			maps to Data Source Value Column			
	Dimension 5	Time Period	0	Time	V5				
	Dimension 6		0						
Operating Revenue & Expense by Group	Value		1						
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping			
	Dimension 2	Cost Center Group	1	CostCenter	V2	Lookup		Cost Center	Cost Center Group
	Dimension 3	Product	1	Prod	V3	1to1 Mapping			
	Dimension 4	Version	1			maps to Data Source Value Column			
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping			
	Dimension 6		0						
Operating Expense Allocations	Value		1						
	Dimension 1	Operating Accounts	1	Acct	V1	1to1 Mapping			
	Dimension 2	Cost Center	1	CostCenter	V2	1to1 Mapping			
	Dimension 3	Product	1	Prod	V3	1to1 Mapping			
	Dimension 4	Version	1			maps to Data Source Value Column			
	Dimension 5	Time Period	1	Time	V5	1to1 Mapping			
	Dimension 6	Allocations Measure	1			Default Element	Source Value		

&



2014-2015 OpExAndRevenue Act Fcst Plan File

TM1 Cube	TM1 Cube Dimension	Value 1 Source Column	Value 1 Source Variable	Value 1 Target Element	Value 2 Source Column	Value 2 Source Variable	Value 2 Target Element	Value 3 Source Column	Value 3 Source Variable	Value 3 Target Element
Operating Revenue & Expense	Value	ActValue	V6		FcstValue	V7		PlanValue	V8	
	Dimension 1									
	Dimension 2									
	Dimension 3									
	Dimension 4			Actual			Forecast			Plan
	Dimension 5									
	Dimension 6									
Operating Revenue & Expense New	Value	ActValue	V6		FcstValue	V7		PlanValue	V8	
	Dimension 1									
	Dimension 2									
	Dimension 3									
	Dimension 4			Actual			Forecast			Plan
	Dimension 5									
	Dimension 6									
Operating Revenue & Expense by Group	Value	ActValue	V6		FcstValue	V7		PlanValue	V8	
	Dimension 1									
	Dimension 2									
	Dimension 3									
	Dimension 4			Actual			Forecast			Plan
	Dimension 5									
	Dimension 6									
Operating Expense Allocations	Value	ActValue	V6		FcstValue	V7		PlanValue	V8	
	Dimension 1									
	Dimension 2									
	Dimension 3									
	Dimension 4			Actual						
	Dimension 5						Forecast			Plan
	Dimension 6									

2.4 Cube Data Load & Update

2.4.1 Generic Cube Load/Update process

Once a data source is mapped to a cube, the process '**Manage Cube - Load or Update Data.pro**' is used to update/load the data as per Cube configuration, Data Source Configuration and Cube to Data Source Mapping. A cube can be mapped to any number of data sources.

Versions of this process as of 1/2020:

a) **Manage Cube - Load or Update Data - 16Dim.pro:** original process
b) **Manage Cube - Load or Update Data - 23Dim - V2.pro**: original process, yet enhanced for 23 dimensions)
c) **Manage Cube - Load or Update Data - 23Dim - V2 - C2C.pro**: >10%faster process than (b) to use for Cube-to-Cube processing or if files/SQL load records contain only one column with numeric values)
d) **Manage Cube - Load or Update Data - 23Dim - V2 - C2C - Basic.pro**: >10%faster process than (c) to use for Cube-to-Cube processing or if files/SQL load records contain only one column with numeric values and data mapping does not require lookup of a 3rd, cube with 3-12 dimensions)
e) **Manage Cube - Load or Update Data - 23Dim - V2 - C2C - Basic - NoMissingELementUpdate.pro**: >10%faster process than (d) to use for Cube-to-Cube processing or if files/SQL load records contain only one column with numeric values and data mapping does not require lookup of a 3rd, cube with 3-12 dimensions, and if process does not need to create missing elements)

Parameters of process 'Manage Cube - Load or Update Data.pro':

- pCube: Target Cube = Cube as per configuration in 'TM1 Cube to Data Source Mapping.cub'
- pCubeDataSource: Data Source = Data Source as per configuration in 'TM1 Cube to Data Source Mapping.cub'
- pZeroOutTarget: if Y, the target cube will be zeroed out prior to update/load according to the parameter values pZeroOut_DimensionX_Name, pZeroOut_DimensionX_Element, pZeroOut_DimensionX_Attribute, pZeroOut_DimensionX_AttributeValue (see below). If no parameter values are specified for the pZeroOut_DimensionX_* parameters and if pZeroOutTarget = Y, the entire cube will be zeroed-out.
- pAddElements: default = N; set to Y to insert/create missing dimension elements under the 'orphans' node specified in 'SYS_IBM_Control.cub'; recommendation: leave at N and update master data separately as this will allow faster data load.
- pUpdateData: Default = Y, set to Y for testing purposes or zero-out operations only.

With x = A-F: Zero-Out parameters/options[3]

- pZeroOut_DimensionX_Name: name of a dimension in the target cube for which to apply a specific zero-out
- pZeroOut_DimensionX_Element: name of an element, group of elements, element specification (corresponding to the above dimension) for which to apply a specific zero-out. See below information on 'Element Specification Options' for details
- pZeroOut_DimensionX_Attribute: name of an element attribute to use for the zero-out filter
- pZeroOut_DimensionX_AttributeValue: element attribute value to use for the zero-out operation

With x = A-G & only if data source = cube[4]

[3] Note: Wherever no parameter is defined in 'Manage Cube - Load or Update Data.pro', the defaults specified in 'TM1 Cube to Data Source Mapping.cub' will be applied'. See section <ZeroOut-Defaults> for details.

- pSource_DimensionX_Name: name of a dimension in the source cube from which to process data only for certain elements
- pSource_DimensionX_Element: name of an element, group of elements, element specification (corresponding to the above dimension) for which data is to be queried / processed from the source cube. See below information on 'Element Specification Options' for details
- pSource_DimensionX_Attribute: name of an element attribute to use for determining source data elements/intersections
- pSource_DimensionX_AttributeValue: element attribute value to use for determining the source data elements/intersections

- pSourceViewIsCLevel: set to Y if the source view is at a C-Level / contains C-Level elements. If set at N or left at '' (empty, with default N), only N-Level views are created

[4] Note: Wherever no parameter is defined in 'Manage Cube - Load or Update Data.pro', the defaults specified in 'TM1 Cube to Data Source Mapping.cub' will be applied'. See section <Cube Source View Defaults > for details.

Debugging:

- pLogging: only set to Y for detailed debugging (very large log files will be created); leave empty (default – NB) or at N for normal operation.
- pRecordLimitForDebug: leave at 0 for regular (complete) processing; if set to a value >0, the process will terminate processing once the # of records = RecordLimitForDebut have been processed. The process will be flagged with an error. Set to >0 for debugging only.

Other:

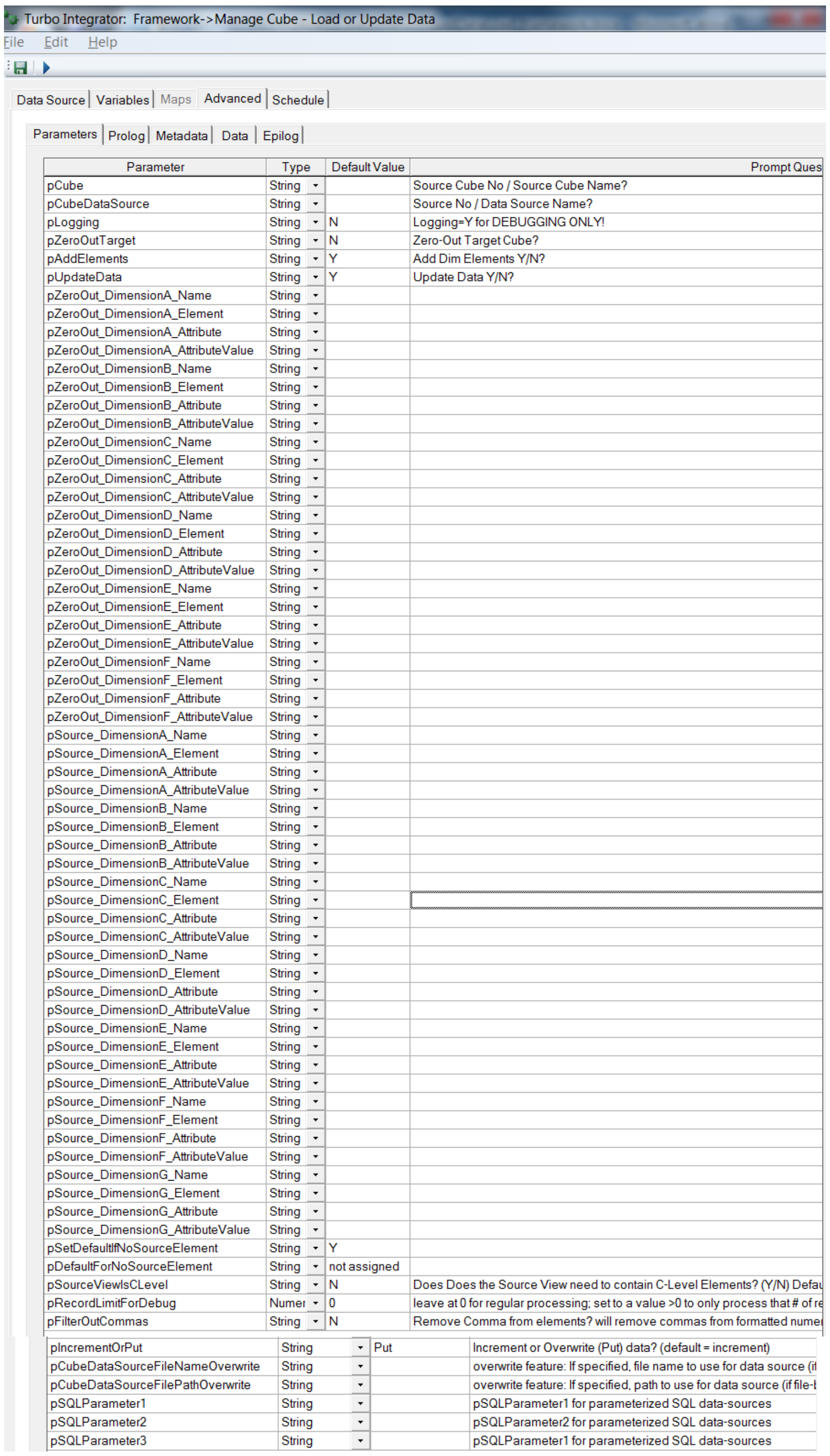
- pFilterOutCommas: Y/N; => set to Y if numeric columns from the SQL db are formatted to include a comma separator and to remove this comma; example: Time Period 201,101.00 will not be converted to 201,101 but to 201101.
- pSetDefaultIfNoSourceElement: Set fo Y if the data source may contain source column values that are empty and shall be assigned to a specific element name. If set to Y, the element name defined in pSetDefaultIfNoSourceElement (below) will be used.
- pSetDefaultIfNoSourceElement: default element name to use if data source element is empty
- pIncrementOrPut: Increment data or overwrite (Put) data? (If empty, then default = Increment)
- pCubeDataSourceFileNameOverwrite: overwrite feature; if specified, file name to use for data source (if file-based data source) instead of file name defined with data source
- pCubeDataSourceFilePathOverwrite: overwrite feature: If specified, path to use for data source (if file-based data source)
- pSQLParameter1: pSQLParameter1 for parameterized SQL data-sources; If the SQL Statement for the ODBC data source is modified like
  Select … from … where … = '%pSQLParameter1%';
  the corresponding parameter will be inserted into the SQL query at process runtime.
- pSQLParameter2: pSQLParameter2 for parameterized SQL data-sources
- pSQLParameter3: pSQLParameter3 for parameterized SQL data-sources

Element Specification Options: One may use the following entries and/or prefixes in pZeroOut_DimensionX_Element and/or pSource_DimensionX_Element:

- Element Name Parameter Values
  - <ElementName> = single Element Name
    - ⇨ filter view by element name
  - Value =* or All:
    - ⇨ SubsetIsAllSet = all elements in dimension, filtered by attribute value if specified
  - Value = '' (empty):
    - ⇨ all N-level Elements (DEFAULT), filtered by attribute value if specified

- Element Name Parameter Value with Prefix (<Prefix><ElementName>)[5]:
  - A;<ElementName> or AD;<ElementName>
    - ⇨ All Descendants, filtered by attribute value if specified
  - ND;<ElementName>
    - ⇨ All N-Level Descendants (excluding Parent), filtered by attribute value if specified
  - IC;<ElementName>
    - ⇨ Immediate Children, filtered by attribute value if specified
  - CD;<ElementName>
    - ⇨ C-Level Descendants (including Parent), filtered by attribute value if specified
  - Multi:<ElementName1>;<ElementName2>;...;<ElementNameN>
    - ⇨ Multiple Elements, filtered by attribute value if specified

[5] The same process parameters for element query specifications ('A;*','ND;*'IC;8',…) apply to processes 'SYS_IBM_View_Create.pro', 'SYS_IBM_Subset_Create.pro', 'SYS_IBM_Data_Processing_-_File_Export.pro'


Turbo Integrator: Framework->Manage Cube - Load or Update Data
File Edit Help
Data Source | Variables | Maps | Advanced | Schedule
Parameters | Prolog | Metadata | Data | Epilog

