Best Practices for Implementing Essbase as a Data Source for Oracle Business Intelligence Enterprise Edition versions 10.1.3.3.2 and above

November 2008

1 Background

Oracle Business Intelligence Suite Enterprise Edition is an enterprise Business Intelligence platform with the ability to source from multiple heterogeneous data sources to enable pervasive business analytics. Oracle Business Intelligence Enterprise Edition releases (10.1.3.3.2 and higher) support Oracle Essbase as a physical data source. Essbase participates in the Common Enterprise Information Model (which allows integration with the other data sources in the enterprise), to build BI content that can be deployed to a broad user audience through Oracle BI Dashboards, Answers, and Delivers.

The purpose of this document is to highlight the features and limitations of the Oracle BI Server connectivity to Essbase, and to provide modeling techniques that utilize Essbase as a source within Oracle Business Intelligence.

2 Oracle Essbase Versions Supported with Oracle Business Intelligence

Oracle Business Intelligence supports Essbase version 7.1.6 and higher. Specific certified versions are documented in *System Requirements and Supported Platforms for Oracle BI Suite Enterprise Edition*.

3 Enabling the use of Oracle Essbase as a Data Source for the Oracle BI Server

3.1 Essbase Client Libraries

Oracle BI Server connectivity to Essbase is through the Essbase client libraries. The client libraries must be installed on the Oracle BI Server. Please see the most current *System Requirements and Supported Platforms for Oracle Business Intelligence Suite Enterprise Edition* document for the supported versions of the Essbase Client and corresponding Essbase Server versions for connectivity to the Oracle BI Server.

On Windows installations, the client installer typically configures environment variables correctly. To avoid possible issues, make sure that the PATH, ARBORPATH, and HYPERION_HOME variables are set correctly.

Note that there is no relationship between the Essbase Server being 32-bit or 64-bit and whether the client is 32-bit or 64-bit, e.g. for example a 32-bit client can access a 64-bit server.

3.2 BI Server Client Configuration for UNIX and Linux Systems

Perform the following configuration steps to access the Essbase client libraries through the BI server on UNIX and Linux Platforms:

Solaris: Oracle BI Server 64 bit mode:

1. Define ARBORPATH = <Essbase Client installation folder>

   For example:

```
ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
export  ARBORPATH
```

2. Add the Essbase Client Libraries folder to LD_LIBRARY_PATH:
   LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:<Essbase Client Libraries folder>
   For example:

   ```
   LD_LIBRARY_PATH_64=$LD_LIBRARY_PATH_64:$ARBORPATH/bin
   export LD_LIBRARY_PATH_64
   ```

Solaris: Oracle BI Server 32 bit mode

1. Define ARBORPATH = <Essbase Client installation folder>

   For example:

   ```
   ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to LD_LIBRARY_PATH:
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:<Essbase Client Libraries folder>

   For example:

   ```
   LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$ARBORPATH/bin
   export LD_LIBRARY_PATH
   ```

HP-UX PARISC: Oracle BI Server 32 bit mode

1. Define ARBORPATH = <Essbase Client installation folder>

   For example:

   ```
   ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to SHLIB_PATH:
   SHLIB_PATH=$SHLIB_PATH:<Essbase Client Libraries folder>

   For example:
   ```
   SHLIB_PATH=$SHLIB_PATH:$ARBORPATH/bin
   export SHLIB_PATH
   ```

HP-UX Itanium: Oracle BI 64 bit mode

1. Define ARBORPATH = <Essbase Client installation folder>

   For example:

   ```
   ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

2. Add the Essbase Client Libraries folder to SHLIB_PATH:
   SHLIB_PATH=$SHLIB_PATH:<Essbase Client Libraries folder>

   For example:
   ```
   SHLIB_PATH=$SHLIB_PATH:$ARBORPATH/bin
   export SHLIB_PATH
   ```

3. Define ESSLANG and LANG

   For example:

   ```
   ESSLANG=English_UnitedStates.UTF-8@Binary
   export ESSLANG

   LANG=en_US.utf8
   export LANG
   ```

4. Comment out the following three lines from the NQSConfig.ini file:

   ```
   [ GENERAL ]
   // Localization/Internationalization parameters.
   LOCALE="English-usa";
   SORT_ORDER_LOCALE="English-usa";
   SORT_TYPE="binary";
   ```

AIX: Oracle BI Server 32 and 64 bit mode

1. Define ARBORPATH = <Essbase Client installation folder>

For example:

```
ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
export ARBORPATH
```

2. Add the Essbase Client Libraries folder to LIBPATH:
   LIBPATH = $LIBPATH:<Essbase Client Libraries folder>

   For example:

   ```
   LIBPATH = LIBPATH:$ARBORPATH/bin
   export LIBPATH
   ```

Linux: Oracle BI Server 32 bit mode

1. Define ARBORPATH = <Essbase Client installation folder>

   For example:

   ```
   ARBORPATH=/export/home/Hyperion/AnalyticServicesClient
   export ARBORPATH
   ```

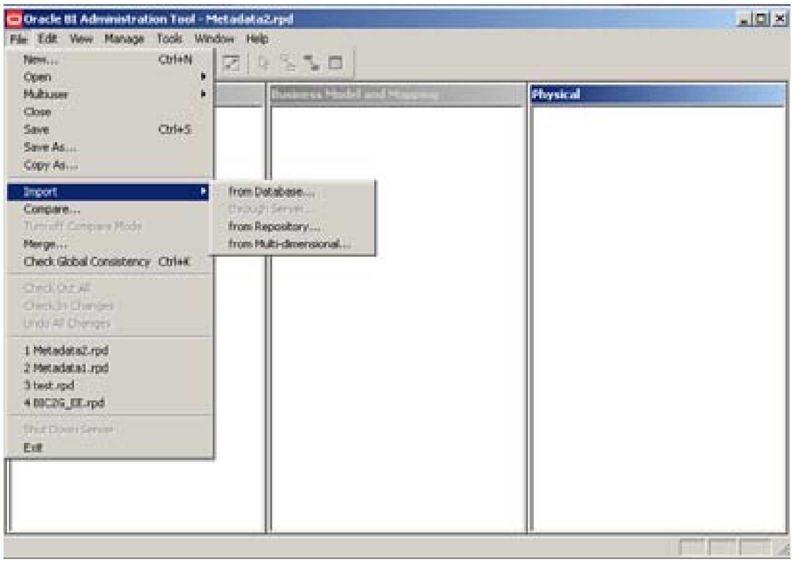
2. Add the Essbase Client Libraries folder to LD_LIBRARY_PATH:
   LD_LIBRARY_PATH=$ LD_LIBRARY_PATH:<Essbase Client Libraries folder>

   For example:

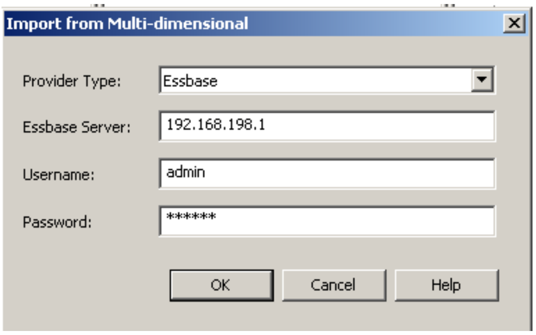
   ```
   LD_LIBRARY_PATH=$ LD_LIBRARY_PATH:$ARBORPATH/bin
   export LD_LIBRARY_PATH
   ```

4 Creating BI Server Metadata

The "Import from Multi-Dimensional" menu option in the Oracle BI Administration tool is used to create BI Server Physical layer metadata necessary to access Essbase cubes for reporting and analysis. The following illustrations outline the process of creating Oracle BI Server metadata for Essbase.



In the "Import from Multi-Dimensional" dialog, select Essbase as the provider type. Input the server and authentication information. For the import process, you will need to specify an Essbase Admin user. This ensures that substitution variables available on the Essbase Server are imported. After import, the connection pool to the Essbase Server can be re-configured for use only for the initialization block that is used to retrieve substitution variables and their values.   Additional connection pools can be added for end-user access.


Import from Multi-dimensional

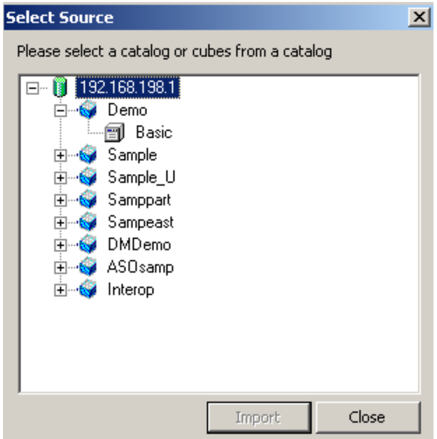
Provider Type: Essbase

Essbase Server: 192.168.198.1

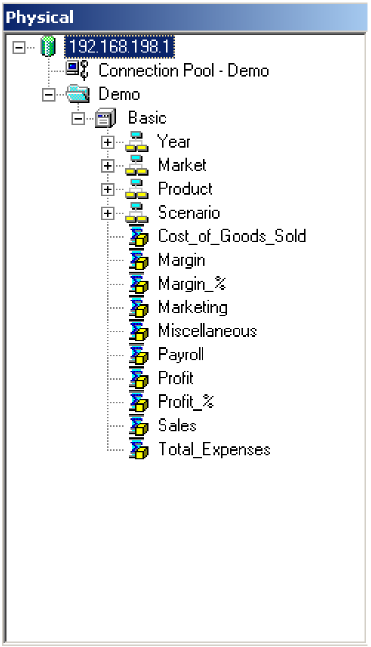
Username: admin

Password: ******

After selecting "OK", a list of applications and cubes available on the Essbase Server is displayed.


Please select a catalog or cubes from a catalog

The Administrator then has the option of selecting either Essbase applications or cubes to import. The example below shows how this is displayed in the physical layer. Note that Essbase applications are mapped as Physical catalogs.



After import you will notice that the physical cube consists of a collection of dimension objects and measure objects. Essbase generations will be mapped to BI Server levels. Essbase measure dimension members will be imported as a flat list of BI Server measure columns. No other member data will be imported into the BI Server metadata.

Once the physical cubes are imported, you can drag and drop them into the Business Model and Mapping Layer to automatically create a Business Model that includes associated dimensions. The Business Model can then be customized to take advantage of BI Server capabilities such as aggregate and fragment navigation.

4.1 Subsequent Changes to the Essbase Outline

Essbase outline changes are generally of two types:

- New dimension members
  New dimension members (other than measure members) are transparent to the BI Server, as member data is not imported into the BI Server metadata.
- Cube structure changes
  Cube structure changes (that is, adding or deleting dimensions, and levels) require either a re-import of the cube, or manual modification to the BI Server physical metadata objects to reflect changes.

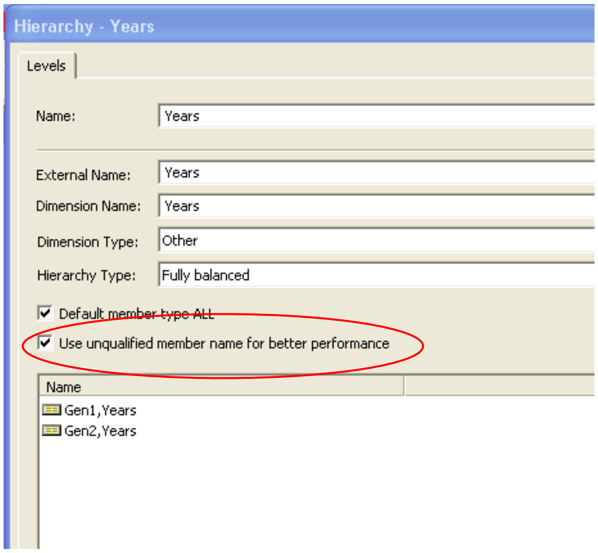
If new dimension members are added to the Essbase outline while cache is enabled, cache will need to be cleared to ensure new members are accessed on subsequent queries.

5 Adjusting Physical Layer Metadata Properties

The following topics should be considered when modifying physical layer metadata properties:

5.1 Unqualified Member Names

After importing the cube object into the Physical layer, the “Use unqualified member name for better performance” check box is selected by default. This check box must be cleared if the hierarchy contains duplicate or shared member names.



Name: Years

External Name: Years

Dimension Name: Years

Dimension Type: Other

Hierarchy Type: Fully balanced

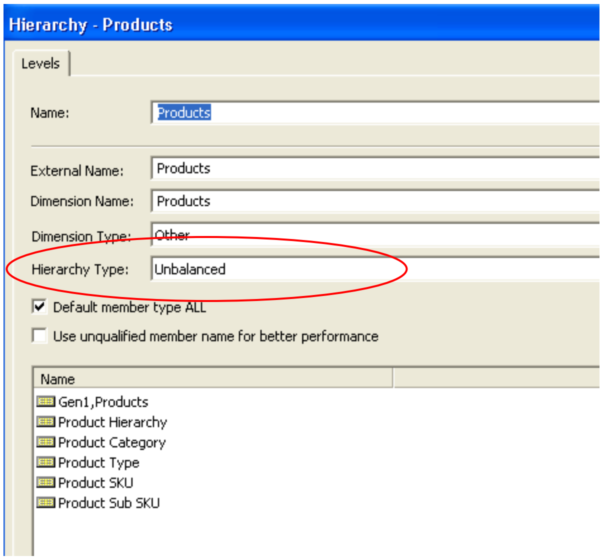
☑ Default member type ALL

☑ Use unqualified member name for better performance

Name
Gen1,Years
Gen2,Years

5.2 Unbalanced Hierarchies

All hierarchies are imported with a Hierarchy Type value of “Fully Balanced”. If a hierarchy is unbalanced you must change the Hierarchy Type value to “Unbalanced”.



Hierarchy - Products

Levels

Name: Products

External Name: Products

Dimension Name: Products

Dimension Type: Other

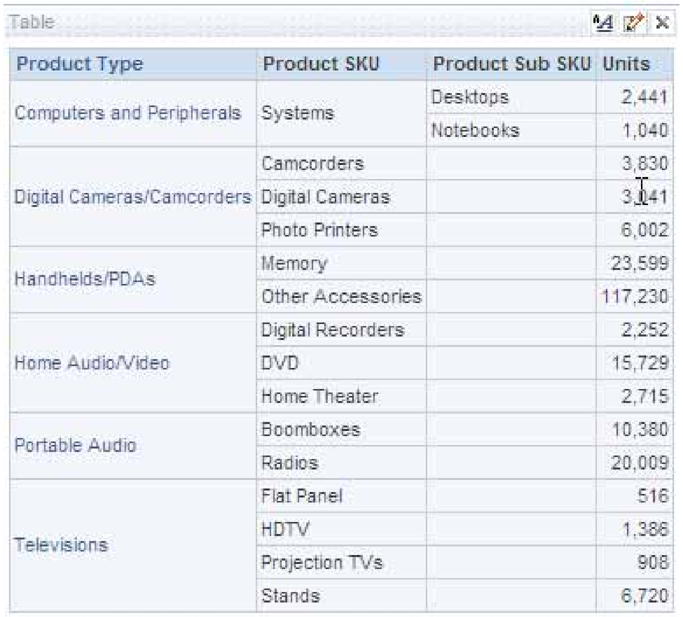
Hierarchy Type: Unbalanced

☑ Default member type ALL

☐ Use unqualified member name for better performance

Name
- Gen1,Products
- Product Hierarchy
- Product Category
- Product Type
- Product SKU
- Product Sub SKU

How an unbalanced hierarchy behaves in a report is illustrated below:



Table

Product Type	Product SKU	Product Sub SKU	Units
Computers and Peripherals	Systems	Desktops	2,441
		Notebooks	1,040
Digital Cameras/Camcorders	Camcorders		3,830
	Digital Cameras		3,041
	Photo Printers		6,002
Handhelds/PDAs	Memory		23,599
	Other Accessories		117,230
Home Audio/Video	Digital Recorders		2,252
	DVD		15,729
	Home Theater		2,715
Portable Audio	Boomboxes		10,380
	Radios		20,009
