



Exporting Data from Essbase Using Report Script

Description:

BISP is committed to provide BEST learning material to the beginners and advance learners. In the same series, we have prepared a complete end-to end Hands-on Beginner's Guide for Oracle Hyperion Essbase, Planning, HFM, FDQM, FR. The document shows you how to extract data from one Essbase cube using Report Script and Load into another cube. **Join our professional training program and learn from experts.**

History:

Version
0.1

Description Change
Review#1

Author
Amit Sharma

Publish Date
29th Nov 2012

Contents

..... 2

Contents..... 2

Introduction: 3

Example#1 5

Example#2 6

Example#3 7

Example#4 8

Example#5 9



Introduction:

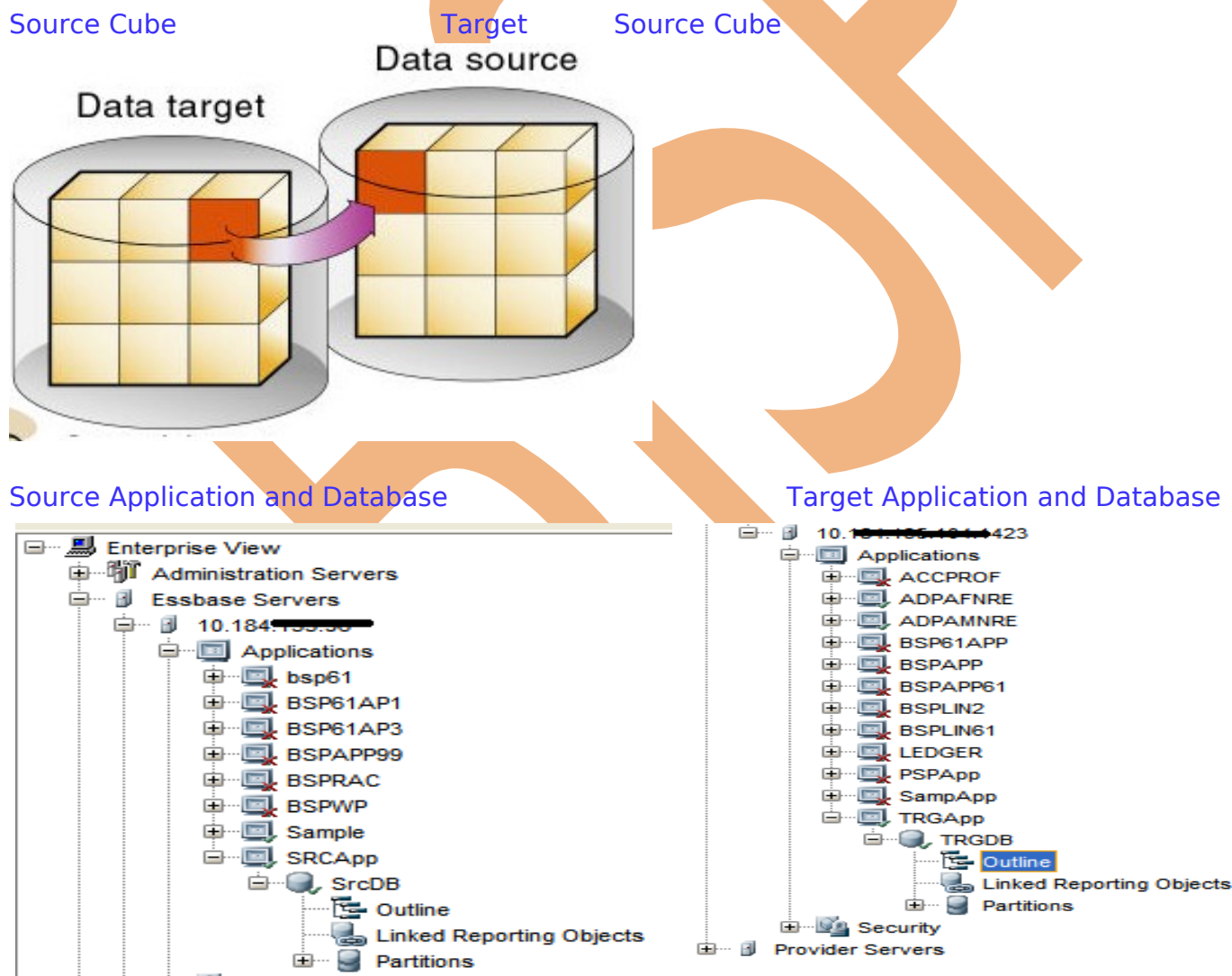
Moving Data between Essbase Cubes: There may be situations where you require to migrate the data from one Essbase cube to another Essbase cube. One of the possible ways is to do this job by using report script. Report script is an excellent approach to exact data subset from Essbase cube. We have used couple of examples to demonstrate how can be achieved. Below are the possible ways to export data from Essbase, in this document we are going to export data using Report Script.