Parameter	Type	Default Value	Prompt Ques
pCube	String		Source Cube No / Source Cube Name?
pCubeDataSource	String		Source No / Data Source Name?
pLogging	String	N	Logging=Y for DEBUGGING ONLY!
pZeroOutTarget	String	N	Zero-Out Target Cube?
pAddElements	String	Y	Add Dim Elements Y/N?
pUpdateData	String	Y	Update Data Y/N?
pZeroOut_DimensionA_Name	String		
pZeroOut_DimensionA_Element	String		
pZeroOut_DimensionA_Attribute	String		
pZeroOut_DimensionA_AttributeValue	String		
pZeroOut_DimensionB_Name	String		
pZeroOut_DimensionB_Element	String		
pZeroOut_DimensionB_Attribute	String		
pZeroOut_DimensionB_AttributeValue	String		
pZeroOut_DimensionC_Name	String		
pZeroOut_DimensionC_Element	String		
pZeroOut_DimensionC_Attribute	String		
pZeroOut_DimensionC_AttributeValue	String		
pZeroOut_DimensionD_Name	String		
pZeroOut_DimensionD_Element	String		
pZeroOut_DimensionD_Attribute	String		
pZeroOut_DimensionD_AttributeValue	String		
pZeroOut_DimensionE_Name	String		
pZeroOut_DimensionE_Element	String		
pZeroOut_DimensionE_Attribute	String		
pZeroOut_DimensionE_AttributeValue	String		
pZeroOut_DimensionF_Name	String		
pZeroOut_DimensionF_Element	String		
pZeroOut_DimensionF_Attribute	String		
pZeroOut_DimensionF_AttributeValue	String		
pSource_DimensionA_Name	String		
pSource_DimensionA_Element	String		
pSource_DimensionA_Attribute	String		
pSource_DimensionA_AttributeValue	String		
pSource_DimensionB_Name	String		
pSource_DimensionB_Element	String		
pSource_DimensionB_Attribute	String		
pSource_DimensionB_AttributeValue	String		
pSource_DimensionC_Name	String		
pSource_DimensionC_Element	String		
pSource_DimensionC_Attribute	String		
pSource_DimensionC_AttributeValue	String		
pSource_DimensionD_Name	String		
pSource_DimensionD_Element	String		
pSource_DimensionD_Attribute	String		
pSource_DimensionD_AttributeValue	String		
pSource_DimensionE_Name	String		
pSource_DimensionE_Element	String		
pSource_DimensionE_Attribute	String		
pSource_DimensionE_AttributeValue	String		
pSource_DimensionF_Name	String		
pSource_DimensionF_Element	String		
pSource_DimensionF_Attribute	String		
pSource_DimensionF_AttributeValue	String		
pSource_DimensionG_Name	String		
pSource_DimensionG_Element	String		
pSource_DimensionG_Attribute	String		
pSource_DimensionG_AttributeValue	String		
pSetDefaultIfNoSourceElement	String	Y	
pDefaultForNoSourceElement	String	not assigned	
pSourceViewIsCLevel	String	N	Does Does the Source View need to contain C-Level Elements? (Y/N) Defau
pRecordLimitForDebug	Numer	0	leave at 0 for regular processing; set to a value >0 to only process that # of re
pFilterOutCommas	String	N	Remove Comma from elements? will remove commas from formatted nume
pIncrementOrPut	String	Put	Increment or Overwrite (Put) data? (default = increment)
pCubeDataSourceFileNameOverwrite	String		overwrite feature: If specified, file name to use for data source (i
pCubeDataSourceFilePathOverwrite	String		overwrite feature: If specified, path to use for data source (if file-l
pSQLParameter1	String		pSQLParameter1 for parameterized SQL data-sources
pSQLParameter2	String		pSQLParameter2 for parameterized SQL data-sources
pSQLParameter3	String		pSQLParameter1 for parameterized SQL data-sources

2.5 Performance Optimized ‘Custom’ load process template

Process ‘Manage Cube - Load or Update Data.pro’ is generic and uses advanced TM1 Turbo integrator logic and algorithms to assign/map source data to the target cube as defined in the mapping model. The resulting generic code takes approximately 5x longer to run than a custom load process that was tailored to cube and data source. For many TM1 cubes, data sources & data update requirements this additional run time ‘penalty’ is insignificant (if smaller cubes or even larger cubes are not updated frequently). If very large cubes however are to be updated frequently (for example cubes with millions or even billions of facts that are updated on a daily basis) it may be better to build a custom load process. For such cases – where a custom load process is desired to optimize load performance – a custom load TI-process template/sample called ‘**Cube X - Data Load or Update – Template.pro**’ is included with this framework. The TI-process template provides skilled TM1 developers to create a custom, performance optimized TI data load process within minutes. The custom load process template leverages
- The included utilities for View & Subset creation and hence provides the developer and user with the same flexible options for zero-out and source view (if applicable) creation as ‘Manage Cube - Load or Update Data.pro’
- the data source configuration information in ‘}ElementAttributes_TM1 Cube Data Source.cub’ to determine if the data source is File, Cube or ODBC and in the prolog will assign the corresponding load parameters accordingly. Cube to data source mapping information however is ignored. In the provided template, we use sample code to map cube ‘Operating Revenue & Expense New’ to data soruce ‘2014-2015 OpExAndRevenue File’. Variable mappings are custom to the source file and the target cube.

For as long as the data source is configured in ‘}ElementAttributes_TM1 Cube Data Source.cub’, the only customization to TI process ‘**Cube X - Data Load or Update – Template.pro**’ needs to occur in the data tab:

```
nRecordCounter = nRecordCounter + 1;
IF ( pRecordLimitForDebug > 0 );
  IF ( nRecordCounter > pRecordLimitForDebug );
    sDebugFlag = 'process terminated after '| numbertostring  (nRecordCounter) | ' records as per variable pRecordLimitForDebug!' ;
    processbreak;
  ENDIF;
ENDIF;

IF ( pAddElements @= 'Y' );
#####Customization Instruction: Add below block per Dimension; adjust Variable name depending on source
       #Start: add elements to Dimension 1#
       IF ( DIMIX ( sDim1, V1 ) = 0 );
         DimensionElementInsertDirect ( sDim1,'',  V1 , 'N'  );
         ASCIIOUTPUT ( LogFileData, 'added dimension element', V1, 'to dimension', sDim1);
         IF ( DIMIX ( sDim1, sOrphansNode ) > 0 );
           DimensionElementComponentAddDirect ( sDim1, sOrphansNode,  V1 , 1 );
           ASCIIOUTPUT ( LogFileData, 'added dimension element', V1, 'to dimension', sDim1, 'parent', sOrphansNode);
         ENDIF;
       ENDIF;
       #END: add elements to Dimension 1#
ENDIF;

IF ( pUpdateData @= 'Y' );
#####Customization Instructiion: Adjust below block depending on data source dimesionality and  content; un-comment optional code if desired
#####Notes:
#####a) remove check and related IF/THEN logic for CellIsUpdateable when loading very large data volumes (yet ensure that only permissible data is contained in the data source)
#####b) remove check and related IF/THEN logic DTYPE N or S if only one data type is loaded.
       IF ( CellIsUpDateable ( sCubeName,V1,V2,V3,V4,V5) = 0 );
           IF ( pLogging @= 'Y' );
             ASCIIOUTPUT ( LogFileData, 'The following Intersection is not Updateable:', sCubeName,V1,V2,V3,V4,V5);
             nDataErrorFlag = nDataErrorFlag + 1;
          ENDIF;
       ELSE;
          IF ( DTYPE ( sDim5, V5) @= 'N' );
             nValue = NUMBR ( V6 );
             CellIncrementN ( nValue, sCubeName,V1,V2,V3,V4,V5);
          ELSEIF ( DTYPE ( sDim5, V5) @= 'S' );
             CellPutS ( V6, sCubeName,V1,V2,V3,V4,V5);
          ENDIF;
       ENDIF;
ENDIF;
```

If the data source is not configured in ‘}ElementAttributes_TM1 Cube Data Source.cub’, the following section of the TI process prolog tab needs to be modified (parameters that would need to be customized in this case are highlighted in red):

```
### Data Source = File
~~IF ( ATTRS ( 'TM1 Cube Data Source', pCubeDataSource, 'Type' ) @= 'File');~~
  sFileName = <FILENAME>
  sPath =  <FILEPATH=DIRECTORY Incl. last backslash>
~~  sTM1DataSourceType = ATTRS ( 'TM1 Cube Data Source', pCubeDataSource, 'TM1 TI Data Source Type' );~~
~~  sTM1DataSourceASCIIDelimiter = ATTRS ( 'TM1 Cube Data Source', pCubeDataSource, 'TM1 TI Data Source ASCII Delimiter' );~~
~~  nTM1DataSourceHeaderRecords = ATTRN ( 'TM1 Cube Data Source', pCubeDataSource, 'TM1 TI Data Source ASCII Header Records' );~~
  DataSourceNameForServer = sPath | sFileName ;
  DataSourceNameForClient = sPath | sFileName ;
  DataSourceType = <DataSourceType as per source file and TM1 documentation>;
  DatasourceASCIIDelimiter = <DatasourceASCIIDelimiter as per source file and TM1 documentation>;
  DatasourceASCIIHeaderRecords=<# of header records>;
~~  IF ( pLogging @= 'Y' );~~
~~    ASCIIOUTPUT ( LogFileProlog, 'Data Source Config:',~~
~~                      'DataSourceNameForServer', DataSourceNameForServer,~~
~~                      'DataSourceNameForClient', DataSourceNameForServer,~~
~~                      'DataSourceType', DataSourceType,~~
~~                      'DatasourceASCIIDelimiter', DatasourceASCIIDelimiter,~~
~~                      'DatasourceASCIIHeaderRecords', NumberToString (DatasourceASCIIHeaderRecords ));~~
~~  ENDIF;~~

### Data Source = SQL/ODBC
~~ELSEIF ( ATTRS ( 'TM1 Cube Data Source', pCubeDataSource, 'Type' ) @= 'ODBC');~~
# note: use the data source tab instead if the ODBC password is not to be entered visibly in the TI process
  DataSourceType = 'ODBC' ;
  sDSN = <DSN>;
  sUser = <User for ODBC connection>;
  sPassword = <PW for ODBC connection>;
  ODBCOpenEx ( sDSN, sUser, sPassword, 1);
  DatasourceNameForServer =  sDSN;
  DatasourceNameForClient =  sDSN;
  DataSourceUserName = sUser;
  DataSourcePassword = sPassword;
  DataSourceQuery = <SQL Statement>;
~~  IF ( pLogging @= 'Y' );~~
~~    ASCIIOUTPUT ( LogFileProlog, 'Data Source Config:',~~
~~                      'DataSourceNameForServer', DataSourceNameForServer,~~
~~                      'DataSourceNameForClient', DataSourceNameForServer,~~
~~                      'DataSourceType', DataSourceType,~~
~~                      'DatasourceUserName', DatasourceUsername,~~
~~                      'DatasourcePassword', DatasourcePassword);~~
~~  ENDIF;~~

### Data Source = Cube
~~ELSEIF ( ATTRS ( 'TM1 Cube Data Source', pCubeDataSource, 'Type' ) @= 'Cube');~~
  sSourceCube = <Source Cube>;
  # START: create source view
  sProcess = 'SYS_IBM_View_Create' ;
  IF ( ExecuteProcess ( sProcess, 'pCubeName', sSourceCube,
                             'pViewName', sViewName, 'pSubsetName', sViewName,
                             'pIncludeRules', 'Y', 'pIncludeCLevels', sSourceViewIsCLevel,
                             'pDefaultSubset', '',
                             'pZeroSupress', 'Y',
                             'pCreateMDXSubsets', sCreateMDXSubsets,
                             'pUseExistingView', sUseExistingView,
                            'pDimensionA_Name', pSource_DimensionA_Name,
                            'pDimensionA_Element', pSource_DimensionA_Element,
                            'pDimensionA_Attribute', pSource_DimensionA_Attribute,
                            'pDimensionA_AttributeValue' , pSource_DimensionA_AttributeValue,
                            'pDimensionB_Name', pSource_DimensionB_Name,
                            'pDimensionB_Element', pSource_DimensionB_Element,
                            'pDimensionB_Attribute', pSource_DimensionB_Attribute,
                            'pDimensionB_AttributeValue' , pSource_DimensionB_AttributeValue,
                            'pDimensionC_Name', pSource_DimensionC_Name,
                            'pDimensionC_Element', pSource_DimensionC_Element,
                            'pDimensionC_Attribute', pSource_DimensionC_Attribute,
                            'pDimensionC_AttributeValue' , pSource_DimensionC_AttributeValue,
                            'pDimensionD_Name', pSource_DimensionD_Name,
                            'pDimensionD_Element', pSource_DimensionD_Element,
                            'pDimensionD_Attribute', pSource_DimensionD_Attribute,
                            'pDimensionD_AttributeValue', pSource_DimensionD_AttributeValue,
                            'pDimensionE_Name', pSource_DimensionE_Name,
                            'pDimensionE_Element', pSource_DimensionE_Element,
                            'pDimensionE_Attribute', pSource_DimensionE_Attribute,
                            'pDimensionE_AttributeValue' , pSource_DimensionE_AttributeValue,
                            'pDimensionF_Name', pSource_DimensionF_Name,
                            'pDimensionF_Element', pSource_DimensionF_Element,
                            'pDimensionF_Attribute', pSource_DimensionF_Attribute,
                            'pDimensionF_AttributeValue' , pSource_DimensionF_AttributeValue,
                            'pDimensionF_Name', pSource_DimensionG_Name,
                            'pDimensionF_Element', pSource_DimensionG_Element,
                            'pDimensionF_Attribute', pSource_DimensionG_Attribute,
                            'pDimensionF_AttributeValue' , pSource_DimensionG_AttributeValue )
    <>ProcessExitNormal() );
    sError =  'Prolog Error running Prolog process "' | sProcess | '" for source cube "' | sSourceCube | '".';
    processbreak;
  ENDIF;
  # END: create source view
  DataSourceType = 'VIEW' ;
  DatasourceNameForServer = sSourceCube ;
  DatasourceNameForClient = sSourceCube ;
  DatasourceCubeview = sViewName;
~~  IF ( pLogging @= 'Y' );~~
~~    ASCIIOUTPUT ( LogFileProlog, 'Data Source Config:',~~
~~                      'DataSourceNameForServer', DataSourceNameForServer,~~
~~                      'DataSourceNameForClient', DataSourceNameForServer,~~
~~                      'DataSourceType', DataSourceType);~~
~~  ENDIF;~~
ENDIF;
```

3. Dimension Maintenance

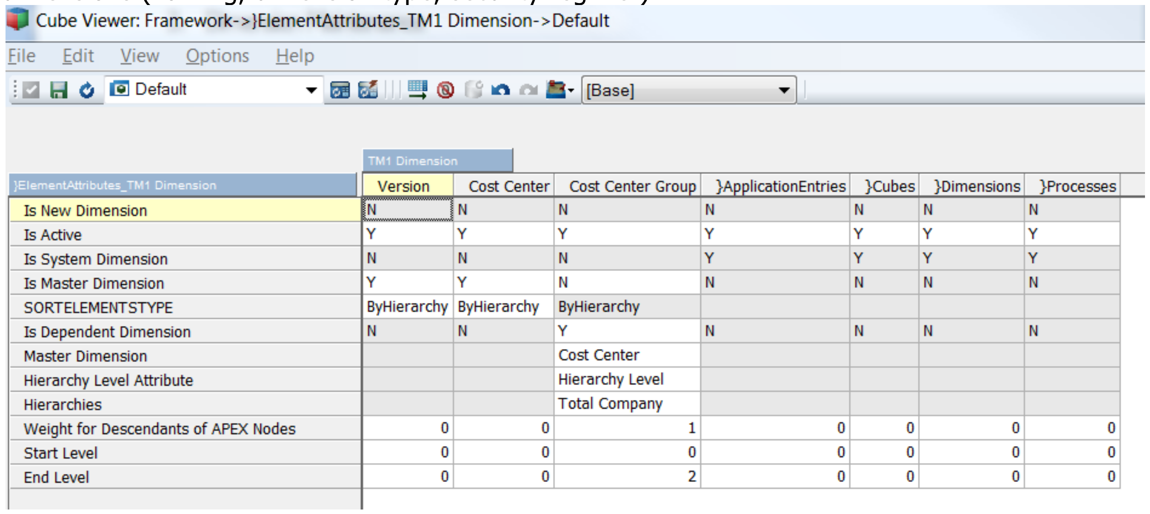
The Dimension maintenance model allows the configuration of new and existing dimensions based on data source to dimension mappings or definition of relationships between master and depending dimensions.