Televisions	Flat Panel		516
	HDTV		1,386
	Projection TVs		908
	Stands		6,720

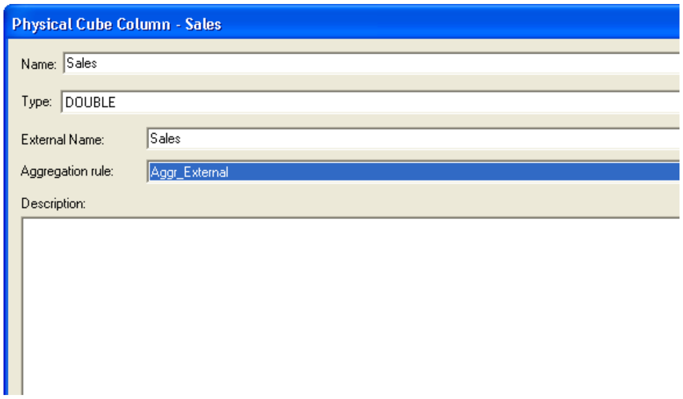
Notice that measure data is reported for the leaf member.

Unbalanced hierarchy support in Oracle Business Intelligence (release 10.1.3.3.2 and higher) is for Essbase only, and does not extend to other multi-dimensional or relational sources.

As indicated earlier Essbase generations are mapped to Oracle Business Intelligence levels upon import. Essbase levels (which are numbered starting from the leaf levels), are not imported. This restricts the ability to perform reporting on members at explicit Essbase levels. For example, in the case of an unbalanced hierarchy, members at level 0 in an Essbase outline will have a varying generation number. Since only Essbase generation information is mapped in the Oracle Business intelligence metadata (that is, mapped to an Oracle Business Intelligence level), explicit reporting on level 0 members is not possible.

5.3 Aggregation Type

For physical layer measure columns, the aggregation rule is set to Aggr_External by default. This setting allows Essbase to optimally access aggregate values. However, there are cases where an explicit aggregation rule is necessary. These cases will be described later in this document.



Physical Cube Column - Sales

Name: Sales

Type: DOUBLE

External Name: Sales

Aggregation rule: Aggr_External

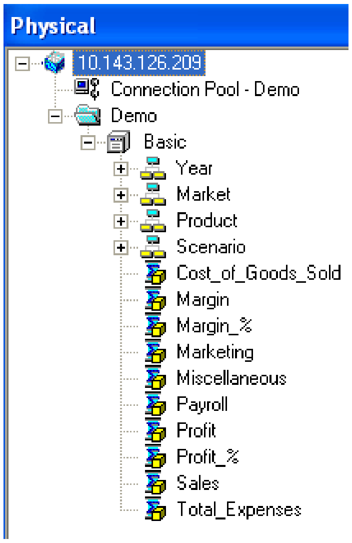
Description:

5.4 Measure Hierarchies

Essbase supports the concept of measure hierarchies. Measure hierarchies allow end users to drill from a measure to components that make up the measure (for example, you could drill from profit to revenue and costs). Oracle Business Intelligence does not support measure hierarchies. Measure members are imported as a flat list of columns that belong to the cube itself.

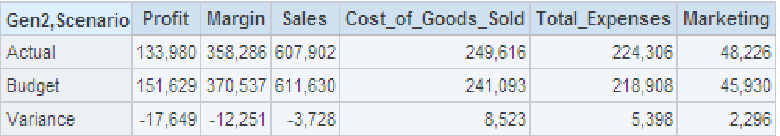
There are scenarios where it may make sense to designate an alternate hierarchy as the Oracle Business Intelligence measure hierarchy. This is typically the case with the Accounts dimension, which by default is treated as a measure hierarchy. For this case, an alternate dimension (for example, Scenario), can be used as the measures hierarchy in the BI Server metadata. The following example illustrates the process of swapping the Accounts dimension for the Scenario dimension as the measures hierarchy:

Default behavior:



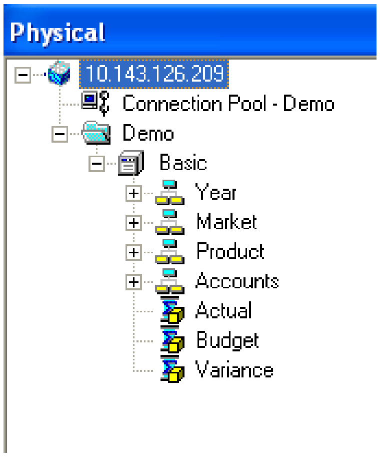
Physical

- 10.143.126.209
  - Connection Pool - Demo
  - Demo
    - Basic
      - Year
      - Market
      - Product
      - Scenario
      - Cost_of_Goods_Sold
      - Margin
      - Margin_%
      - Marketing
      - Miscellaneous
      - Payroll
      - Profit
      - Profit_%
      - Sales
      - Total_Expenses



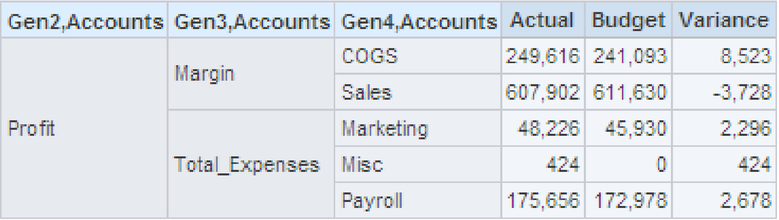
Gen2,Scenario	Profit	Margin	Sales	Cost_of_Goods_Sold	Total_Expenses	Marketing
Actual	133,980	358,286	607,902	249,616	224,306	48,226
Budget	151,629	370,537	611,630	241,093	218,908	45,930
Variance	-17,649	-12,251	-3,728	8,523	5,398	2,296

Desired behavior:



Physical

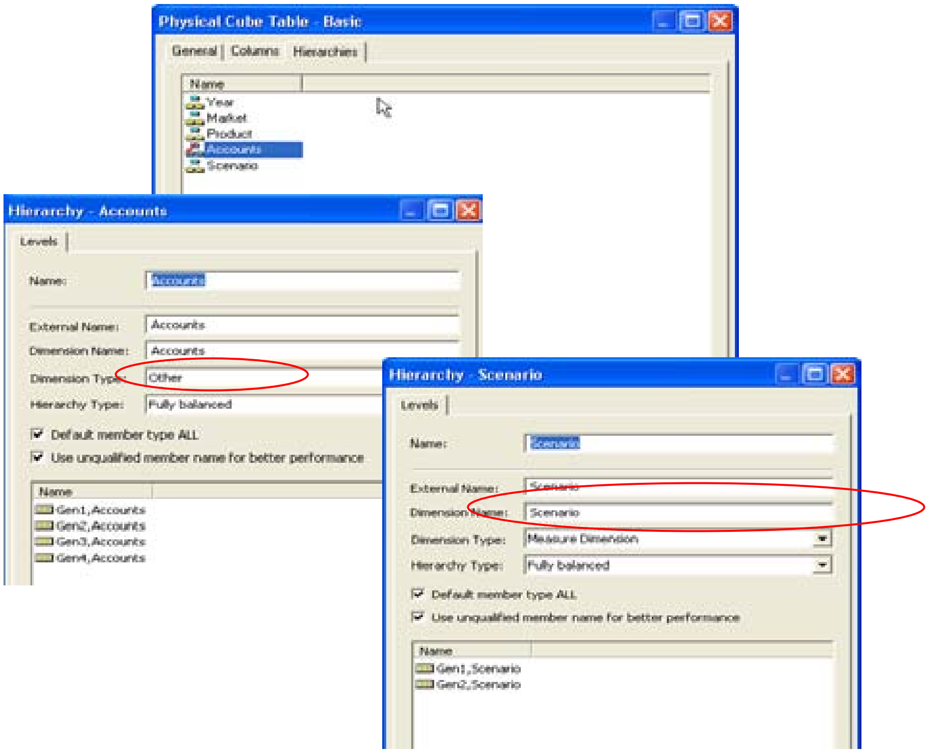
10.143.126.209
- Connection Pool - Demo
- Demo
  - Basic
    - Year
    - Market
    - Product
    - Accounts
    - Actual
    - Budget
    - Variance



Gen2,Accounts	Gen3,Accounts	Gen4,Accounts	Actual	Budget	Variance
Profit	Margin	COGS	249,616	241,093	8,523
		Sales	607,902	611,630	-3,728
	Total_Expenses	Marketing	48,226	45,930	2,296
		Misc	424	0	424
		Payroll	175,656	172,978	2,678

The desired behavior is achieved with the following steps:

1. Using the “Demo.Basic” cube as an example, after import, select the cube object and select properties.
2. On the Hierarchies tab:
    a. Edit the properties of the Accounts dimension by changing the Dimension type to “Other”
    b. Edit the properties of the Scenario dimension by changing the Dimension type to “Measure Dimension”
3. After changing the cube properties, you will then need to delete the existing measures from the BI Server metadata physical cube and add the members from the Scenario dimension (Actual, Budget, Variance)



Year
Market
Product
Accounts
Scenario

Name: Accounts
External Name: Accounts
Dimension Name: Accounts
Dimension Type: Other
Hierarchy Type: Fully balanced
☑ Default member type ALL
☑ Use unqualified member name for better performance

Gen1,Accounts
Gen2,Accounts
Gen3,Accounts
Gen4,Accounts

Name: Scenario
External Name: Scenario
Dimension Name: Scenario
Dimension Type: Measure Dimension
Hierarchy Type: Fully balanced
☑ Default member type ALL
☑ Use unqualified member name for better performance

Gen1,Scenario
Gen2,Scenario

6 Other Modeling Techniques

6.1 Federation with other data sources

Oracle Business Intelligence supports joining of Essbase data with other data sources in a single report. Federation requires that Essbase dimensions conform across the disparate sources. A common use case where dimensions may be non-conforming is analysis along the Accounts dimension. In this case, Account members at a given logical level may exist in Essbase in various generations. Since the BI Server assigns generations to explicit levels, the Server cannot determine the logical level information for a given Account member. Federation cannot be modeled for this scenario.

Example federation use cases are described below:

1) Disparate facts: Forecast data is available for Region level data in Essbase. Actual data is available for Regions in relational.
2) Disparate fact sources: Unit Sales are available at aggregated levels in Essbase (State and above). Unit Sales are available at a detailed level (City) in relational. There is a common dimension level (State) across Essbase and relational at which a join can be made.

In general, federation with Essbase can be achieved when all of the following conditions are met:

1) There is a conforming dimension(s) between Essbase and the disparate source.
2) For the conforming dimension, there is a common level with common data at which both sources can be joined
3) For the Essbase dimension, members at the same logical level exist in one and only one generation.
4) The BI Server assumes that measure data can be retrieved at or above the level specified in the logical fact source for the Essbase source.

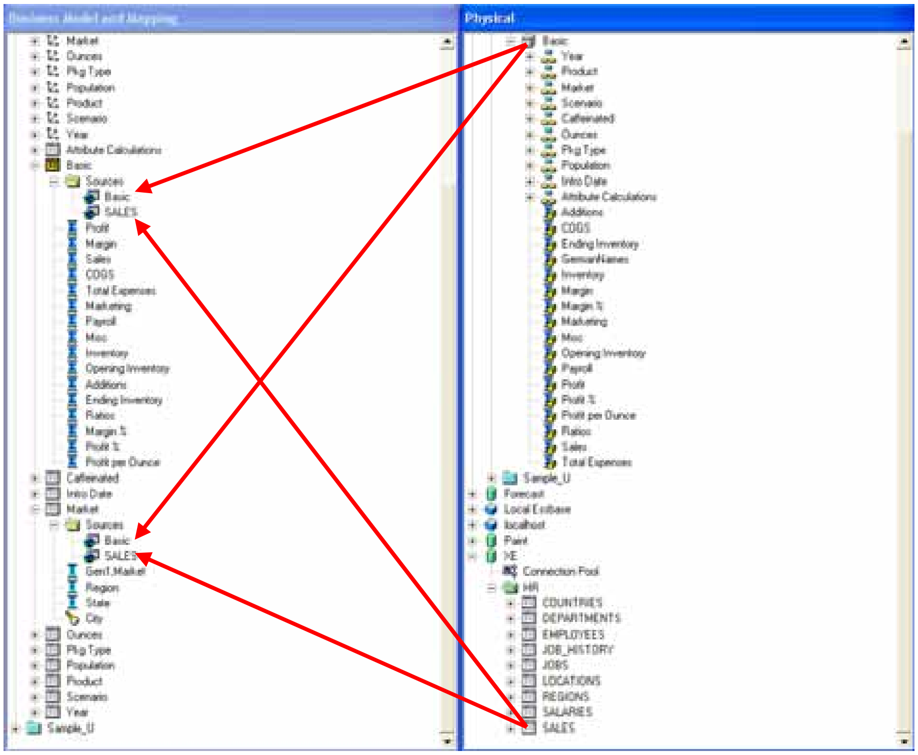
When modeling federation:

1) Ensure the level is set appropriately for both the dimension and fact table sources.
2) Set the aggregation rule for the logical fact column to an explicit rule.

The following illustration describes modeling federation for a drill-through use case:

I can view Total Sales (sourced from Essbase) for States. I now want to see the transaction details for each City within a State.

By modeling the repository such that the “Sales” logical column is sourced from Essbase for State and above (logical table source for aggregated data) while the detailed transactions are mapped to an Oracle DB (LTS for detailed transactional data), the BI Server will seamlessly navigate from an Essbase cube to the Oracle RDB for transaction level details when a user drills down from the Total Sales for State to the City level.



Note that when federating on the Time dimension, multiple sources for time series calculations are not supported.

6.2 Time Series Functions

Oracle BI Server Time Series functions, Ago and ToDate, are supported for use with Essbase. The BI Server function-ships the appropriate time series functions to Essbase.

6.3 Evaluate Function for MDX Sources

In Oracle Business Intelligence Enterprise Edition (release 10.1.3.3.2 and higher) you can use Evaluate expressions to leverage functions for Essbase. Note that different versions of Essbase have variations in the native functions supported. An Evaluate expression that works against one version of Essbase may not necessarily work in another version of Essbase.

Support for Evaluate does not extend across all multi-dimensional sources at this time.

The following use case examples use the EVALUATE_AGGREGATE and EVALUATE functions. Note that expressions are applied to columns in the Logical Table Source that refers to the physical cube:

EVALUATE_AGGREGATE is used to implement custom aggregations. For example, the user may want to compare overall regional profit to profits for the top 3 products in the region. A new measure can be defined to represent the profits for top 3 products resulting in the logical SQL statement:

> *Select Region, Profit, EVALUATE_AGGREGATE('SUM(TopCount(%1.members, 3, %2), %3)', Products, Profit, Profit) Top_3_prod_Profit From SampleBasic*

The Oracle BI Server will generate the following expression for the custom aggregation:

> *member [Measures].[MS1] as*
> *'SUM(Topcount([Product].Generations(6).members,3,[Measures].[Profit]),[Measures].[Profit])'*

Similarly, the EVALUATE function can be used implement scalar functions that are computed post-aggregation. EVALUATE will change the grain of the query, if its definition makes explicit references to dimensions (or attributes) that are not in query.

Let's say the user would like to see the Profits for the top 5 products ranked by Sales sold in a Region, after creating the applicable measure, the resulting Logical SQL statement is as follows:

> *Select Region, EVALUATE('TopCount(%1.members, 5, %2)' as VARCHAR(20), Products, Sales), Profits*
> *From SampleBasic*

The Oracle BI Server will generate the following expression to retrieve the top 5 products:

> *set [Evaluate0] as '{Topcount([Product].Generations(6).members,5,[Measures].[Sales]) }'*

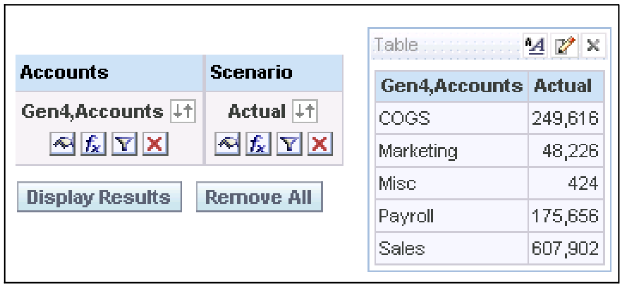
6.4 Outline Sort Order

Essbase users may want to see dimension members returned in the order they are physically stored in the outline.

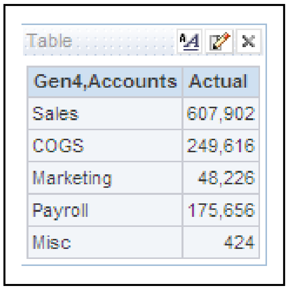
There are two options that leverage the EVALUATE function to enable this view:
1. In the metadata repository, create a Sort Order column (Recommended)
2. In Answers, create a column solely used for sorting

By default, Answers returns dimension members in alphabetical (or numeric) order
The following example compares default behavior with desired behavior.

Default **Desired**



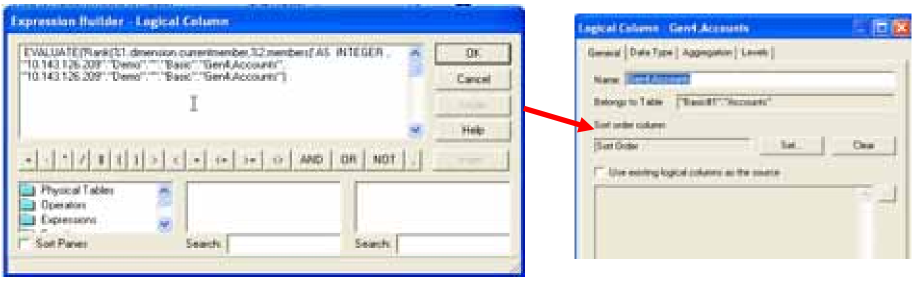
Gen4,Accounts	Actual
COGS	249,616
Marketing	48,226
Misc	424
Payroll	175,656
Sales	607,902



Gen4,Accounts	Actual
Sales	607,902
COGS	249,616
Marketing	48,226
Payroll	175,656
Misc	424

In order to achieve the desired results for this example::

1. Define a new column in the metadata repository using an Evaluate expression (as shown in the illustration below).
2. Use this new column as the "Sort Order" for the dimension where sort order needs to be preserved.