1. Smart View "Working with Query Designer"
2. Exporting Data Using the DATAEXPORT Command
3. Exporting Data Using MaxL
4. Exporting Text Data Using Calculation Scripts

5. Exporting Text Data Using Report Scripts

In order to demonstrate this example, we have two Essbase servers (one is source and another one is target).

Note : Both source and target data source should have same outline structure.



Source Cube

Target Cube

Outline Editor: [10.184.155.36:SRCAApp.SrcDB]

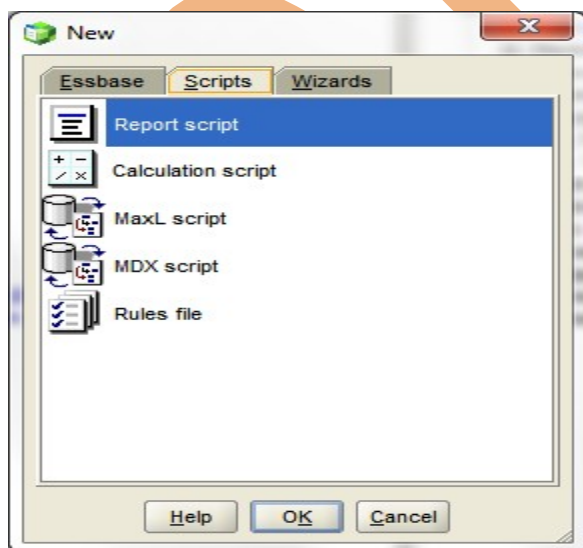
Outline: SrcDB (Active Alias Table: Default)

- Year Time <4> (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
- GL_Accounts Accounts <4> (Label Only)
 - Profit (+) <2> (Dynamic Calc)
 - Account_Type (+) <5> (Dynamic Calc)
 - Inventory (-) <3> (Label Only)
 - Ratios (-) <3> (Label Only)
- Product <4>
 - F&B (+) <3> (Alias: Colas)
 - Electronics (+) <4> (Alias: Root Beer)
 - HeavyElectricals (+) <3> (Alias: Cream Soda)
 - Cosmetics (+) <3> (Alias: Fruit Soda)
- Gep_Region <4>
 - NA_East (+) <5> (UDAS: Major Market)
 - CA_West (+) <5>
 - South_Center (+) <4> (UDAS: Small Market)
 - Central (+) <6> (UDAS: Major Market)
- Scenario <4> (Label Only)
 - Actual (+)
 - Budget (-)
 - Variance (-) (Dynamic Calc) (Two Pass) [Formula: @VAR(Actual, Budget);]
 - Variance % (-) (Dynamic Calc) (Two Pass) [Formula: @VARPER(Actual, Budget);]

Outline Editor: [10.184.155.36:TRGApp.TRGDB]

Outline: TRGDB (Active Alias Table: Default)

- Year Time <4> (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Dynamic Calc)
- GL_Accounts Accounts <4> (Label Only)
 - Profit (+) <2> (Dynamic Calc)
 - Account_Type (+) <5> (Dynamic Calc)
 - Inventory (-) <3> (Label Only)
 - Ratios (-) <3> (Label Only)
- Product <4>
 - F&B (+) <3> (Alias: Colas)
 - Electronics (+) <4> (Alias: Root Beer)
 - HeavyElectricals (+) <3> (Alias: Cream Soda)
 - Cosmetics (+) <3> (Alias: Fruit Soda)
- Gep_Region <4>
 - NA_East (+) <5> (UDAS: Major Market)
 - CA_West (+) <5>
 - South_Center (+) <4> (UDAS: Small Market)
 - Central (+) <6> (UDAS: Major Market)
- Scenario <4> (Label Only)
 - Actual (+)
 - Budget (-)
 - Variance (-) (Dynamic Calc) (Two Pass) [Formula: @VAR(Actual, Budget);]
 - Variance % (-) (Dynamic Calc) (Two Pass) [Formula: @VARPER(Actual, Budget);]



Example#1

Extracting data from Source Data Cube and load into target using different formatting options.