For existing dimensions, the model provides the ability to configure
- dimension element security,
- to update and maintain dependent dimension based on a master dimension
- and to update/maintain master dimensions based on dimension data sources (files and/or SQL queries)

Components:

i. **a TI process called 'Manage Dimension - Create New Dimension Element for Dimension Configuration.pro'**
to create a new dimension element in 'TM1 Dimension.dim'. Alternatively, one may create the 'dimension' manually via the dimension editor. The name of the new dimension element will be the name of the new dimension

ii. **a lookup cube '}ElementAttributes_TM1 Dimension.cub'**, allowing configuration of new dimensions (naming, dimension type, security regime[6]):


Cube Viewer: Framework->}ElementAttributes_TM1 Dimension->Default
File Edit View Options Help
Default
[Base]
TM1 Dimension

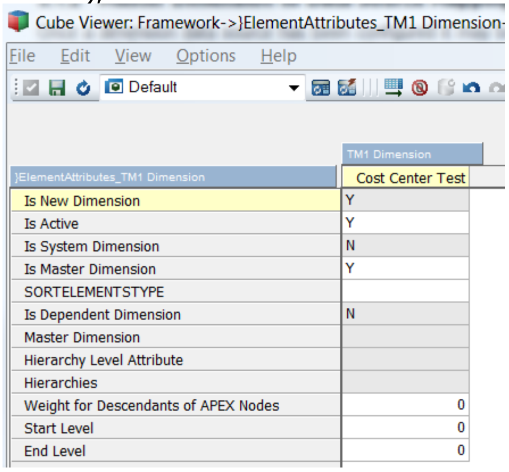
}ElementAttributes_TM1 Dimension	Version	Cost Center	Cost Center Group	}ApplicationEntries	}Cubes	}Dimensions	}Processes
Is New Dimension	N	N	N	N	N	N	N
Is Active	Y	Y	Y	Y	Y	Y	Y
Is System Dimension	N	N	N	Y	Y	Y	Y
Is Master Dimension	Y	Y	N	N	N	N	N
SORTELEMENTSTYPE	ByHierarchy	ByHierarchy	ByHierarchy				
Is Dependent Dimension	N	N	Y	N	N	N	N
Master Dimension			Cost Center				
Hierarchy Level Attribute			Hierarchy Level				
Hierarchies			Total Company				
Weight for Descendants of APEX Nodes	0	0	1	0	0	0	0
Start Level	0	0	0	0	0	0	0
End Level	0	0	2	0	0	0	0

In **'}ElementAttributes_TM1 Dimension.cub'**, define a dimension as a **Master dimension** if the dimension is to be updated via external data feeds (file of SQL based) or if the dimension is manually maintained.
Define a dimension as a **Dependent dimension** (Master Dimension = N, Dependent Dimension = Y) if the dimension is directly dependent on a (master-) dimension, i.e. if changes in the master- or meta-data of the master dimension will lead to a related change in the dependent dimension (example: an organization hierarchy that is derived from the cost center dimension but that will not go to the cost center level but whose leaf level elements are for example the cost center type.

[6] Security configuration measures are not documented in here. Please refer to the security framework documentation for details on dimension element security configuration options.

Each dependent dimension needs to have a master dimension defined.

In the following example, we (1) created a new dimension element 'Cost Center Test' by using the TI process from (i) above, and then configured it as a new master dimension (still to be created), named 'Cost Center Test':


Cube Viewer: Framework->}ElementAttributes_TM1 Dimension-
File Edit View Options Help
Default

	TM1 Dimension
}ElementAttributes_TM1 Dimension	Cost Center Test
Is New Dimension	Y
Is Active	Y
Is System Dimension	N
Is Master Dimension	Y
SORTELEMENTSTYPE	
Is Dependent Dimension	N
Master Dimension	
Hierarchy Level Attribute	
Hierarchies	
Weight for Descendants of APEX Nodes	0
Start Level	0
End Level	0

iii. **a lookup cube '}ElementAttributes_TM1 Dimension Data Source.cub'**, allowing configuration of data sources for master dimensions only (dependent dimensions are sourced based on master dimensions)

iv. **a mapping model 'TM1 Dimension to Data Source Mapping.cub' to map Master Dimensions to Dimension Data Sources**

v. **a TI process 'Manage Dimension - Create or Update Master Dimension.pro'** to update master dimensions with new master- and meta-data based on mapping information in 'TM1 Dimension to Data Source Mapping.cub'.

vi. **a process 'Manage Dimension - Create or Update Dependent Dimension'** to update dependent dimensions based on their configuration in '}ElementAttributes_TM1 Dimension.cub' (f).

vii. **a TI process 'Manage Dimension - Create Alternate Hierarchy based on Attributes.pro' & hierarchy configuration cube '}ElementAttributes_TM1 Hierarchies.cub'**, allowing for configuration and maintenance of alternate hierarchies.

3.1 Update of Master Dimensions

Master Dimensions are to be updated via processing file- or SQL- (ODBC) based master- and meta-data into the dimension.

3.1.1 Master Dimension Data Source Maintenance

The dimension data source maintenance model allows the configuration of new and existing dimension data sources by means of

c) **a TI process called 'Manage Dimension Data Source - Create New Dimension Data Source Element for Dimension Data Source Configuration.pro'**
to create a new dimension element in 'TM1 Dimension Data Source.dim'. Alternatively, one may create the 'data source' manually via the dimension editor. The name of the new dimension element will be the name of the new data source.

d) **a lookup cube '}ElementAttributes_TM1 Dimension Data Source.cub'**, allowing configuration of data sources (type, names, locations, SQL etc.)

Data sources configured via the model can then be used for loading/updating TM1 master dimensions as per '}ElementAttributes_TM1 Dimension.cub'.

Attributes / Configuration parameters in '}ElementAttributes_TM1 Dimension Data Source.cub' (analog to '}ElementAttributes_TM1 Cube Data Source.cub' above:

- **Type**: 'File', 'ODBC'

For data source type = File:

- **Location** (only for data source Type 'File'): inbound data directory as per control cube 'SYS_IBM_Control.cub' = location where csv/txt/cma types will be dropped.
- **File Name** (only for data source Type 'File'): name of load file (note: the name of the load file could be determined automatically based on current period or other parameters.)
- **TM1 TI Data Source Type** (only for data source Type 'File'): CHARACTERDELIMITED or POSITIONDELIMITED
- **TM1 TI Data Source ASCII Delimiter** (only for data source Type 'File'): separator character
  TM1 TI Data Source ASCII Header Records (only for data source Type 'File'): # of header records
- **Is TM1 generated Attribute Export**: if the Elements and Attributes were exported out of TM1 using the included generic dimension meta- & master-data export process 'SYS_IBM_DIM_Export_Dimension_Element_Attributes.pro', set this value to Y to auto-configure the data source.
- **Is TM1 generated Hierarchy Export**: If the dimension elements and hierarchy (as a parent–child relation) were exported out of TM1 using the included generic dimension meta- & master-data export process 'SYS_IBM_DIM_Export_Dimension_Elements_and_Hierarchy.pro', set this value to Y to auto-configure the data source.

For data source type = ODBC:

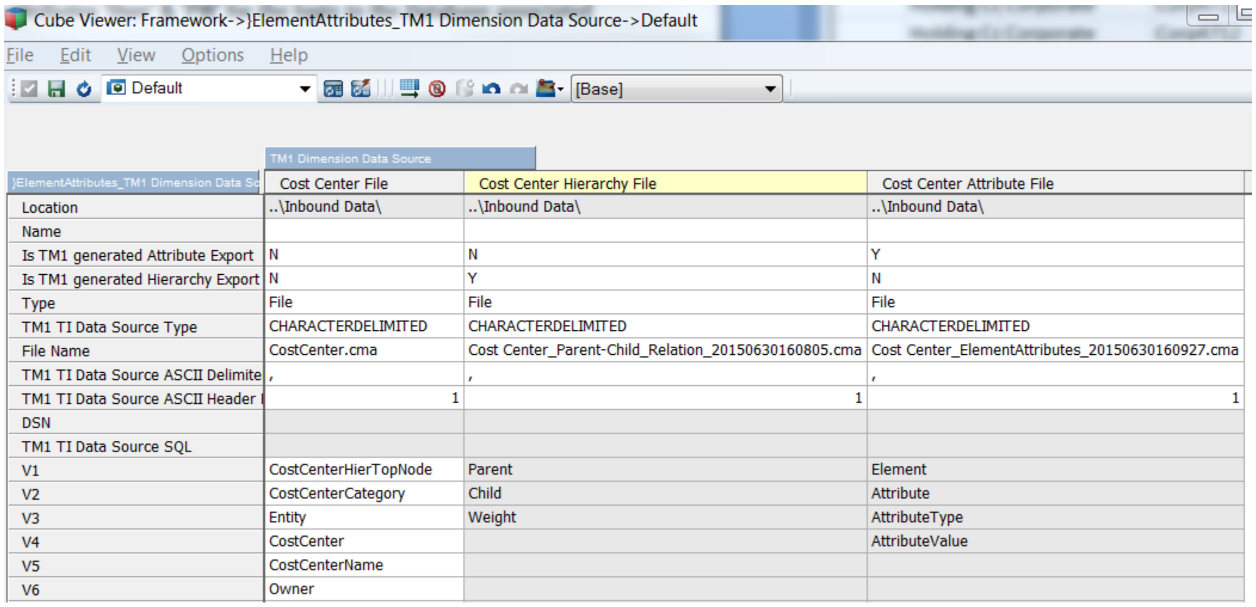
- **DSN** (only for data source Type 'ODBC'): the **D**ata **S**ource **N**ame to use for the SQL Query. You may chose any of the DSNs available in dimension 'TM1 SQL DSN'. **note that TM1 will use the values for SQL DSN attributes 'User' & 'PW' for the login to the Database associated with the DSN!**

- **TM1 TI Data Source SQL** (only for data source Type 'File'): The SQL to run. Note: if the SQL is to be parameterized, we recommend to add a corresponding number of measures/attributes to the dimension (like 'TM1 TI Data Source SQL Pt.I', 'TM1 TI Data Source SQL Pt.II',… as well as parameter attributes like 'SQL Parameter I', 'SQL Parameter II' and then to concatenate the SQL within **TM1 TI Data Source SQL** per cube rule.)

For all data source types:

- **V1-50 (Column1-50)**: free text input. For ODBC and File data sources, V1-50 represent the columns & corresponding names in the data source. For each column, assign a name that you feel best represents the data in the corresponding data source column. You can assign any name.

3.1.2 Sample Dimension Data Sources


Cube Viewer: Framework->}ElementAttributes_TM1 Dimension Data Source->Default
File
Edit
View
Options
Help
Default
[Base]
TM1 Dimension Data Source

}ElementAttributes_TM1 Dimension Data So	Cost Center File	Cost Center Hierarchy File	Cost Center Attribute File
Location	..\Inbound Data\	..\Inbound Data\	..\Inbound Data\
Name			
Is TM1 generated Attribute Export	N	N	Y
Is TM1 generated Hierarchy Export	N	Y	N
Type	File	File	File
TM1 TI Data Source Type	CHARACTERDELIMITED	CHARACTERDELIMITED	CHARACTERDELIMITED
File Name	CostCenter.cma	Cost Center_Parent-Child_Relation_20150630160805.cma	Cost Center_ElementAttributes_20150630160927.cma
TM1 TI Data Source ASCII Delimite	,	,	,
TM1 TI Data Source ASCII Header I	1	1	1
DSN			
TM1 TI Data Source SQL			
V1	CostCenterHierTopNode	Parent	Element
V2	CostCenterCategory	Child	Attribute
V3	Entity	Weight	AttributeType
V4	CostCenter		AttributeValue
V5	CostCenterName		
V6	Owner		

- **Cost Center_ElementAttributes_20150630160927.cma**: **Is TM1 generated Attribute Export of dimension 'Cost Center' as per included generic dimensio**n meta- & master-data export process 'SYS_IBM_DIM_Export_Dimension_Element_Attributes.pro':

```
"Element","Attribute","AttributeType","AttributeValue"
"Total Company","Cost Center Group","AS",""
"Total Company","Hierarchy Level","AS","0"
"Total Company","TM1 Level","AN"," 3"
"Total Company","SQFT","AN",""
"Holding Company","Cost Center Group","AS",""
"Holding Company","Hierarchy Level","AS","1"
"Holding Company","TM1 Level","AN"," 2"
"Holding Company","SQFT","AN",""
…
```

- **Cost Center_Parent-Child_Relation_20150630160805.cma**: Is TM1 generated Hierarchy Export of dimension 'Cost Center' as per included generic dimension meta- & master-data export process 'SYS_IBM_DIM_Export_Dimension_Elements_and_Hierarchy.pro':

```
"Parent","Child","Weight"
"Total Company","Holding Company","1"
"Holding Company","Corporate","1"
"Corporate","Corp4711","1"
"Corporate","Corp4712","1"
"Corporate","Corp4713","1"
```

```
"Corporate","Corp4714","1"
"Corporate","Corp4715","1"
"Corporate","Corp4716","1"
"Corporate","Corp4717","1"
"Corporate","Corp4718","1"
"Corporate","Corp4719","1"
"Corporate","Corp4720","1"
"Total Company","North America","1"
"North America","Legal Entity A","1"
"Legal Entity A","A4711","1"
"Legal Entity A","A4712","1"
"Legal Entity A","A4713","1"
…
```