```
EVALUATE('Rank(%1.dimension.currentmember,%2.members)' AS  INTEGER ,
"10.143.126.209"."Demo".""."Basic"."Gen4,Accounts",
"10.143.126.209"."Demo".""."Basic"."Gen4,Accounts")
```

Name: Gen4,Accounts

Belongs to Table: "Basic#1"."Accounts"

Sort order column: Sort Order

□ Use existing logical columns as the source

Note that the RANK function is only supported in Essbase 9.3 and higher.

6.5 User-Defined Attributes

Essbase supports the concept of user-defined attributes (UDAs). A UDA is essentially any arbitrary textual string that can be associated with any member from a dimension. A member can have multiple strings associated. The import process to build Oracle BI Server metadata does not retrieve UDA's.

Users can still query using UDAs by leveraging the EVALUATE function.

Consider the following example where "Major Market" is a UDA.

After creating the applicable column, the resulting logical SQL statement is as follows:

> *Select EVALUATE('FILTER(%1.Members, isUDA([%2.Dimension, "Major Market"))', State, State), Sales*
> *From SampleBasic*

The Oracle BI Server will generate the following expression for the custom column:

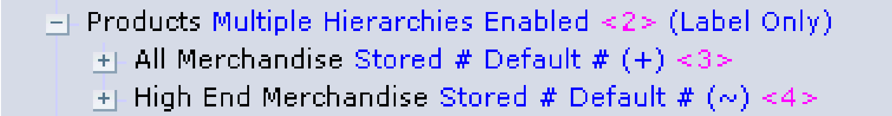
```
set [Evaluate0]  as
'{FILTER([Customer].Generations(3).members,isUDA([Customer].Generations(3).Dimension,"Major
Market")) }'
```

6.6 Alternate Hierarchies

Essbase supports the concept of alternate hierarchies. Essbase alternate hierarchies are not automatically created when dragging and dropping a cube from the repository Phyiscal layer to create a Business Model. Essbase's alternate hierarchy can be modeled manually using the Oracle Business Intelligence multiple hierarchy support.

The recommended best practice is to model multiple hierarchies within Oracle Business Intelligence metadata to avoid potential double-counting issues.

Consider the following example using the ASOSample.Sample cube:

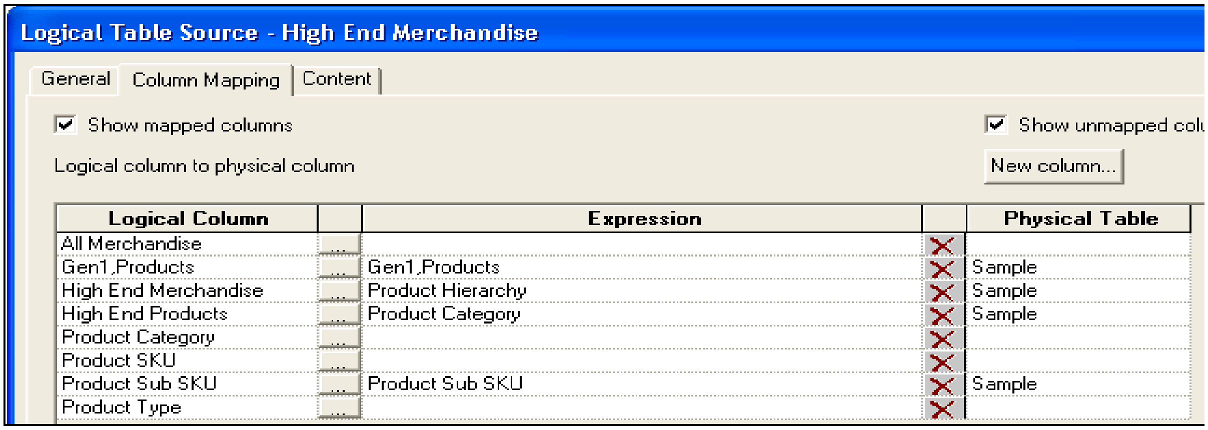


⊟ Products Multiple Hierarchies Enabled <2> (Label Only)
⊞ All Merchandise Stored # Default # (+) <3>
⊞ High End Merchandise Stored # Default # (~) <4>

In this cube, the All Merchandise hierarchy is as follows:
Product Category -> Product Type -> Product SKU -> Product Sub SKU

The High End Merchandise hierarchy contains members from Product Category

To add the alternate hierarchy, add a new logical table source to the Products folder and two new columns (High End Merchandise, High End Products) that correspond to the new hierarchy.



Logical Table Source - High End Merchandise

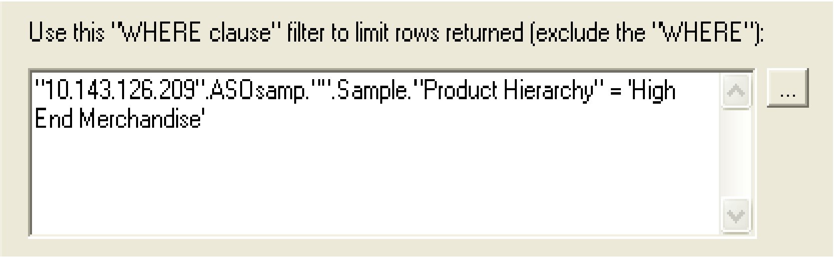
☑ Show mapped columns

☑ Show unmapped colu

Logical column to physical column

Logical Column		Expression		Physical Table
All Merchandise	...		✕	
Gen1,Products	...	Gen1,Products	✕	Sample
High End Merchandise	...	Product Hierarchy	✕	Sample
High End Products	...	Product Category	✕	Sample
Product Category	...		✕	
Product SKU	...		✕	
Product Sub SKU	...	Product Sub SKU	✕	Sample
Product Type	...		✕	

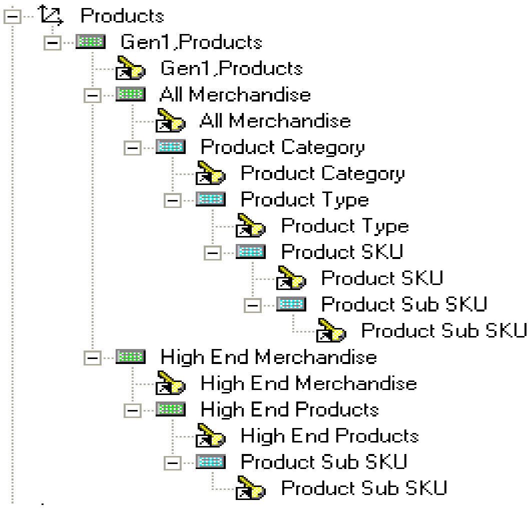
Next, in the Content tab of the new Logical Table Source, specify a filter on the “Product Hierarchy” column as illustrated below:



Use this "WHERE clause" filter to limit rows returned (exclude the "WHERE"):