<PAGE(Scenario)

<COLUMN(Year)

<ROW (Gep_Region, Product, GL_Accounts)

<CHILDREN "NA_East"

<IDESCENDANTS "F&B"

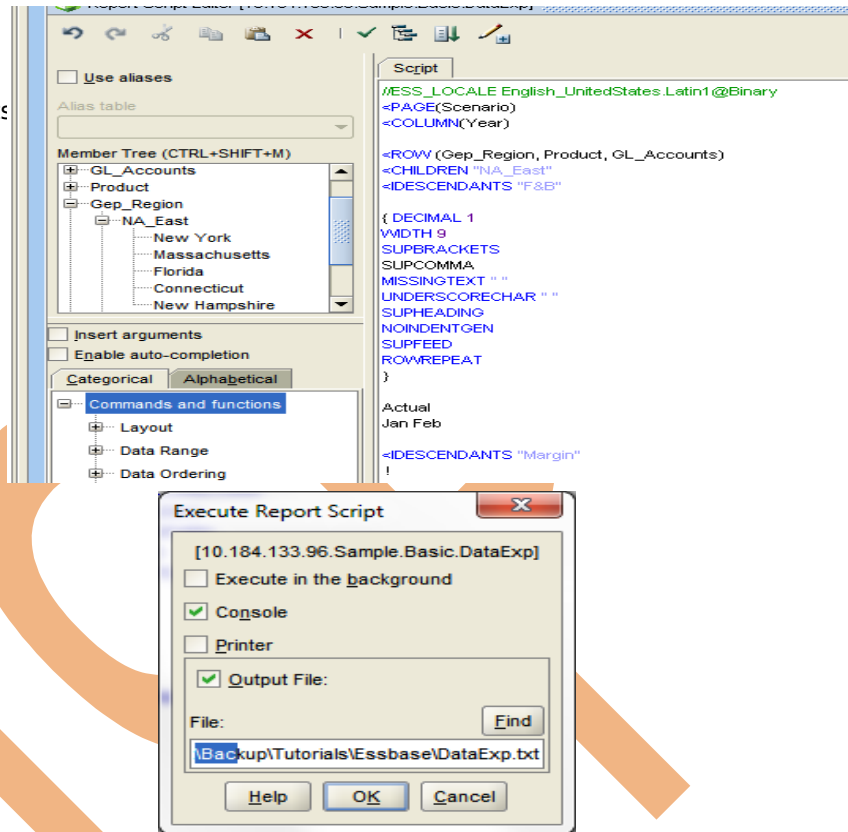
```
{  
  DECIMAL 1  
  WIDTH 9  
  SUPBRACKETS  
  SUPCOMMA  
  MISSINGTEXT " "  
  UNDERSCORECHAR " "  
  SUPHEADING  
  NOINDENTGEN  
  SUPFEED  
  ROWREPEAT  
}
```

Actual

Jan Feb

<IDESCENDANTS "Margin"

!

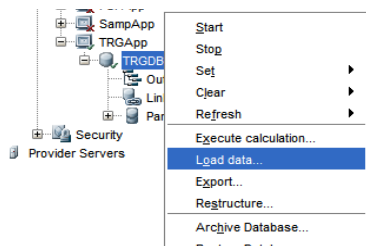


Output

DataExp1 - Notepad

File	Edit	Format	View	Help
New York	100-10	Sales	100.0	220.0
New York	100-10	COGS	223.0	234.0
New York	100-10	Margin	-223.0	-234.0
New York	100-20	Sales	233.0	340.0
New York	100-20	COGS	534.0	453.0
New York	100-20	Margin	-534.0	-453.0
New York	100-30	Sales	54.0	354.0
New York	100-30	COGS	53.0	543.0
New York	100-30	Margin	-53.0	-543.0
New York	F&B	Sales		