- **CostCenter.cma**: Cost Center Hier Levels in separate columns, incl. cost center name and cost center owner:

```
“CostCenterHierTopNode”, “CostCenterCategory”, “Entity”, “CostCenter”, “CostCenterName”, “Owner”
"Total Company","Holding Company","Corporate","Corp4711","Mktg","Jack Frost"
"Total Company","Holding Company","Corporate","Corp4712","Finance","Donald Duck"
"Total Company","Holding Company","Corporate","Corp4713","HO","Davy Jones"
"Total Company","Holding Company","Corporate","Corp4714","Sales","Jack Sparrow"
"Total Company","Holding Company","Corporate","Corp4715","",
"Total Company","Holding Company","Corporate","Corp4716","",
"Total Company","Holding Company","Corporate","Corp4717","",
"Total Company","Holding Company","Corporate","Corp4718","",
"Total Company","Holding Company","Corporate","Corp4719","",
"Total Company","Holding Company","Corporate","Corp4720","",
"Total Company","North America","Legal Entity A","A4711","Mktg",Peter P.
"Total Company","North America","Legal Entity A","A4712","Finance",Anna K.
"Total Company","North America","Legal Entity A","A4713","HO",Clark Kent
"Total Company","North America","Legal Entity A","A4714","Sales",Lois Lane
"Total Company","North America","Legal Entity A","A4715","",
"Total Company","North America","Legal Entity A","A4716","",
"Total Company","North America","Legal Entity A","A4717","",
```

3.1.3 Master Dimension to Data Source Mapping

Once a dimension data source has been configured it may be mapped to an existing master dimension. You can define multiple mappings, i.e. a dimension can be mapped to multiple data sources. Mapping occurs via corresponding entries in cube 'TM1 Dimension to Data Source Mapping.cub':

Mapping Measures:

- Mapping is Valid: = 1 if
  - Target dimension is a master dimension
  - parent/ child relationship is properly defined OR only child is defined (Child = element name)
- Level N (with N = 2-6): picklist; column in data source that contains element level N
- Level N Variable (with N = 2-6): automatically derived based on entry in 'Level N' and dimension data source configuration in '}ElementAttributes_TM1 Dimension Data Source.cub'.
- Parent: picklist; column in data source that contains element parent (or level 1)
- Parent Variable: automatically derived based on entry in 'Parent' and dimension data source configuration in '}ElementAttributes_TM1 Dimension Data Source.cub'
- Child: picklist; column in data source that contains element name
- Child Variable: automatically derived based on entry in 'Child' and dimension data source configuration in '}ElementAttributes_TM1 Dimension Data Source.cub'
- Weight: optional; name of column to be used to determine element weight (weight of element in relation to the parent)
- Weight Variable: automatically derived based on entry in 'Weight' and dimension data source configuration in '}ElementAttributes_TM1 Dimension Data Source.cub'
- Hierarchy: If PA V2.0 Hierarchies are enabled for the TM1 Database instance (EnableNewHierarchyCreation=T in TM1s.cfg) and if the 'SYS_IBM_Control' cube Framework parameter 'Hierarchies Enabled' is set to Y, 'Hierachy' allows to specify the target hierarchy container for the dimension update/load. If Hierarchy is left empty and hierarchies are enabled, the default container (name = dimension name) will be used. If a Hierarchy is specified and it does not exist, it will be created.
- Make Leaf Element Names a compound of <Parent> - <Child>: if set to Y, will make the dimension leafs a compound of the Parent element and the child element defined in the data source, i.e. child = <Parent> - <Child>
- Compound Alias Attribute I: optional; picklist with data source columns; if an alias is to be created based on multiple columns, select the column containing the first portion of the alias
- Compound Alias Attribute I Variable: automatically derived based on entry in 'Compound Alias Attribute I' and dimension data source configuration in '}ElementAttributes_TM1 Dimension Data Source.cub'
- Compound Alias Attribute II: optional; picklist with data source columns; if an alias is to be created based on multiple columns, select the column containing the 2nd portion of the alias
- Compound Alias Attribute II Variable: automatically derived based on entry in 'Compound Alias Attribute II' and dimension data source configuration in '}ElementAttributes_TM1 Dimension Data Source.cub'
- Compound Alias Separator: separator to be inserted between Compound Alias Attribute I & II. Example: <Description> - (<ElementKey>), with ' – ' as the separator
- Compound Alias Attribute II in Parenthesis: set to Y if to enclose the 2nd Alias component in parenthesis as in <Description> - (<ElementKey>)
- Compound Alias Name: Name of the compound alias attribute
- Is Valid Compound Alias: automatic check (Y/N) if compound alias is properly configured

- One Attribute Name/Type per record:
  - Attribute: Column with contains the Attribute Value

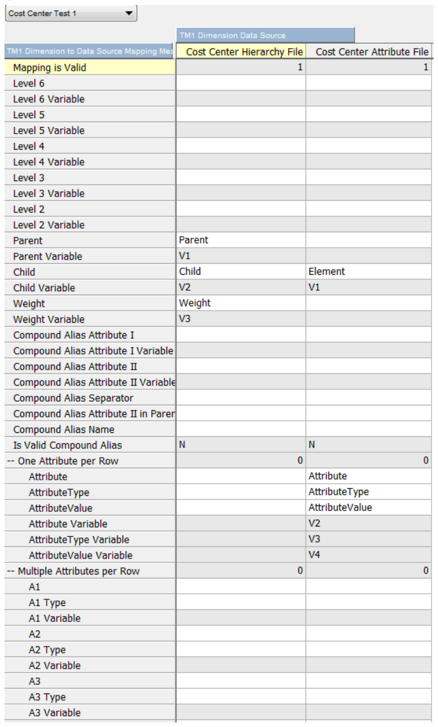
  - <u>Attribute Variable</u>: Variable for Column with contains the Attribute Value, auto-generated
  - <u>AttributeType</u>: Column which contains the Attribute Type
  - <u>AttributeType Variable</u>: Variable Column which contains the Attribute Type, auto-generated
  - <u>AttributeName</u>: Column which contains the Attribute Name
  - <u>AttributeName Variable</u>: Variable for Column which contains the Attribute Name, auto-generated

- Multiple Attributes Name/Type per record = One Attribute per Column:
  With X = 1-13
  - <u>AX</u>: Data Source Column name to use for Attribute X
  - <u>AX Type</u>: Attribute X type (Picklist with 'Alias', 'Text', 'Number')
  - <u>AX Variable</u>: data source variable (automatically generated based on entry in AX

Notes:
- Weight only applies to the leaf element as in relation to its parent.
- If Weight is not defined, a weight of 1 will automatically be set.
- A Parent-Child relationship hierarchy is loaded by only populating 'Parent', 'Child' and 'Weight'
- A hierarchy load where different levels are not organized in a parent-child relationship but where different levels are in different columns is loaded by using
  - Child as level 0
  - Parent as level 1
  - Level 2-N as levels 2-N
- Hierarchy Levels (Parent / Level 1 & Levels 2-N) are created with weight 1)

3.1.4 Sample Dimension Mappings

Dimension 'Cost Center Test 1':

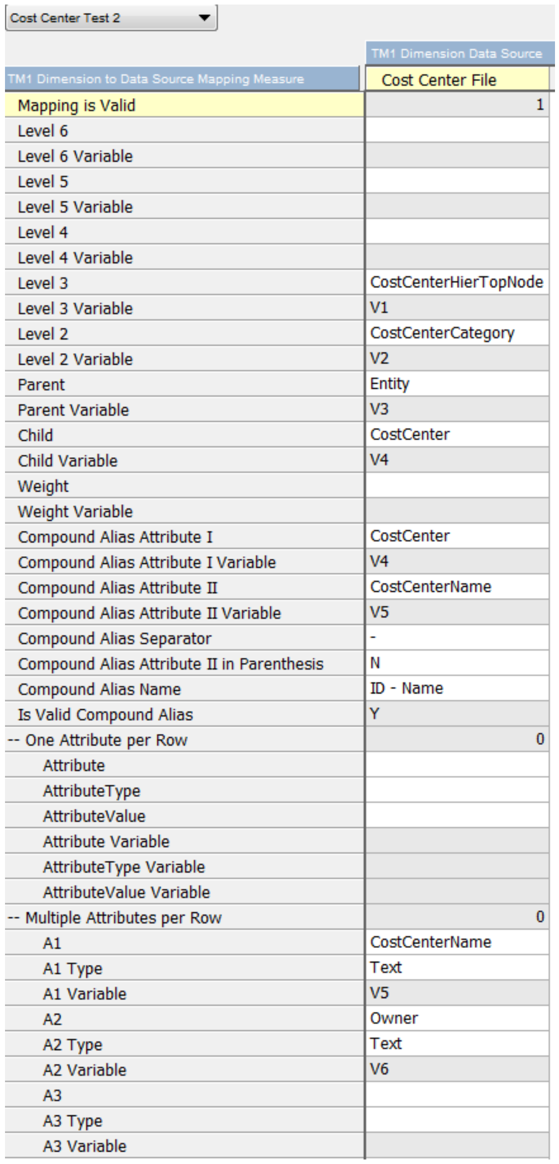


Cost Center Test 1

	TM1 Dimension Data Source	
TM1 Dimension to Data Source Mapping Mea	Cost Center Hierarchy File	Cost Center Attribute File
Mapping is Valid	1	1
Level 6		
Level 6 Variable		
Level 5		
Level 5 Variable		
Level 4		
Level 4 Variable		
Level 3		
Level 3 Variable		
Level 2		
Level 2 Variable		
Parent	Parent	
Parent Variable	V1	
Child	Child	Element
Child Variable	V2	V1
Weight	Weight	
Weight Variable	V3	
Compound Alias Attribute I		
Compound Alias Attribute I Variable		
Compound Alias Attribute II		
Compound Alias Attribute II Variable		
Compound Alias Separator		
Compound Alias Attribute II in Parer		
Compound Alias Name		
Is Valid Compound Alias	N	N
-- One Attribute per Row	0	0
Attribute		Attribute
AttributeType		AttributeType
AttributeValue		AttributeValue
Attribute Variable		V2
AttributeType Variable		V3
AttributeValue Variable		V4
-- Multiple Attributes per Row	0	0
A1		
A1 Type		
A1 Variable		
A2		
A2 Type		
A2 Variable		
A3		
A3 Type		
A3 Variable		

=> Mapped to (a) the Hierarchy File (defining the parent child relationship) & mapped to (b) the Attributes File (containing attributes for the dimension elements)

Dimension 'Cost Center Test 2':



Cost Center Test 2

TM1 Dimension to Data Source Mapping Measure	TM1 Dimension Data Source: Cost Center File
Mapping is Valid	1
Level 6	
Level 6 Variable	
Level 5	
Level 5 Variable	
Level 4	
Level 4 Variable	
Level 3	CostCenterHierTopNode
Level 3 Variable	V1
Level 2	CostCenterCategory
Level 2 Variable	V2
Parent	Entity
Parent Variable	V3
Child	CostCenter
Child Variable	V4
Weight	
Weight Variable	
Compound Alias Attribute I	CostCenter
Compound Alias Attribute I Variable	V4
Compound Alias Attribute II	CostCenterName
Compound Alias Attribute II Variable	V5
Compound Alias Separator	-
Compound Alias Attribute II in Parenthesis	N
Compound Alias Name	ID - Name
Is Valid Compound Alias	Y
-- One Attribute per Row	0
Attribute	
AttributeType	
AttributeValue	
Attribute Variable	
AttributeType Variable	
AttributeValue Variable	
-- Multiple Attributes per Row	0
A1	CostCenterName
A1 Type	Text
A1 Variable	V5
A2	Owner
A2 Type	Text
A2 Variable	V6
A3	
A3 Type	
A3 Variable	

=> Mapped to a source File with
a) elements (defined/mapped per measure 'Child')
b) Parents (Level 1), Levels 2-3
c) A compound alias consisting of Cost Center Element Name (Child Variable) & the Cost Center Name Attribute, separated by '-'
d) Text Attributes Cost Center Name & Owner

3.1.5 Master Dimension Load & Update

Once a data source is mapped to a dimension, the process **'Manage Dimension - Create or Update Master Dimension.pro'** is used to update/load the master & meta-data.