```
"10.143.126.209".ASOsamp."".Sample."Product Hierarchy" = 'High End Merchandise'
```

Next, add the alternate hierarchy to the Products Dimension as illustrated below:



- Products
  - Gen1,Products
    - Gen1,Products
    - All Merchandise
      - All Merchandise
      - Product Category
        - Product Category
        - Product Type
          - Product Type
          - Product SKU
            - Product SKU
            - Product Sub SKU
              - Product Sub SKU
    - High End Merchandise
      - High End Merchandise
      - High End Products
        - High End Products
        - Product Sub SKU
          - Product Sub SKU

One limitation to this modeling approach is that all multiple hierarchies in the Oracle Business Intelligence metadata must terminate at a common leaf member.

6.7 Substitution Variables

Essbase substitution variables are automatically retrieved and populated into corresponding BI Server dynamic system variables.

- Depending on the scope of the Essbase variable, the naming convention for the BI Server variable is as described below.

  Server instance scope
  <server name>:<var name>

  Application scope
  <server name>:<app name>:<var name>

  Cube scope

<server name>:<app name>:<cube name>:<var name>

- The refresh interval should be set appropriately to reflect anticipated update cycles for Essbase variables.

6.8 Attribute Dimensions

Essbase attribute dimensions are available upon import. However, member attributes are not automatically associated with corresponding dimensions and levels during the import process. To manually create the association, the Oracle Business Intelligence Administrator may choose to:

1. Map the member attribute to the appropriate logical table in the Business Model Layer or
2. Include the attribute in the appropriate presentation table in the Presentation Layer

6.9 Connecting to Multiple Nodes in an Essbase Server Cluster

It may be necessary to connect BI Server(s) to an Essbase Server cluster. Using BI Server session variables it is possible to load balance Essbase server nodes using a round robin approach.

General Instructions:

1. Create a database-stored package with a function in any of the Oracle Business Intelligence supported relational databases, which returns an Essbase node from a List of available / active Essbase nodes in a round-robin fashion.
2. An external polling mechanism is required to constantly test the availability of the Essbase Nodes participating in the Clustered environment and should update the availability on List of available/active Essbase nodes table maintained on the relational schema and used by the stored function.
3. Using the Oracle Business Intelligence Administration Tool, create a session based initialization block to make a call to this stored function and a session variable that this initialization block will populate.
4. Use this session variable as the Essbase Server node in the Connection Pool object created for the physical database object mapped to Essbase database.

Detailed Instructions are provided in Appendix A.

7 Other Considerations and Limitations

7.1 Shared Members and Double Counting in Sub-Totals

When reporting on shared members, double-counting may occur with sub-totals in the following scenarios:

1. Aggregation function other than ‘External_Aggregate’ (example – SUM, AVG, COUNT etc.). An explicit aggregation rule may under some circumstances result in the BI Server processing the aggregation. When the BI Server does the internal execution, the BI Server may not have the member level metadata to suppress shared members.
2. Filter below query grain – When you have a filter below the aggregate grain, the BI Server may double counts shared members for all types of aggregation functions including ‘External_Aggregate’.

7.2 Explicit Aggregation Rules for Essbase Measure Columns

As mentioned earlier there are cases where explicit aggregation rules need to be set for an Essbase measure column. This is generally driven by reporting requirements where the aggregation cannot be “pushed” down to Essbase. Examples of these cases include:

1. Use of logical column with expressions that cannot be pushed down to Essbase, e.g. CASE statements
2. Use of logical column expressions using EVALUATE to return a scalar list of members
3. Filter expressions other than equivalency

For these cases, performance will be degraded as aggregation is done on the BI Server rather than in Essbase.

Additionally, when the Aggregation rule is set to “Aggr_External” for Cube Measures in the Physical layer, the BI Server issues MDX Aggregate functions for these Time balance measures. Essbase started supporting the Aggregate function in its MDX language starting from release 9.3.x. For users connecting to an Essbase cube versions prior to 9.3.1, these types of queries will fail with the error message “Essbase Error: Unknown Member AGGREGATE used in query” and would have to set the Aggregation rule as Sum for these cube members in the Physical and Business Model layer.

7.3 Member Name versus Alias

When displaying members, the BI Server will return the default alias if available and the member name if a default alias is not available. We do not currently provide the capability for a user to dynamically determine whether to display member name or values from alias tables.

It is possible to add a workaround to implement a workaround to report on member name, if the default alias is not desired. This is only available for the lowest level in the dimension. To implement this:

1) Add a column to the lowest level of the desired dimension in the physical layer. You can give it any name.
2) In the “External Name” field, please type 'MEMBER_NAME'.
3) Set the data type to ‘VARCHAR’

7.4 Security

Login credentials used to authenticate for Oracle Business Intelligence can be propagated for authentication by Essbase using the :USER, :PASSWORD variables in the connection pool. Note that erroneous data or even query errors may occur when the connection pool user queries for data when a visibility restriction rule is applied to the data (#NoAccess). This issue is addressed in a patch for version 10.1.3.4

7.5 Combining Multiple Cubes into a Single Business Model

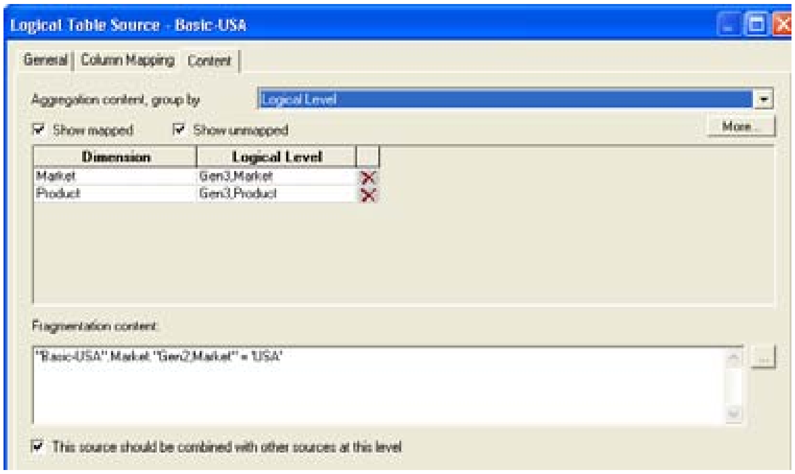
As a general best practice, when the desire is to combine multiple cubes into a single business model, we recommend using partitioning capabilities within the Essbase server. If this is not possible and the desire is to model this in the Oracle Business Intelligence metadata, there are several considerations to keep in mind.

Inconsistent query results may occur if a physical database or catalog folder of type Essbase is dragged from the Physical layer of the repository to the Business Model layer to create a single Business Model.

To work around this issue, you must take one of the following actions:

- If the Administrator wants to create a separate Business Model for each cube, then each cube should be dragged individually to the Business Model layer.
- If a single Business Model is desired for each physical database or physical catalog, the database or physical catalog can be dragged to the Business Model layer. However, by doing so, any dimensions with the same name will be treated as conforming dimensions and a single Logical dimension will be created for all such dimensions. This means the Logical dimension will be sourced from all such dimensions with the same name and hence may lead to potential query issues. It is the responsibility of the Administrator to correct the mappings or to create individual logical dimensions for dimensions that are not conforming.

When creating logical dimension sources, ensure that the fragmentation filter uses only a physical cube column and not a derivation, and that the option to combine sources is checked.



Aggregation content, group by: Logical Level

☑ Show mapped ☑ Show unmapped

Dimension	Logical Level	
Market	Gen3,Market	✗
Product	Gen3,Product	✗

Fragmentation content:

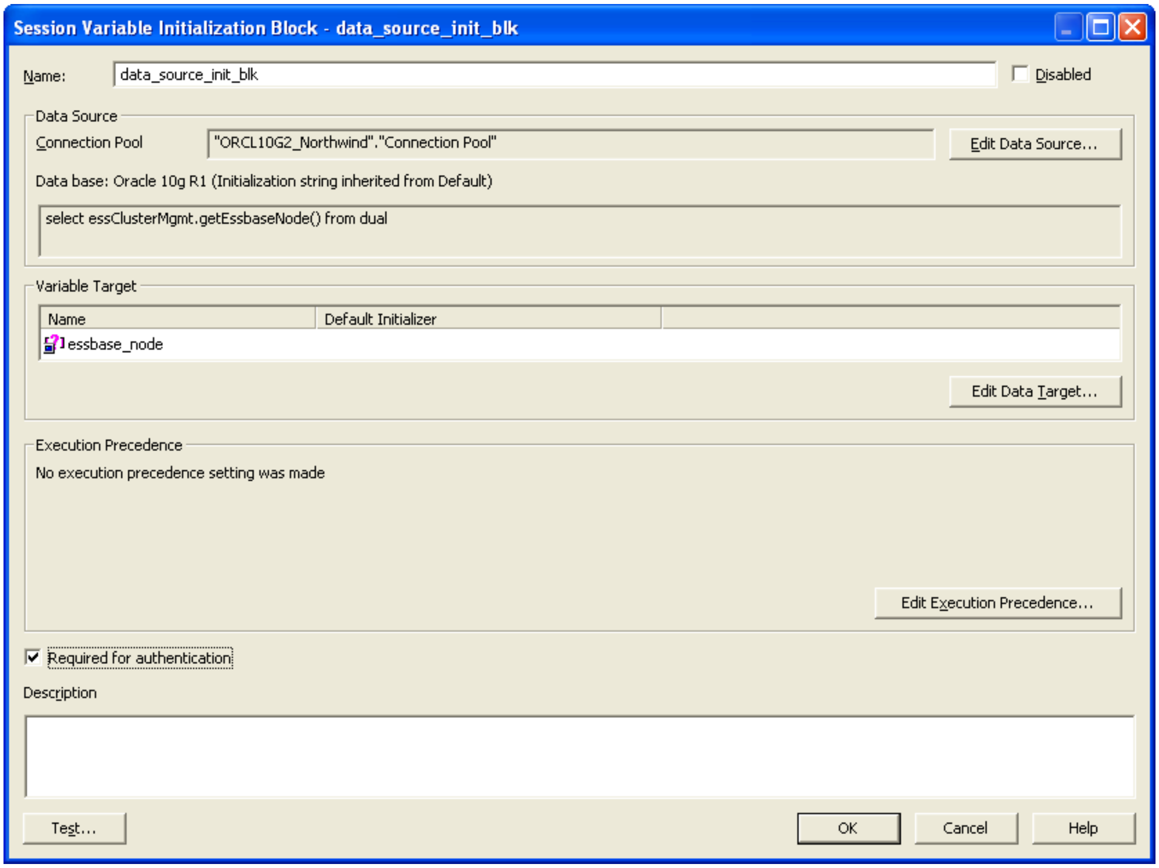
"Basic-USA".Market."Gen2,Market" = 'USA'

☑ This source should be combined with other sources at this level

Appendix A – Essbase Clustering

The following detailed instructions enable you to create a clustered Essbase environment in the BI Server metadata:

1. Create a Session-based Initialization Block as follows:
    a. Click Manage → Variables… → Session → Initialization Blocks → New Session Initialization Block…
    b. Provide a name (data_source_init_blk) for the initialization block
    c. Click → "Edit Data Source…" button
        i. Select "Data Source Type" → Database
        ii. For Default Initialization String, provide the following SQL:
            "select essClusterMgmt_Pkg.getEssbaseNode_fn() from dual"
        iii. Click → "Browse…" to select a Relational database based connection pool (which connects to a schema that contains the getEssbaseNode stored procedure)
        iv. Click → "Test…" to test connectivity and check if the store procedure returns desired results, that is Essbase Node.
        v. Click → OK to return to Initialization Block dialog
    d. Click → "Edit Data Target…" button to specify the Session variable
        i. Click → "New…" button to open Session Variable dialog
            1. Enter a name for the session variable (Essbase_node), which will be populated with Essbase Node value returned by the Initialization Block.
            2. Enter the following Description in the space provided:
                "This session variable will be populated with the next available Essbase node value for each User session by data_source_init_blk initialization block and all subsequent queries for the User session will be sent to this Essbase node for execution."
            3. Click → OK to re turn to Session Variable dialog
        ii. Click → OK to return to Initialization Block dialog
    e. Click "Required for Authentication" check box
    f. Click → "Test…" button to test connectivity and check that essbase_node session variable is now populated correctly with the Essbase Node value returned by the initialization block.
    g. Click → OK to close the dialog.



Session Variable Initialization Block - data_source_init_blk

Name: data_source_init_blk

☐ Disabled

Data Source

Connection Pool: "ORCL10G2_Northwind"."Connection Pool"

Data base: Oracle 10g R1 (Initialization string inherited from Default)