Explanation: Above script displays <PAGE(Scenario) Actual in Page Layout, <COLUMN(Year) Jan, Feb in column Layout, <ROW (Gep_Region, Product, GL_Accounts) Region="NA_East" Product = IDESCENDANTS "F&B" and Measures as GL_Accounts Margin. The data file has been created in such a way that, it can be loaded to target system application database. In order to ensure that is compatible with target cube, multiple formatting options are used (SUPHEADING SUPCOMMA). The final step is to load data into target cube(if require the dataload rule file is to be created)



Example#2

Extracting data using Attribute Dimension.

The screenshot displays the Essbase Report Script Editor interface. On the left, the Outline Editor shows a hierarchy where 'Consumer_Type' is an attribute dimension. The Member Tree (CTRL+SHIFT+M) shows 'Kids' as a member of 'Consumer_Type'. The Script window shows the use of the <ATTRIBUTE function to filter data by 'Kids'.

Outline Editor:

- Outline: SrcDB (Active Alias Table: Default)
 - Year Time <4> (Active Dynamic Time Series Members: H-T-D, Q-T-D) (Label Only)
 - GL_Accounts Accounts <4> (Label Only)
 - Product <4> (Consumer_Type)
 - F&B (+) <3> (Alias: Colas)
 - 100-10 (+) (Alias: Cola) (Consumer_Type: Kids)
 - 100-20 (+) (Alias: Diet Cola)
 - 100-30 (+) (Alias: Caffeine Free Cola)
 - Electronics (+) <4> (Alias: Root Beer)
 - 200-10 (+) (Alias: Old Fashioned) (Consumer_Type: Kids)
 - 200-20 (+) (Alias: Diet Root Beer) (Consumer_Type: Kids)
 - 200-30 (+) (Alias: Sasparilla) (Consumer_Type: Kids)
 - 200-40 (+) (Alias: Birch Beer)
 - HeavyElectricals (+) <3> (Alias: Cream Soda)
 - Cosmetics (+) <3> (Alias: Fruit Soda)
 - Gep_Region <4>
 - Scenario <4> (Label Only)
 - Consumer_Type Attribute [Type: Text] <3>
 - Kids
 - Ladies
 - Gentlemen

Member Tree (CTRL+SHIFT+M):

- SrcApp SrcDB
 - Year
 - GL_Accounts
 - Product
 - Gep_Region
 - Scenario
 - Consumer_Type
 - Attribute Calculations
 - Kids

Script:

```

/ESS_LOCALE English_UnitedStates.Lat
<PAGE(Scenario,Gep_Region,Year)
Actual
"New York"
Jan
<COLUMN(GL_Accounts)
Sales
<ATTRIBUTE "Kids"
{ DECIMAL 1 WMDTH 9 }

```

Output Window:

Actual New York Jan Sales		
100-10	120	
200-10	231	
200-20	233	
200-30	219	

Explanation: In the above example, we are accessing the products based on Attribute Dimension Name Consumer_Type. We want only those products to be migrated to target cube which are consumed by Kids. It looks to the end use as if attribute is a real dimension on one hand, on another it doesn't add any overheads to the Essbase cube size. You can see in the output window, Product "100-10" is visible, as it is associated with attribute dimension member "Kids".

ATTRIBUTE

Returns all base-dimension members associated with a specified attribute.

Syntax

<ATTRIBUTE attMbrName

Example#3

Exacting Top 5 Products from source cube and load into target.

Report Script Editor [10.184.133.96.SRCApp.SrcDB.Example2]

Agency FB

Actual New York Jan Sales

200-20	233
200-40	232
200-10	231

Script

```
//ESS_LOCALE English_UnitedStates.Latin1@Binary
<PAGE(Scenario,Gep_Region,Year)
Actual
"New York"
Jan
<COLUMN(GL_Accounts)
Sales
<ROW ( Product)
Lev0,Product
{ DECIMAL 1 WIDTH 9 }
<TOP(3,@Datacol(1))
!
```

Member Tree (CTRL+SHIFT+M)

SRCAApp.SrcDB

- Year
- GL_Accounts
- Product
- Gep_Region
- Scenario
- Consumer_Type
- Attribute Calculations

//ESS_LOCALE English_UnitedStates.Latin1@Binary

<PAGE(Scenario,Gep_Region,Year)

Actual

"New York"

Jan

<COLUMN(GL_Accounts)

Sales

<ROW (Product)

Lev0,Product

{ DECIMAL 1 WIDTH 9 }

<TOP(3,@Datacol(1))

!