Parameters:

- pDimension: Target Dimension = Dimension as per configuration in 'TM1 Dimension to Data Source Mapping.cub'
- pDimensionDataSource: Data Source = Data Source as per configuration in 'TM1 Dimension to Data Source Mapping.cub'
- pLogging: only set to Y for detailed debugging; set to empty (default=N) or N for normal operation.
- pOverwriteExistingDimension: Set to Y ONLY to do a complete dimension overwrite. All Dimension Elements will be removed prior to re-load.
- pAppendToExistingDimension: Set to Y to append elements to the existing hierarchies. The existing dimension elements will not be deleted nor will any existing hierarchies be deleted.
- pOverwriteCLevelsOnly: Set to Y remove Consolidated elements from the dimension prior to re-load/update[7]
- pZeroOutTarget: if Y, the target dimension }ElementAttributes cube will be zeroed out prior to the dimension update operation.
- pSkipMetadata: set to Y to skip metadata tab, i.e. to skip dimension element and hierarchy update (for example if you only want to update element attribute values)
- pSkipData: set to Y to skip data tab, i.e. to skip update of element attribute values (for example if you only want to update dimension elements & hierarchies)
- pUpdateSYSControlDimAttributes: will automatically update metadata in 'SYS_ControlDimAttributes.dim'. See section <Maintenance of Alternate Master Dimension Hierarchies> below for information on this functionality.
- pForceCompoundAlias: will enforce creating compound alias names (hence avoiding any possible issues with alias names not being unique)
- pUnwindHierarchies: se to Y to 'unwind' hierarchies only (Overwrites&Appends need to be set to N)

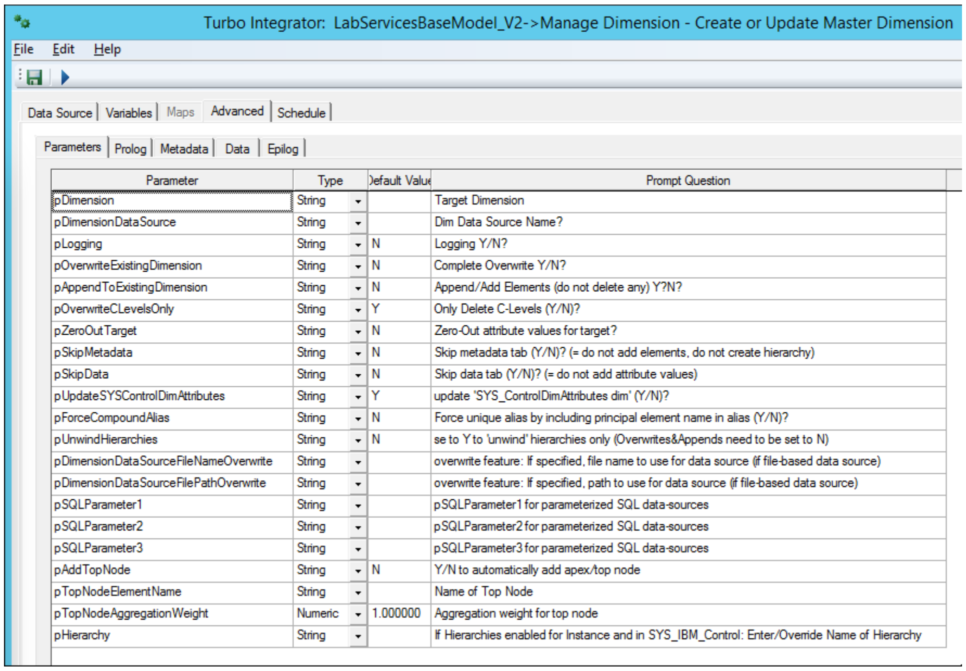
- pDimensionDataSourceFileNameOverwrite: overwrite feature; if specified, file name to use for data source (if file-based data source) instead of file name defined with data source
- pDimensionDataSourceFilePathOverwrite: overwrite feature: If specified, path to use for data source (if file-based data source)
- pSQLParameter1: pSQLParameter1 for parameterized SQL data-sources; If the SQL Statement for the ODBC data source is modified like *Select … from … where … = '**%pSQLParameter1%**';* the corresponding parameter will be inserted into the SQL query at process runtime.
- pSQLParameter2: pSQLParameter2 for parameterized SQL data-sources
- pSQLParameter3: pSQLParameter3 for parameterized SQL data-sources

- pAddTopNode: IF pAddTopNode @= 'Y' & pTopNodeElementName @<> '' , process 'SYS_IBM_DIM_Hierarchy_Assign_Top_Node' will be triggered. Note that process 'SYS_IBM_DIM_Hierarchy_Assign_Top_Node' also contains runtime parameters to allow excluding

[7] An alternate method of deleting C-Levels prior to re-load: for very large dimensions, it may be faster not to use parameter pOverwriteCLevelsOnly to remove rollups; in other cases you may want to delete only specific elements from a dimension prior to update. Use the following alternate procedure to delete unwanted elements and to update the dimension:
1) create an MDX subset with all elements with level >0.
2) use process 'SYS_IBM_DIM_Delete_Elements_according_to_Subset.pro' to delete all elements from the dimension that are part of the subset created in (1)
3) run 'Manage Dimension - Create or Update Master Dimension.pro' with pAppendToExistingDimension = Y and pOverwriteExistingDimension = N
4) if necessary, delete the subset (1)

hierarchies and elements to be added to the top node. The exclusions are not passed to this process at this time. Either add default exclusions (such as an orphans node) to the paramters being passed to 'SYS_IBM_DIM_Hierarchy_Assign_Top_Node' or save 'SYS_IBM_DIM_Hierarchy_Assign_Top_Node' with default exclusions.

- pTopNodeElementName: Name of Top Node Element (will be created if it does not exist)
- pTopNodeAggregationWeight: aggregation/consolidation weight for children of the top node.
- pHierarchy: Override feature: If Hierarchies enabled for Instance and in SYS_IBM_Control Enter/Override Name of Hierarchy if Hierarchy is to be different from default mapping in 'TM1 Dimension to Data Source Mapping.cub'.


Turbo Integrator: LabServicesBaseModel_V2->Manage Dimension - Create or Update Master Dimension
File Edit Help
Data Source | Variables | Maps | Advanced | Schedule
Parameters | Prolog | Metadata | Data | Epilog

Parameter	Type	Default Value	Prompt Question
pDimension	String		Target Dimension
pDimensionDataSource	String		Dim Data Source Name?
pLogging	String	N	Logging Y/N?
pOverwriteExistingDimension	String	N	Complete Overwrite Y/N?
pAppendToExistingDimension	String	N	Append/Add Elements (do not delete any) Y?N?
pOverwriteCLevelsOnly	String	Y	Only Delete C-Levels (Y/N)?
pZeroOutTarget	String	N	Zero-Out attribute values for target?
pSkipMetadata	String	N	Skip metadata tab (Y/N)? (= do not add elements, do not create hierarchy)
pSkipData	String	N	Skip data tab (Y/N)? (= do not add attribute values)
pUpdateSYSControlDimAttributes	String	Y	update 'SYS_ControlDimAttributes dim' (Y/N)?
pForceCompoundAlias	String	N	Force unique alias by including principal element name in alias (Y/N)?
pUnwindHierarchies	String	N	se to Y to 'unwind' hierarchies only (Overwrites&Appends need to be set to N)
pDimensionDataSourceFileNameOverwrite	String		overwrite feature: If specified, file name to use for data source (if file-based data source)
pDimensionDataSourceFilePathOverwrite	String		overwrite feature: If specified, path to use for data source (if file-based data source)
pSQLParameter1	String		pSQLParameter1 for parameterized SQL data-sources
pSQLParameter2	String		pSQLParameter2 for parameterized SQL data-sources
pSQLParameter3	String		pSQLParameter3 for parameterized SQL data-sources
pAddTopNode	String	N	Y/N to automatically add apex/top node
pTopNodeElementName	String		Name of Top Node
pTopNodeAggregationWeight	Numeric	1.000000	Aggregation weight for top node
pHierarchy	String		If Hierarchies enabled for Instance and in SYS_IBM_Control: Enter/Override Name of Hierarchy

Notes:

- priority: pOverwriteExistingDimension over pAppendToExistingDimension over pOverwriteCLevelsOnly over pUnwindHierarchies
- If you need to delete a specific hierarchy prior to rebuild, you can use processes SYS_IBM_Subset_Create.pro and SYS_IBM_DIM_Delete_Elements_according_to_Subset.pro to delete a specific C-Level hierarchy from the dimension prior to update/refresh
- if pOverwriteExistingDimension = N & pAppendToExistingDimension = N & pOverwriteCLevelsOnly = N & pUnwindHierarchies = N, the process will behave as if pAppendToExistingDimension = Y

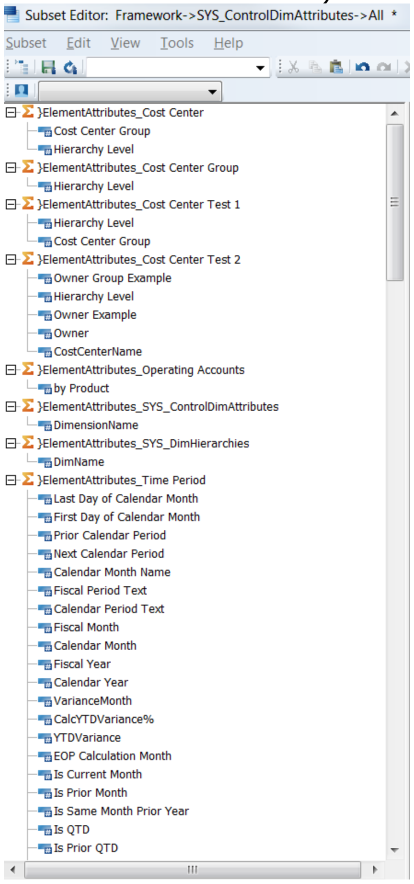
3.1.6 Maintenance of Alternate Master Dimension Hierarchies

<u>This feature pertains to alternate hierarchies within **one** PA dimension hierarchy container, not to PA Hierarchies</u>. PA Hierarchies are specified, managed and maintained via the 'Hierarchies' mapping measure (see above). IN other words: via the following feature, alternate hierarchies within one hierarchy container may be managed.

Alternate Hierarchies can be generated and automatically-maintained by using lookup cube '}ElemetnAttributes_TM1 Hierarchies.cub'. The hierarchies are defined via assigning relationships between hierarchy levels (Level 0 = leaf level) and element attributes and their corresponding values.

Run process 'Manage Dimension - Create Alternate Hierarchy based on Attributes.pro' to create alternate hierarchies based on the entries in '}ElementAttributes_TM1 Hierarchies.cub'. The process will

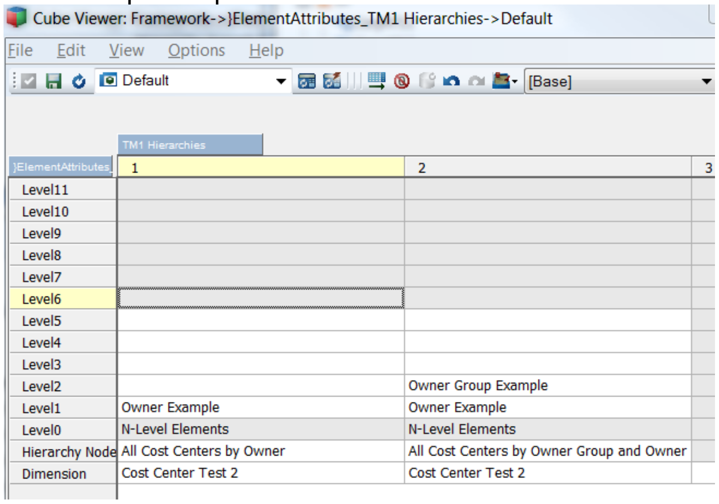
i. Determine if the hierarchy level attributes defined in }ElementAttributes_TM1 Hierarchies.cub are in fact valid attributes of the target dimension. The process will leverage Dimension 'SYS_ControlDimAttributes.dim'. This dimension can be updated using process 'Manage Dimension – Update SYS_ControlDimAttributes.pro' (to include all attributes) or one can update the dimension manually to only include attributes that are permissible for building a hierarchy. The control dimension 'SYS_ControlDimAttributes.dim' is also used by the hierarchy maintenance cube '}ElementAttributes_TM1 Hierarchies.cub' to determine the max # of permissible hierarchy levels.
The following possible hierarchy levels where identified in our sample database (note the 5 possible hierarchy levels for dimension 'Cost Center Test 2'):



- }ElementAttributes_Cost Center
  - Cost Center Group
  - Hierarchy Level
- }ElementAttributes_Cost Center Group
  - Hierarchy Level
- }ElementAttributes_Cost Center Test 1
  - Hierarchy Level
  - Cost Center Group
- }ElementAttributes_Cost Center Test 2
  - Owner Group Example
  - Hierarchy Level
  - Owner Example
  - Owner
  - CostCenterName
- }ElementAttributes_Operating Accounts
  - by Product
- }ElementAttributes_SYS_ControlDimAttributes
  - DimensionName
- }ElementAttributes_SYS_DimHierarchies
  - DimName
- }ElementAttributes_Time Period
  - Last Day of Calendar Month
  - First Day of Calendar Month
  - Prior Calendar Period
  - Next Calendar Period
  - Calendar Month Name
  - Fiscal Period Text
  - Calendar Period Text
  - Fiscal Month
  - Calendar Month
  - Fiscal Year
  - Calendar Year
  - VarianceMonth
  - CalcYTDVariance%
  - YTDVariance
  - EOP Calculation Month
  - Is Current Month
  - Is Prior Month
  - Is Same Month Prior Year
  - Is QTD
  - Is Prior QTD

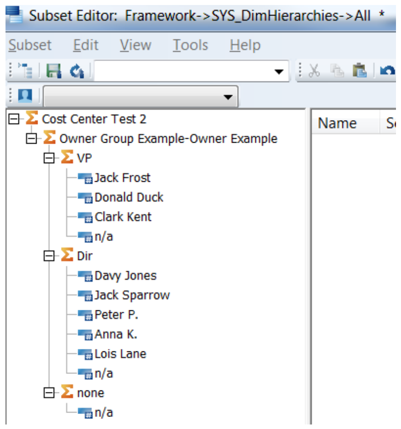
ii. Launch 'Manage Dimension - Create Alternate Hierarchy based on Attributes - Check Duplicates.pro' to check for duplicate parent entries in }ElementAttributes_TM1 Hierarchies.cub and to hence validate the target hierarchy:
   - By using 'SYS_IBM_Dim_Hierarchy_Create_Alternate_Attribute_Hierarchy_Delete_Hierarchy', the process will first reset a temporary dimension called 'SYS DimHierarchies.dim'
   - If more than one Ancestor Level (not just a Level 1) are defined for a hierarchy, the process will build/update a temporary dimension called 'SYS DimHierarchies.dim' according to the hierarchy levels defined in 'SYS Dim Hierarchies.cub'.
   - In case where a hierarchy level has multiple parents within a hierarchy, a message is generated in the Process Logs output folder with content such as "Dimension: <Dimension Name>","Multiple parent(s) for <Level N+1>: <LevelN+1ElementName> in PFS-GFS Level<N> Level<N+1>"
     for example: "Dimension: cost center hierarchy test 2","Multiple parent(s) for GFS : N/A in PFS-GFS Level1 Level2"

Example: based on the 5 possible hierarchy levels for dimension 'Cost Center Test 2' - see (i) above – only up to 5 levels can be defined for dimension 'Cost Center Test 2'. In our example, we define hierarchies 'All Cost Centers by Owner' (APEX Node) with only one level (1) Owner Example & 'All Cost Centers by Owner Group and Owner' (APEX Node) with only levels (1) Owner Example and (2) Owner Group Example:


Cube Viewer: Framework->}ElementAttributes_TM1 Hierarchies->Default
File Edit View Options Help
Default
[Base]
TM1 Hierarchies

}ElementAttributes	1	2	3
Level11			
Level10			
Level9			
Level8			
Level7			
Level6			
Level5			
Level4			
Level3			
Level2		Owner Group Example	
Level1	Owner Example	Owner Example	
Level0	N-Level Elements	N-Level Elements	
Hierarchy Node	All Cost Centers by Owner	All Cost Centers by Owner Group and Owner	
Dimension	Cost Center Test 2	Cost Center Test 2	

For this hierarchy definition, the following temporary test hierarchy was created in 'SYS DimHierarchies.dim:


Subset Editor: Framework->SYS_DimHierarchies->All *
Subset Edit View Tools Help
Name

```
Cost Center Test 2
  Owner Group Example-Owner Example
    VP
      Jack Frost
      Donald Duck
      Clark Kent
      n/a
    Dir
      Davy Jones
      Jack Sparrow
      Peter P.
      Anna K.
      Lois Lane
      n/a
    none
      n/a
```

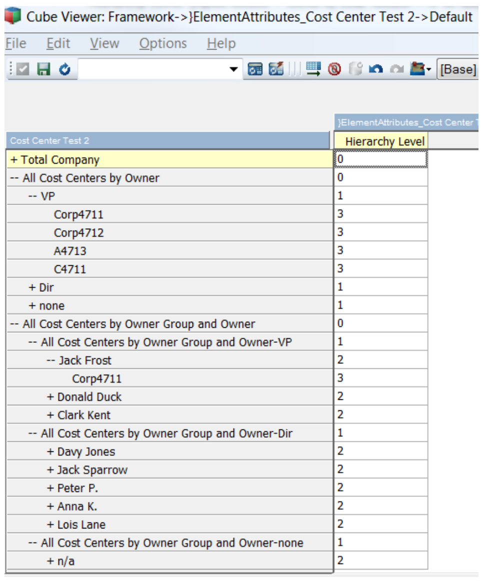
iii. Launch 'Manage Dimension - Create Alternate Hierarchy based on Attributes - Build Alternate Hierarchy.pro' to build the alternate hierarchy:

a) Prolog: Delete the existing hierarchy via process 'Manage Dimension - Create Alternate Hierarchy based on Attributes - Delete Hierarchy.pro' by deleting all elements where the value of Attribute 'HierName' is = the name of the hierarchy that is to be rebuilt:

b) MetaData tab: Rebuild the hierarchy, with <u>branch level names being adjusted automatically</u> (concatenated with the element parent name) <u>**if** the element already exists in the dimension under a different branch or hierarchy</u>.