```
select essClusterMgmt.getEssbaseNode() from dual
```

Variable Target

Name	Default Initializer
essbase_node	

Execution Precedence

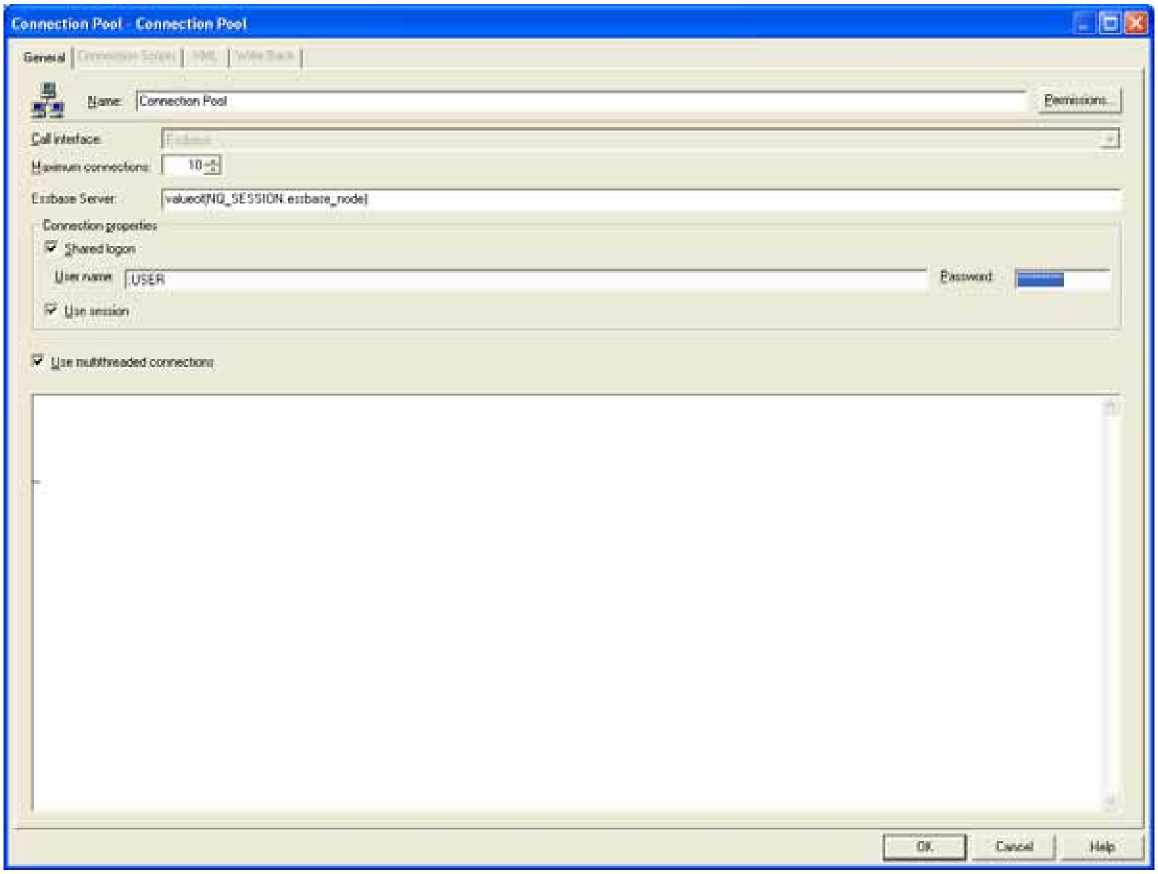
No execution precedence setting was made

Edit Execution Precedence...

☑ Required for authentication

Description

2. Use this Session variable in the Connection Pool object for Essbase data source.
    a. In the Physical Layer of Admintool, expand the physical database object that maps to the Essbase database
    b. Double Click → Connection Pool object to edit the properties of this connection
    c. Enter → "valueof(NQ_SESSION.Essbase_node)" in the Essbase Server Text box.
    d. Enter → ":USER" and ":PASSWORD" for the "User name" and "Password" in the Connection properties section.
    e. Leave the other settings as is.
    f. Click → OK to close the Connection Pool dialog.



Name: Connection Pool

Call interface: Essbase

Maximum connections: 10

Essbase Server: valueof(NQ_SESSION.essbase_node)

Connection properties

☑ Shared logon

User name: :USER Password:

☑ Use session

☑ Use multithreaded connections

Limitations:

- If an Essbase node goes down, the external polling mechanism should remove it from the relational table, so that stored function does not return this node and Oracle Business Intelligence doesn't use it any more until it is back on.
- If an Essbase node goes down, queries on active connections would fail and would require you to login again.

Sample Oracle Database Package for Cluster Management:

Table DDL:

create table essbase_clustered_nodes (node_num number, node_name varchar2(100), node_active varchar(1));

Package DDL:

```
/*
        Name:   essClusterMgmt_Pkg
        Author:  Deepak Bhatia
        Created: 21-APR-2008
        Purpose: This package defines a function which returns an Essbase node from list of
active nodes in a round-robin fashion.
*/

CREATE OR REPLACE PACKAGE essClusterMgmt_Pkg AS
g_last_returned_node number;
FUNCTION getEssbaseNode_fn return varchar2;

END essClusterMgmt_Pkg ;
/

CREATE OR REPLACE PACKAGE BODY essClusterMgmt_Pkg AS

FUNCTION getEssbaseNode_fn return varchar2
IS

l_essnode_name varchar2(50);
l_essnode_num number;
l_max_node_num number;

BEGIN

  begin

    select max(node_num) into l_max_node_num from essbase_clustered_nodes where
node_active='Y';

    if (nvl(g_last_returned_node,0) < l_max_node_num) then
                select min(node_num) into l_essnode_num from essbase_clustered_nodes
where node_num > nvl(g_last_returned_node,0) and node_active='Y';
    else
                select min(node_num) into l_essnode_num from essbase_clustered_nodes
where node_active='Y';
    end if;

    g_last_returned_node := l_essnode_num;
    SELECT node_name into l_essnode_name from essbase_clustered_nodes where
node_num = l_essnode_num ;

    return l_essnode_name ;

   Exception
    when OTHERS then
    begin
      raise_application_error (-20001, 'Error: essClusterMgmt_Pkg.getEssbaseNode_fn:: Error
encountered. Ora Error: '|| SQLERRM);
```

```
      end;


  end;

END getEssbaseNode_fn;

END essClusterMgmt_Pkg ;
/
```

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com