Explanation: In the above example, we are extracting Top 3 products based on Sales. Many of time due to high load during peak business hours, you like to migrate the data into another cube which are extensively being used. This can be done by using Top function. As you see in this example, the Top function takes two parameters, number of elements and Data Column position number.

TOP

Returns rows with the highest values of a specified data column.

Syntax

<TOP ([<rowgroupDimension>,<rows>,<column>)

Example#4

Extracting data source cube and load into target using Restrict(Where) Clause.

The screenshot shows the Report Script Editor interface. On the left, the Outline Editor displays a hierarchical tree of data sources. The main window shows the Report Script Editor with the following script:

```

!
/MESS_LOCALE English_UnitedStates.Latin1@
<PAGE(Scenario,Gep_Region,Year)
Actual
"New York"
Jan
<COLUMN(GL_Accounts)
Sales
<ROW ( Product)
Lev0,Product
<RESTRICT (@Datacol(1)>0)
{ DECIMAL 1 WIDTH 9 }
!

```

The output table, titled "Actual New York Jan Sales", shows the following data:

Product	Sales
100-10	120
100-20	210
100-30	221
200-10	231
200-20	233
200-30	219
200-40	232

<PAGE(Scenario,Gep_Region,Year)

Actual
"New York"
Jan

<COLUMN(GL_Accounts)

Sales

<ROW (Product)

Lev0,Product

<RESTRICT (@Datacol(1)>0)

{ DECIMAL 1 WIDTH 9 }

!

Explanation: In the above example, we are extracting data based on condition, in report script the condition can be applied using RESTRICT Command, it works exactly as where clause in SQL query, the difference lies it works for multi dimensional view. The 1st argument is Datacol position, conditional operator and value @Datacol(1)>0

RESTRICT

The RESTRICT command specifies the conditions that the row must satisfy before it becomes part of a result set.

Syntax

<RESTRICT (<column | value> <operator> <column | value> {<logicalOperator><column | value> <operator> <column | value> })

Example#5

Exacting data source cube and load into target using Restrict(Where) Clause.

Script

```
//ESS_LOCALE English_UnitedStates.Latin1@Binary
<PAGE (GL_Accounts, Product)
Sales
"100-10"

<COLUMN (Scenario, Year)
Actual
Qtr1 Qtr2

{CALCULATE COLUMN "Half Yr Sales" = 1 + 2
WIDTH 27 WIDTH 20 20 21
}

<ROW (Gep_Region)
<CHILD "NA_East"
!
```

Agency FB

Sales 100-10 Actual

	Qtr1	Qtr2	Half Yr Sales
New York	120	464	584
Massachusetts	#Missing	1,248	1,248
Florida	#Missing	1,244	1,244
Connecticut	#Missing	1,223	1,223
New Hampshire	#Missing	1,143	1,143
NA_East	120	#Missing	120

Example#5 Calculation

<PAGE (GL_Accounts, Product)

Sales
"100-10"

<COLUMN (Scenario, Year)

Actual
Qtr1 Qtr2

```
{CALCULATE COLUMN "Half Yr Sales" = 1 + 2
WIDTH 27 WIDTH 20 20 21
}
```

<ROW (Gep_Region)

<CHILD "NA_East"
!

Explanation: In the above example, we have used column calculation function. The source outline Time Dimension hierarchy doesn't have "Half year", however the target cube may have. Therefore, we want to calculate the Half Year on the fly and load to target cube. The calculate column function takes "Label" **"Half Yr Sales"** and Expression **1+2**.

CALCULATE COLUMN

Creates a new report column, performs on-the-fly calculations, and displays the calculation results in the newly-created column. Each new calculated column is appended to the right of the existing columns in the order in which it is created, and is given the next available column number. See [ORDER](#) for more information on column numbering and ordering.

Syntax

```
{ CALCULATE COLUMN "newColumn" = expression }
```