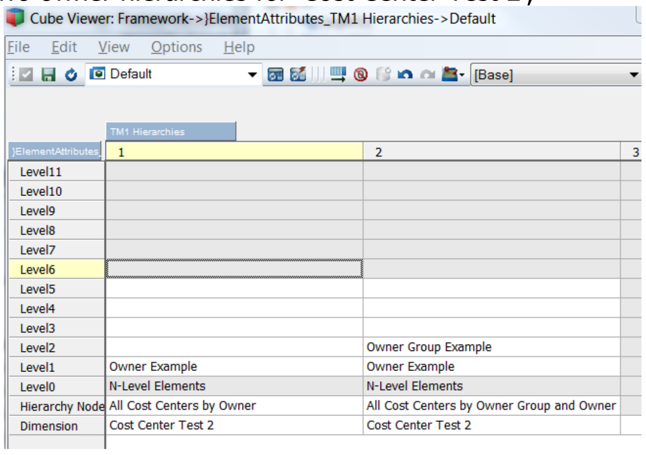
c) Data tab: Update values for attribute 'HierName' (see above under (a)

iv. Launch 'Manage Dimension - Update Hierarchy Level Attribute.pro' to update values for attribute 'Hierarchy Level' (Apex Nodes = Level 0, Leaf Nodes = # of deepest Hierarchy):


Cube Viewer: Framework->}ElementAttributes_Cost Center Test 2->Default
File
Edit
View
Options
Help
[Base]
}ElementAttributes_Cost Center

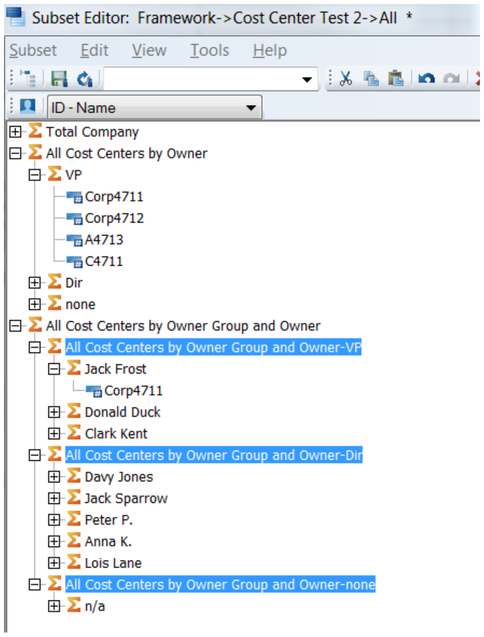
Cost Center Test 2	Hierarchy Level
+ Total Company	0
-- All Cost Centers by Owner	0
-- VP	1
Corp4711	3
Corp4712	3
A4713	3
C4711	3
+ Dir	1
+ none	1
-- All Cost Centers by Owner Group and Owner	0
-- All Cost Centers by Owner Group and Owner-VP	1
-- Jack Frost	2
Corp4711	3
+ Donald Duck	2
+ Clark Kent	2
-- All Cost Centers by Owner Group and Owner-Dir	1
+ Davy Jones	2
+ Jack Sparrow	2
+ Peter P.	2
+ Anna K.	2
+ Lois Lane	2
-- All Cost Centers by Owner Group and Owner-none	1
+ n/a	2

In our example with the two owner hierarchies for 'Cost Center Test 2',


Cube Viewer: Framework->}ElementAttributes_TM1 Hierarchies->Default
File
Edit
View
Options
Help
Default
[Base]
TM1 Hierarchies

}ElementAttributes_	1	2	3
Level11			
Level10			
Level9			
Level8			
Level7			
Level6			
Level5			
Level4			
Level3			
Level2		Owner Group Example	
Level1	Owner Example	Owner Example	
Level0	N-Level Elements	N-Level Elements	
Hierarchy Node	All Cost Centers by Owner	All Cost Centers by Owner Group and Owner	
Dimension	Cost Center Test 2	Cost Center Test 2	

the TI-process 'Manage Dimension - Create Alternate Hierarchy based on Attributes.pro' created the following new rollups, with the Hierarchies automatically built such that hierarchy branch levels are unique across all hierarchies (see highlighted 3 elements that were created as a compound element including the hierarchy name because the element names (VP, Dir, None) are already part of the first Owner hierarchy):


Subset Editor: Framework->Cost Center Test 2->All *
Subset Edit View Tools Help
ID - Name

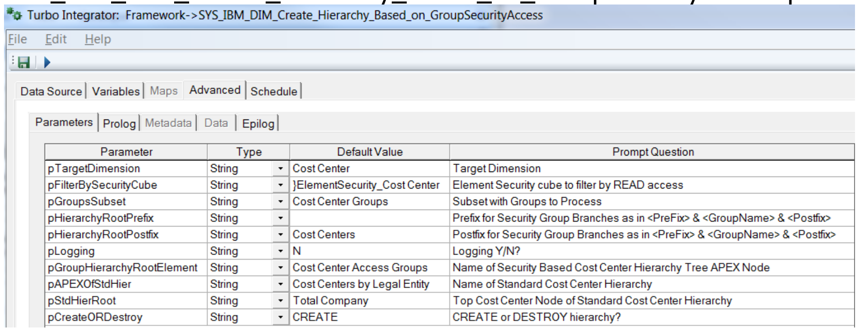
```
Total Company
All Cost Centers by Owner
    VP
        Corp4711
        Corp4712
        A4713
        C4711
    Dir
    none
All Cost Centers by Owner Group and Owner
    All Cost Centers by Owner Group and Owner-VP
        Jack Frost
            Corp4711
        Donald Duck
        Clark Kent
    All Cost Centers by Owner Group and Owner-Dir
        Davy Jones
        Jack Sparrow
        Peter P.
        Anna K.
        Lois Lane
    All Cost Centers by Owner Group and Owner-none
        n/a
```

Note: if the dimension is secured by ElementSecurity <u>and</u> if hierarchy definitions in '}ElementAttributes_TM1 Hierarchies.cub' lead to new branches and/or elements being inserted/created in the target dimension, security may have to be re-processed/adjusted for the Master Dimension and its dependent dimensions.

3.1.7 Maintenance of Security-Specific Rollups

In order to allow users to see a total for all the cost centers that they have access to[8] and to provide the ability to specify the corresponding element as a default in reports and views, a special alternate hierarchy can be created via process
'SYS_IBM_DIM_Create_Hierarchy_Based_on_GroupSecurityAccess.pro'
which will group & organize cost centers by security model group access.

'SYS_IBM_DIM_Create_Hierarchy_Based_on_GroupSecurityAccess.pro' Parameters:



Parameter	Type	Default Value	Prompt Question
pTargetDimension	String	Cost Center	Target Dimension
pFilterBySecurityCube	String	}ElementSecurity_Cost Center	Element Security cube to filter by READ access
pGroupsSubset	String	Cost Center Groups	Subset with Groups to Process
pHierarchyRootPrefix	String		Prefix for Security Group Branches as in <PreFix> & <GroupName> & <Postfix>
pHierarchyRootPostfix	String	Cost Centers	Postfix for Security Group Branches as in <PreFix> & <GroupName> & <Postfix>
pLogging	String	N	Logging Y/N?
pGroupHierarchyRootElement	String	Cost Center Access Groups	Name of Security Based Cost Center Hierarchy Tree APEX Node
pAPEXOfStdHier	String	Cost Centers by Legal Entity	Name of Standard Cost Center Hierarchy
pStdHierRoot	String	Total Company	Top Cost Center Node of Standard Cost Center Hierarchy
pCreateORDestroy	String	CREATE	CREATE or DESTROY hierarchy?

[8] Without requiring users to create a custom rollup

- pTargetDimension: Target Dimension
- pFilterBySecurityCube: Element Security cube to filter by READ access
- pGroupsSubset: Subset with Groups to Process
- pHierarchyRootPrefix: Prefix for Security Group Branches as in <PreFix> & <GroupName> & <Postfix>
- pHierarchyRootPostfix: Postfix for Security Group Branches as in <PreFix> & <GroupName> & <Postfix>
- pLogging: Y/N (debug logging)
- pGroupHierarchyRootElement: Name of Security Based Dimension Hierarchy Tree APEX Node
- pAPEXOfStdCostCenterHier: Name of Standard/Default Dimension Hierarchy
- pStdCostCenterHierRoot: Top Cost Center Node of Standard Cost Center Hierarchy
- pCreateOrDestroy: CREATE to create the security rollups. DESTROY to destroy the security rollup

Note that Process 'SYS_IBM_DIM_Create_Hierarchy_Based_on_GroupSecurityAccess.pro' will allow creation of security group based rollups on any dimension based on the value or parameter pTargetDimension. Also note that the }ElementSecurity cube does not have to reference the same dimension as the target dimension. A different security cube can be referenced and will result in the rollups only including elements that are found in the specified security cube and for which a group has been granted READ access. Also: the process can be modified easily to account for READ & WRITE access etc.

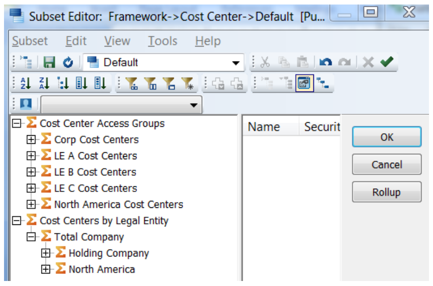
Process Logic:

1) Remove security specific rollups: remove all cost centers underneath a Security Model Group branch

Then, for each security group in pGroupSubset:

2) Create Hierarchy Branch Node for each new Model Group
3) For each Security Model Group in the pGroupsSubset, loop through target dimension add an corresponding element to a group specific subset if
    a. the element is part of the }ElementSecurity Filter cube (specified by pFilterBySecurityCube) AND
    b. the element is an ancestor of the StdCostCenterHierRoot AND
    c. the group has READ access in the }ElementSecurity Cube (specified by pFilterBySecurityCube)
4) Loop through subset created in (3) and add all elements to the element created in (2)
5) Remove duplicates from hierarchy underneath node/element created in (2)
6) Create element specified by pGroupHierarchyRootElement
7) Add the pGroupHierarchyRootElement as a component to the APEXOfStdCostCenterHier
8) Remove subset created in (3)

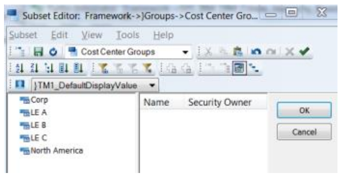
For the example configuration in the screenshot of the process parameters below, the resulting cost center hierarchies will look like:


Subset Editor: Framework->Cost Center->Default [Pu...
Subset Edit View Tools Help
Default
Name Securit
OK
Cancel
Rollup

Cost Center Access Groups
- Corp Cost Centers
- LE A Cost Centers
- LE B Cost Centers
- LE C Cost Centers
- North America Cost Centers

Cost Centers by Legal Entity
- Total Company
  - Holding Company
  - North America

based on groups subset


Subset Editor: Framework->}Groups->Cost Center Gro...
Subset Edit View Tools Help
Cost Center Groups
}TM1_DefaultDisplayValue
Name Security Owner
OK
Cancel

- Corp
- LE A
- LE B
- LE C
- North America

3.2 Maintenance of Dependent Dimensions

Use }ElementAttributes_TM1_Dimension.cub to specify & manage how a dependent dimension is derived from its master dimension:

Attributes:

- Is Dependent Dimension: set to Y (example: Attr. 'is Dependent Dimension' for Dimension Element 'Cost Center Group' is set to Y)
- Master Dimension: set to the master dimension (example: Attr. 'Master Dimension' for Dimension Element 'Cost Center Group' is set to Y)
- Hierarchies: the Master Dimension Hierarchies – defined by their Hierarchy APEX (top or root) node, separated by a semicolon (example: Attr. 'Hierarchies' for Dimension Element 'Cost Center Group' could be set to 'Cost Center Access Groups; Cost Centers by Legal Entity;', meaning the 2 corresponding hierarchies are to be used in dependent dimension 'Cost Center Group')
- Hierarchy Level Attribute: Dimension Element Attribute to use to identify the hierarchy level of a dimension element (example: Attr. 'Hierarchy Level Attribute' for Dimension Element 'Cost Center Grouo' is set to 'Hierarchy Level', meaning the values of Master Dimension attribute 'Hierarchy Level' will be used for building the dependent hierarchy based on Start Level and End Level information (see below))
- Start Level: Hierarchy Start Level as defined by value of Attribute in 'Hierarchy Level Attribute'. The dependent dimension will contain all elements between 'Start Level' and 'End Level' for the Hierarchies defined by the 'Hierarchy Level Attribute'
- End Level: Hierarchy End Level as defined by value of Attribute in 'Hierarchy Level Attribute'
- Weight for Descendants of APEX Nodes: Weight of immediate children of the hierarchy apex nodes.

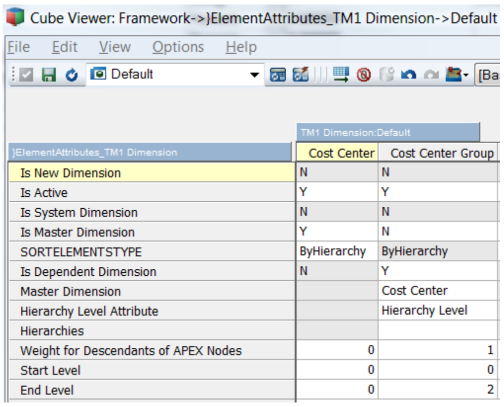
To create of update a dependent dimension (based on settings in '}Elementattributes_TM1 Dimension.cub' or based on manual configurations), run process '**Manage Dimension - Create or Update Dependent Dimension.pro**'**:**

- pTargetDimension: Dependent Dimension = Target Dimension to be updated
- pSourceDimension: Source Dimension = Master Dimension; if empty, then Master Dimension will be queried from '}ElementAttributes_TM1 Dimension.cub'
- pOverwriteExistingDimension: set to Y to overwrite dimension entirely; if set to N or not Y, all C-level elements will be deleted prior to re-build of dimension;
- pStartLevel: ; if empty, then Start Level will be queried from '}ElementAttributes_TM1 Dimension.cub'
- pEndLevel: ; if empty, then End Level will be queried from '}ElementAttributes_TM1 Dimension.cub'
- pApplyCustomWeightForDescendantsOfApexNode: set to Y to apply a custom default weight for descendants of the Apex node, otherwise, the weight as defined in '}ElementAttributes_TM1 Dimension.cub' will be used
- pWeightForDescendantsOfApexNode: custom weight to use if pApplyCustomWeightForDescendantsOfApexNode = Y
- pHiearchyLevelAttribute: Attribute to use to query hierarchy levels to determine start and end level; if empty, then Attr. will be queried from '}ElementAttributes_TM1 Dimension.cub'
- pLogging: debug logging (leave at N)
- pHierarchyName: name of the hierarchies to update in the dependent dimension (separate multiple hierarchies with a semicolon (;); if empty, then the Hiearchies will be queried from '}ElementAttributes_TM1 Dimension.cub' (Attr. 'Hiearchies'), if no hierarchies are defined there, all hierarchies will be used.
- pOverwriteCLevelsOnly: If set to Y and if pOverwriteExistingDimension = N, only C-Level elements will be deleted. If set to N and if pOverwriteExistingDimension = N, no Elements will be deleted in the dependent dimension (note: you can use processes SYS_IBM_Subset_Create.pro and SYS_IBM_DIM_Delete_Elements_according_to_Subset.pro to delete a specific C-Level hierarchy from the dimension prior to update/refresh)

After updating the dimension elements and hierarchies, process 'Manage Dimension - Create or Update Dependent Dimension.pro' will

- update element attributes via sub-process via sub-process Manage Dimension - Create or Update Dependent Dimension - Populate Attributes.pro (copy attribute values from master dimension }ElementAttributes cube to the }ElementAttributes cube of the dependent dimension)
- Update the 'Hierarchy Level' attribute values of the dependent dimension via sub-process SYS_IBM_DIM_Update_Hiearchy_Level_Attributes.pro (starting with level 0 at the root, the level attribute values are adjusted such that the levels are continuous from 0 to N).

Example: Based on the following configuration for dependent dimension 'Cost Center Group'

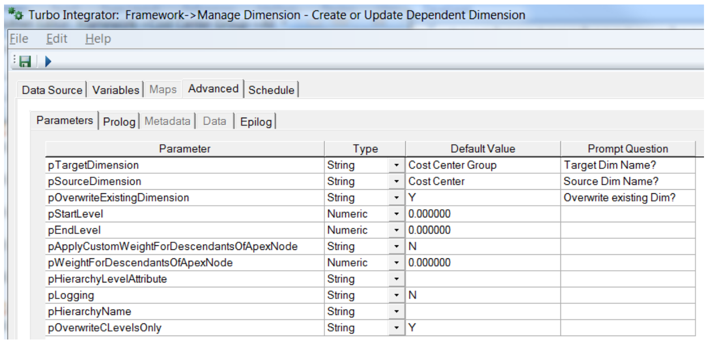


Cube Viewer: Framework->}ElementAttributes_TM1 Dimension->Default

TM1 Dimension:Default

}ElementAttributes_TM1 Dimension	Cost Center	Cost Center Group
Is New Dimension	N	N
Is Active	Y	Y
Is System Dimension	N	N
Is Master Dimension	Y	N
SORTELEMENTSTYPE	ByHierarchy	ByHierarchy
Is Dependent Dimension	N	Y
Master Dimension		Cost Center
Hierarchy Level Attribute		Hierarchy Level
Hierarchies		
Weight for Descendants of APEX Nodes	0	1
Start Level	0	0
End Level	0	2

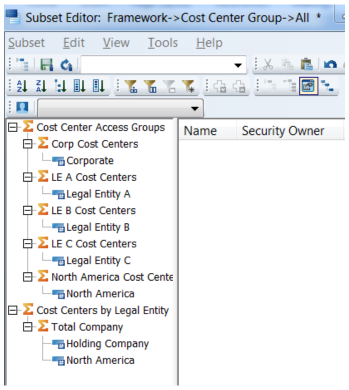
Running the process 'Manage Dimension – Create or Update Dependent Dimension' for dimension 'Cost Center Group'



Turbo Integrator: Framework->Manage Dimension - Create or Update Dependent Dimension

Parameter	Type	Default Value	Prompt Question
pTargetDimension	String	Cost Center Group	Target Dim Name?
pSourceDimension	String	Cost Center	Source Dim Name?
pOverwriteExistingDimension	String	Y	Overwrite existing Dim?
pStartLevel	Numeric	0.000000	
pEndLevel	Numeric	0.000000	
pApplyCustomWeightForDescendantsOfApexNode	String	N	
pWeightForDescendantsOfApexNode	Numeric	0.000000	
pHierarchyLevelAttribute	String		
pLogging	String	N	
pHierarchyName	String		
pOverwriteCLevelsOnly	String	Y	

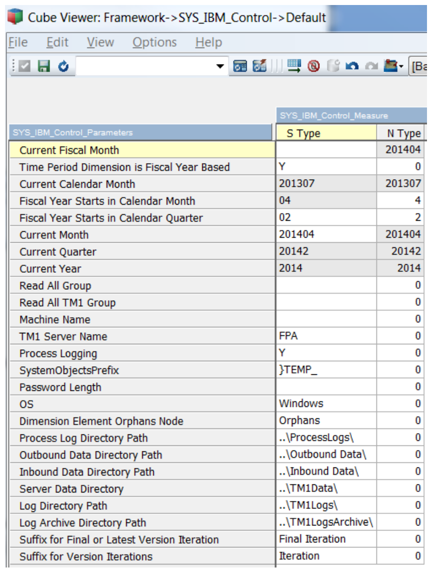
Resulted in the following elements & hierarchies in 'Cost Center Group':


Subset Editor: Framework->Cost Center Group->All *
Subset Edit View Tools Help
Name
Security Owner

```
Cost Center Access Groups
    Corp Cost Centers
        Corporate
    LE A Cost Centers
        Legal Entity A
    LE B Cost Centers
        Legal Entity B
    LE C Cost Centers
        Legal Entity C
    North America Cost Cente
        North America
Cost Centers by Legal Entity
    Total Company
        Holding Company
        North America
```

4. System Control Cubes

4.1 SYS IBM Control


Cube Viewer: Framework->SYS_IBM_Control->Default

SYS_IBM_Control_Parameters	S Type	N Type
Current Fiscal Month		201404
Time Period Dimension is Fiscal Year Based	Y	0
Current Calendar Month	201307	201307
Fiscal Year Starts in Calendar Month	04	4
Fiscal Year Starts in Calendar Quarter	02	2
Current Month	201404	201404
Current Quarter	20142	20142
Current Year	2014	2014
Read All Group		0
Read All TM1 Group		0
Machine Name		0
TM1 Server Name	FPA	0
Process Logging	Y	0
SystemObjectsPrefix	}TEMP_	0
Password Length		0
OS	Windows	0
Dimension Element Orphans Node	Orphans	0
Process Log Directory Path	..\ProcessLogs\	0
Outbound Data Directory Path	..\Outbound Data\	0
Inbound Data Directory Path	..\Inbound Data\	0
Server Data Directory	..\TM1Data\	0
Log Directory Path	..\TM1Logs\	0
Log Archive Directory Path	..\TM1LogsArchive\	0
Suffix for Final or Latest Version Iteration	Final Iteration	0
Suffix for Version Iterations	Iteration	0

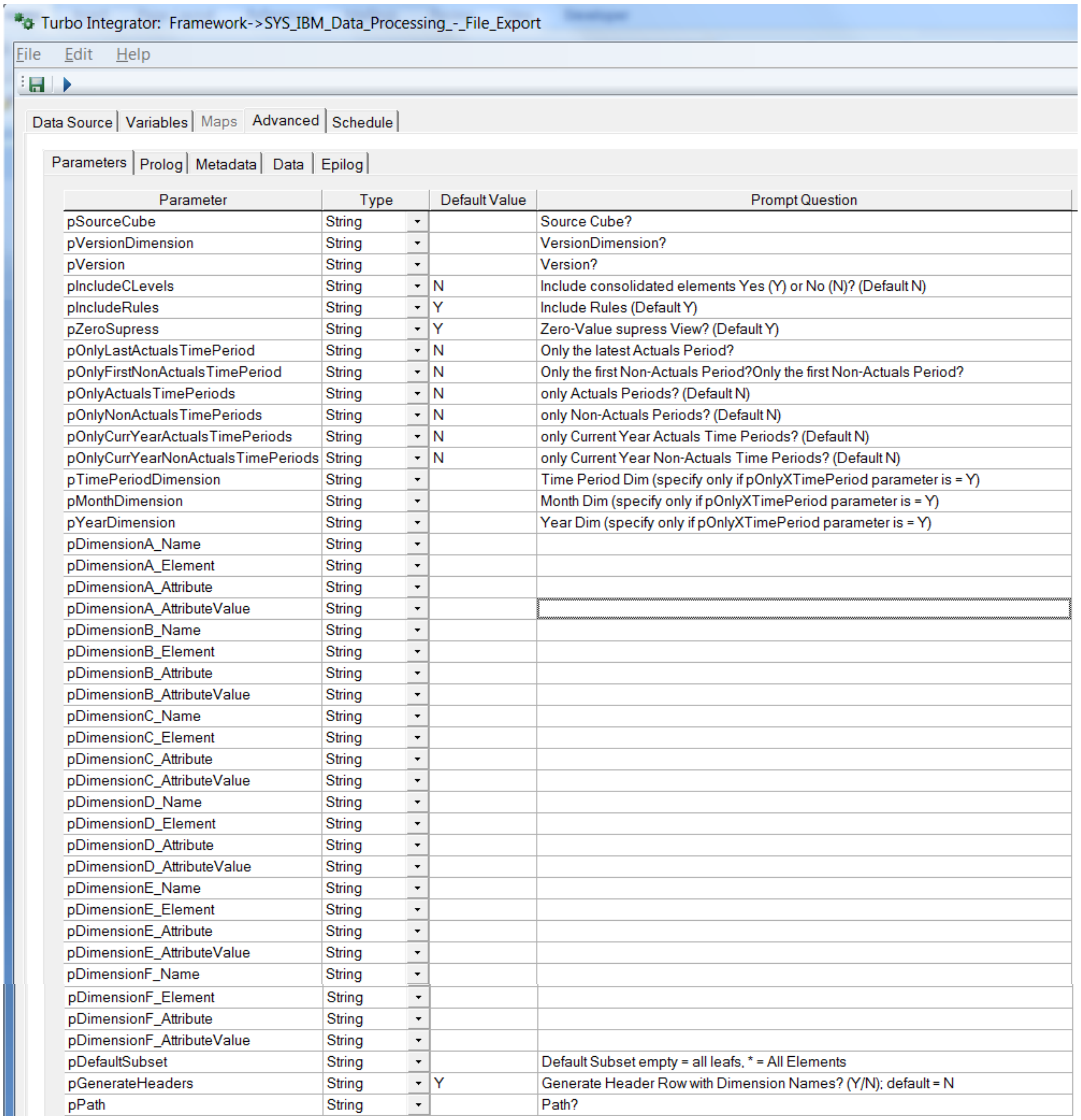
- Read All Group: Security Staging Group used for READ ALL access of Elements, not used here
- Read All TM1 Group: CAM / TM1 Security Group used for READ ALL access of Elements, not used here
- Machine Name: TM1 machine name, not used here
- TM1 Server Name: TM1 DB name, not used here
- Process Logging: set to Y to enable debug logging specific processes that determine logging based on this SYS_IBM_Control parameter
- SystemObjectsPrefix: prefix for system ('SYS_IBM*' etc.) subsets and views
- PasswordLength: not used here
- OS: Used to determine what type of scripts so use for command line script execution, not used here
- Dimension Element Orphans Node: orphans node to use when adding elements during data load, i.e. new elements will be added and attached to 'orphans' node

- **Process Log Directory Path: path to use for process debug or info logging (custom logging)**
- **Outbound Data Directory Path: path to use for data export (export of data from TM1)**
- **Inbound Data Directory Path: path to use when importing data (pick up of flat files etc.)**
- **Server Data Directory: server data directory**
- **Log Directory Path: server log directory**

- Log Archive Directory Path: directory to use when archiving log files, not used here
- Suffix for Final or Latest Version Iteration: used for Planning/FCSTing ONLY, not used here
- Suffix for Version Iteration: used for Planning/FCSTing ONLY, not used here
- Current Month, Current Quarter, Current Year: used for Time Period maintenance and other processes and rules to determine the current Month, Quarter, Year. Input into Current Month; current QTR and Yr are automatically calculated, not used here

5. Exporting Data

Use process 'SYS_IBM_Data_Processing_-_File_Export.pro' to export data from a cube to a file:



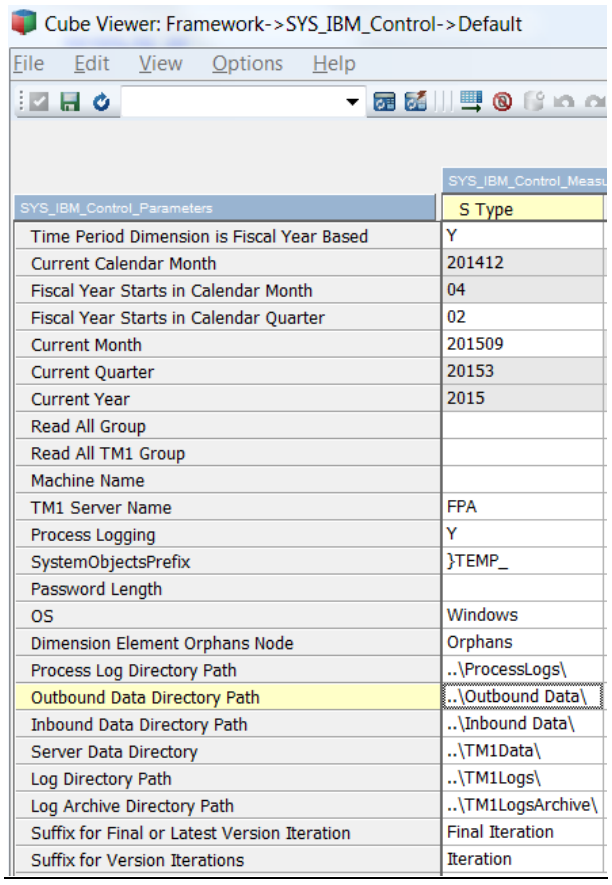
Turbo Integrator: Framework->SYS_IBM_Data_Processing_-_File_Export
File Edit Help
Data Source | Variables | Maps | Advanced | Schedule
Parameters | Prolog | Metadata | Data | Epilog

Parameter	Type	Default Value	Prompt Question
pSourceCube	String		Source Cube?
pVersionDimension	String		VersionDimension?
pVersion	String		Version?
pIncludeCLevels	String	N	Include consolidated elements Yes (Y) or No (N)? (Default N)
pIncludeRules	String	Y	Include Rules (Default Y)
pZeroSupress	String	Y	Zero-Value supress View? (Default Y)
pOnlyLastActualsTimePeriod	String	N	Only the latest Actuals Period?
pOnlyFirstNonActualsTimePeriod	String	N	Only the first Non-Actuals Period?Only the first Non-Actuals Period?
pOnlyActualsTimePeriods	String	N	only Actuals Periods? (Default N)
pOnlyNonActualsTimePeriods	String	N	only Non-Actuals Periods? (Default N)
pOnlyCurrYearActualsTimePeriods	String	N	only Current Year Actuals Time Periods? (Default N)
pOnlyCurrYearNonActualsTimePeriods	String	N	only Current Year Non-Actuals Time Periods? (Default N)
pTimePeriodDimension	String		Time Period Dim (specify only if pOnlyXTimePeriod parameter is = Y)
pMonthDimension	String		Month Dim (specify only if pOnlyXTimePeriod parameter is = Y)
pYearDimension	String		Year Dim (specify only if pOnlyXTimePeriod parameter is = Y)
pDimensionA_Name	String		
pDimensionA_Element	String		
pDimensionA_Attribute	String		
pDimensionA_AttributeValue	String		
pDimensionB_Name	String		
pDimensionB_Element	String		
pDimensionB_Attribute	String		
pDimensionB_AttributeValue	String		
pDimensionC_Name	String		
pDimensionC_Element	String		
pDimensionC_Attribute	String		
pDimensionC_AttributeValue	String		
pDimensionD_Name	String		
pDimensionD_Element	String		
pDimensionD_Attribute	String		
pDimensionD_AttributeValue	String		
pDimensionE_Name	String		
pDimensionE_Element	String		
pDimensionE_Attribute	String		
pDimensionE_AttributeValue	String		
pDimensionF_Name	String		
pDimensionF_Element	String		
pDimensionF_Attribute	String		
pDimensionF_AttributeValue	String		
pDefaultSubset	String		Default Subset empty = all leafs, * = All Elements
pGenerateHeaders	String	Y	Generate Header Row with Dimension Names? (Y/N); default = N
pPath	String		Path?

If the process is run by only specifying a source cube and without changing the other parameters and parameter defaults, all leaf level data from the cube is exported. By specifying dimension elements and subsets more specifically, the process also allows the export of specific data sections/portions.

'**SYS_IBM_Data_Processing_-_File_Export.pro**' process parameters:

- pSourceCube: name of source cube
- pPath: export file path, including last '\'; if not specified, the path set in control cube 'SYS_IBM_Control', measure 'Outbound Data Directory Path' will be used:



SYS_IBM_Control_Parameters	S Type
Time Period Dimension is Fiscal Year Based	Y
Current Calendar Month	201412
Fiscal Year Starts in Calendar Month	04
Fiscal Year Starts in Calendar Quarter	02
Current Month	201509
Current Quarter	20153
Current Year	2015
Read All Group	
Read All TM1 Group	
Machine Name	
TM1 Server Name	FPA
Process Logging	Y
SystemObjectsPrefix	}TEMP_
Password Length	
OS	Windows
Dimension Element Orphans Node	Orphans
Process Log Directory Path	..\ProcessLogs\
Outbound Data Directory Path	..\Outbound Data\
Inbound Data Directory Path	..\Inbound Data\
Server Data Directory	..\TM1Data\
Log Directory Path	..\TM1Logs\
Log Archive Directory Path	..\TM1LogsArchive\
Suffix for Final or Latest Version Iteration	Final Iteration
Suffix for Version Iterations	Iteration

Note: the export file name will be automatically determined:

<TM1ClientID> & '_Export_' & <SourceCube> | '_' | <Version/Scenario> | <DateTimeStamp> | '.cma'

Export File Format:

- Comma Separated File (csv)
- Order: Dimension 1 Element, Dimension 2 Element, …, Dimension N Element, Value
- pASCIIQuoteCharacter = empty (none), i.e. no ASCII Character used to enclose strings/element names

- pIncludeRules: include rule driven data in export Y/N ?
- pIncludeCLevels: include consolidated elements in export Y/N? (if set to Y, requires that subsets contain C-Level elements)
- pGenerateHeaders: set to Y to have the export file generate column headers in the first row. The column headers will be generated based on the corresponding dimension name.
- pZeroSupress: Y/N
- pDefaultSubset: empty = a leafs; * = All

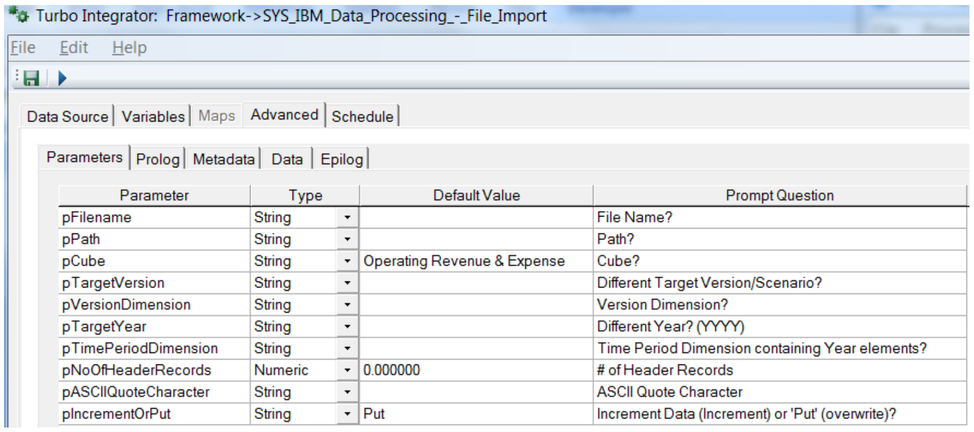
- pVersionDimension: Version Dimension*
- pVersion: Version*
- pTimePeriodDimension: Name of YYYYMM dimension if applicable*
- pMonthDimension: Name of MM dimension if applicable*
- pYearDimension: Name of YYYY dimension if applicable*
- pOnlyLastActualsTimePeriod: View/Subsets to contain last Actuals time periods only Y/N?**
- pOnlyFirstNonActualsTimePeriod: View/Subsets to contain first non-Actuals time period only Y/N?**
- pOnlyActualsTimePeriods: View/Subsets to contain only Actuals time periods Y/N?**
- pOnlyNonActualsTimePeriods: View/Subsets to contain Non-Actuals time periods only Y/N?**
- pOnlyCurrYearActualsTimePeriods: View/Subsets to contain current Year Actuals time periods only Y/N?**
- pOnlyCurrYearNonActualsTimePeriods: View/Subsets to contain current Year non-Actuals time periods only Y/N?**

*: used for Version dependent Time Period Filtering
**:based on Version dimension attribute 'Actuals Through Date'

- pDimensionX_Name
- pDimensionX_Element
  - <ElementName> = single Element Name
  - All or * = SubsetIsAllSet = all elements in dimension
  - '' (empty) = all N-level Elements (DEFAULT)
  - A;<ElementName> or AD;<ElementName> = All Descendants
  - ND;<ElementName> = All N-Level Descendants (excluding Parent(s))
  - IC;<ElementName> = Immediate Children
  - CD;<ElementName> = C-Level Descendants (including Parent)
  - Multi:<ElementName1>;<ElementName2>;...;<ElementNameN> = Multipe Elements
- pDimensionX_Attribute: attribute by which to filter
- pDimensionX_AttributeValue: value of attribute by which to filter ...
- pLogging: Y/N for debug logging

6. Rapid File (Re-)Import

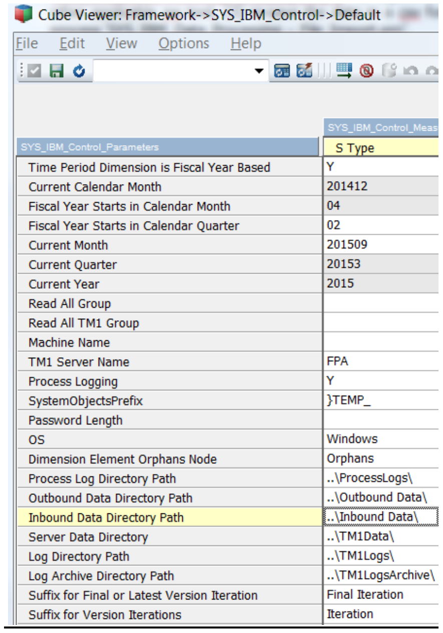
Using process 'SYS_IBM_Data_Processing_-_File_Import.pro', a csv file (such as files created by the process 'SYS_IBM_Data_Processing_-_File_Import.pro') can directly be imported into a cube for as long as the file columns correspond to the dimensions (and dimension order) in the target cube and the numeric or text values are contained in the last column.
The file export/import functionality can also be used for changing smaller data sets: (1) use process 'SYS_IBM_Data_Processing_-_File_Export.pro' to export the data, (2) modify the data in Excel or any other applicable sw tool, (3) re-save the data in a csv format & (4) re-import the data using process process 'SYS_IBM_Data_Processing_-_File_Import.pro'.


Turbo Integrator: Framework->SYS_IBM_Data_Processing_-_File_Import
File Edit Help
Data Source | Variables | Maps | Advanced | Schedule
Parameters | Prolog | Metadata | Data | Epilog

Parameter	Type	Default Value	Prompt Question
pFilename	String		File Name?
pPath	String		Path?
pCube	String	Operating Revenue & Expense	Cube?
pTargetVersion	String		Different Target Version/Scenario?
pVersionDimension	String		Version Dimension?
pTargetYear	String		Different Year? (YYYY)
pTimePeriodDimension	String		Time Period Dimension containing Year elements?
pNoOfHeaderRecords	Numeric	0.000000	# of Header Records
pASCIIQuoteCharacter	String		ASCII Quote Character
pIncrementOrPut	String	Put	Increment Data (Increment) or 'Put' (overwrite)?

'**SYS_IBM_Data_Processing_-_File_Import.pro**' process parameters:

- pFilename: name of csv file (including file extension)
- pPath: file path, including last '\'; if not specified, the path set in control cube 'SYS_IBM_Control', measure 'Inbound Data Directory Path' will be used:


Cube Viewer: Framework->SYS_IBM_Control->Default
File Edit View Options Help

SYS_IBM_Control_Parameters	S Type
Time Period Dimension is Fiscal Year Based	Y
Current Calendar Month	201412
Fiscal Year Starts in Calendar Month	04
Fiscal Year Starts in Calendar Quarter	02
Current Month	201509
Current Quarter	20153
Current Year	2015
Read All Group	
Read All TM1 Group	
Machine Name	
TM1 Server Name	FPA
Process Logging	Y
SystemObjectsPrefix	}TEMP_
Password Length	
OS	Windows
Dimension Element Orphans Node	Orphans
Process Log Directory Path	..\ProcessLogs\
Outbound Data Directory Path	..\Outbound Data\
Inbound Data Directory Path	..\Inbound Data\
Server Data Directory	..\TM1Data\
Log Directory Path	..\TM1Logs\
Log Archive Directory Path	..\TM1LogsArchive\
Suffix for Final or Latest Version Iteration	Final Iteration
Suffix for Version Iterations	Iteration

- pCube: Cube to import data into
- pTargetVersion: Version to import into (for seeding etc.; example: export 2014 Actuals, increase by 10% and import into 2015 Plan)
- pVersionDimension: Version/Scenario dimension
- pTargetYear: Year to import into (for seeding; example: export 2014 Actuals, increase by 10% and import into 2015 Plan)
- pTimePeriodDimension: Time Period Dimension
- pNoOfHeaderRecords: # of header records in file
- pASCIIQuoteCharacter: ASCII Character used to enclose strings/element names (typically " or empty)
- pIncrementOrPut: Increment values (Increment), i.e. if an intersection as a value of 10 and the import file has a value of 1 for that intersection, the value will be increased to 10+1=11? Or 'overwrite' values (Put), where the intersection will be overwritten with value 1?

7. TM1 Framework Objects

Framework Objects