

RELEASE 9.3.1.1.0

ORACLE DATA INTEGRATOR
ADAPTER FOR HYPERION ESSBASE
GETTING STARTED

ORACLE | Hyperion

CONTENTS IN BRIEF

Sample Files for Practice	2
Essbase Sample Application Prerequisites	3
Setting Up an Environment	3
Loading and Extracting Metadata or Data Using Sample Interfaces	10
Creating Models	13
Creating Interfaces and Packages	20

Sample Files for Practice

This document guides you through the use of the sample files that are included with Oracle® Data Integrator Adapter for Oracle's Hyperion® Essbase® – System 9 . The sample files are intended to familiarize you with the adapter and provide practice in using it to load and extract metadata and data, and perform other Essbase specific tasks.

The sample files are delivered in the `odiaess_93110_samples.zip` file, which you must extract to the C drive.

Note:

Some Loading Knowledge Module (LKM) and Integration Knowledge Module (IKM) options within the sample repository are set to use the absolute path of `C:\Essbase_Samples`. If the `odiaess_93110_samples.zip` is unzipped to a different directory, then, any option that references `C:\Essbase_Samples` must be changed.

In the C drive the folder `Essabase_Samples` is created and it contains these folders and files:

- `Calc_Scripts`, which contains these files:
 - `calcall.csc`
 - `ExtData.csc`
- `Data`, which contains these files:
 - `loadMeasures.csv`
 - `loadMarkets.csv`
 - `loadProducts.csv`
 - `loadDataWithDataAsColumn.csv`
 - `loadDataWithMeasuresAsColumns.csv`
 - `extractDataWithDataAsColumn.csv`
 - `extractDataWithMeasuresAsColumns.csv`
 - `extractEssbaseMeasures.csv`
 - `Calcextract.csv`
- `Essbase_Rules`, which contains these files:
 - `ACCTPC.rul`
 - `MKTGEN.rul`
 - `MRKTPC.rul`
 - `PRODPC.rul`
- `MAXL`, which contains these files:
 - `postmxl.mxl`
 - `premaxl.mxl`

- MDX_Scripts, which contains the file—MDXExtract.mdx
- Report_scripts, which contains the file—extract.rep
- Work_Repository, which contains the file—Work_Repository.zip

Essbase Sample Application Prerequisites

Using the sample files requires that you have an Essbase application and database installed. This guide will use the application—Sample and database—Basic. If you do not have the Essbase application—Sample and database—Basic installed, you must create both. For instructions see *Chapter 6. Setting Up Sample Applications* in the *Hyperion Essbase – System 9 Installation Guide for Windows*.

The documentation for the Oracle® Data Integrator Adapter for Hyperion Essbase is based on the assumption that users have previous experience with Oracle Data Integrator and have full access to the documentation. The latest documentation for Oracle Data Integrator is available for download from the Documentation area of the Oracle Technical Network (OTN) Web site (<http://www.oracle.com/technology/index.html>).

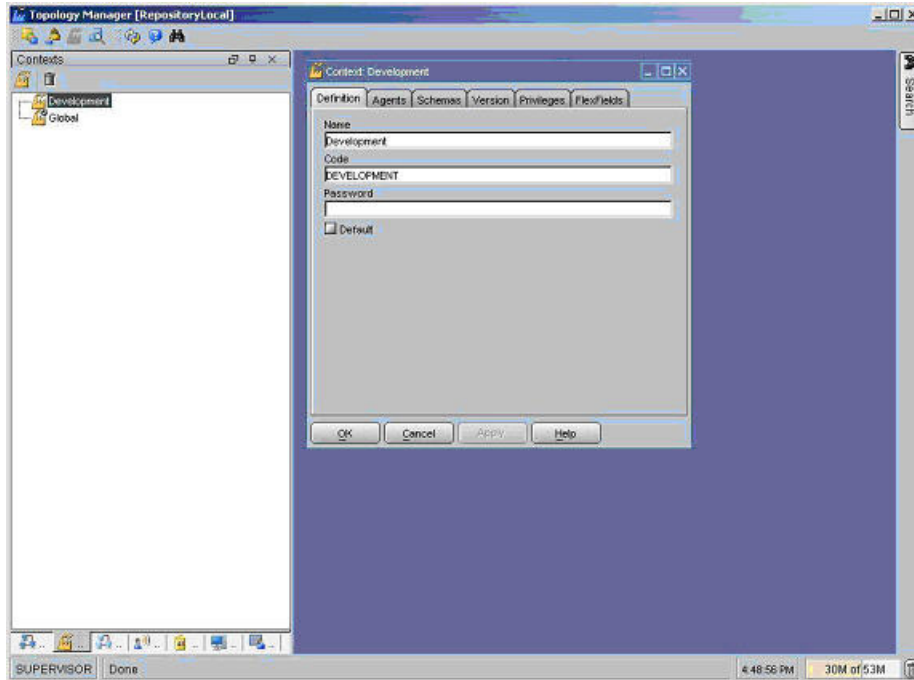
Setting Up an Environment

Before you load the metadata from the sample files, you must set up the sample environment in Oracle Data Integrator, as described in the following topics.

Creating the Context

Launch the Topology Manager, and create a context called Development, as shown in [Figure 1](#). See the *Oracle Data Integrator User's Guide* for instructions.

Figure 1 Creating a Context



Setting Up the Essbase Data Server

Use Oracle Data Integrator to create a data server for the Hyperion Essbase technology and create a physical schema and logical schema for the data server.

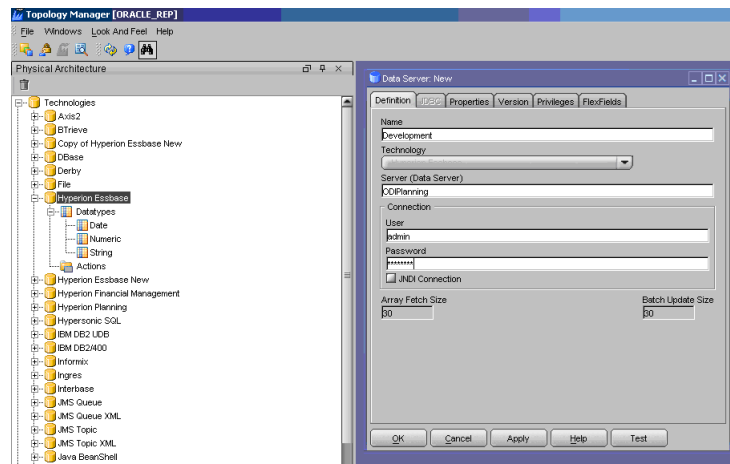
See the *Oracle Data Integrator User's Guide* for more information about creating data servers.

- To set up an Essbase data server:
- 1 Using the Topology Manager, create a data server under the Hyperion Essbase technology:

Note:

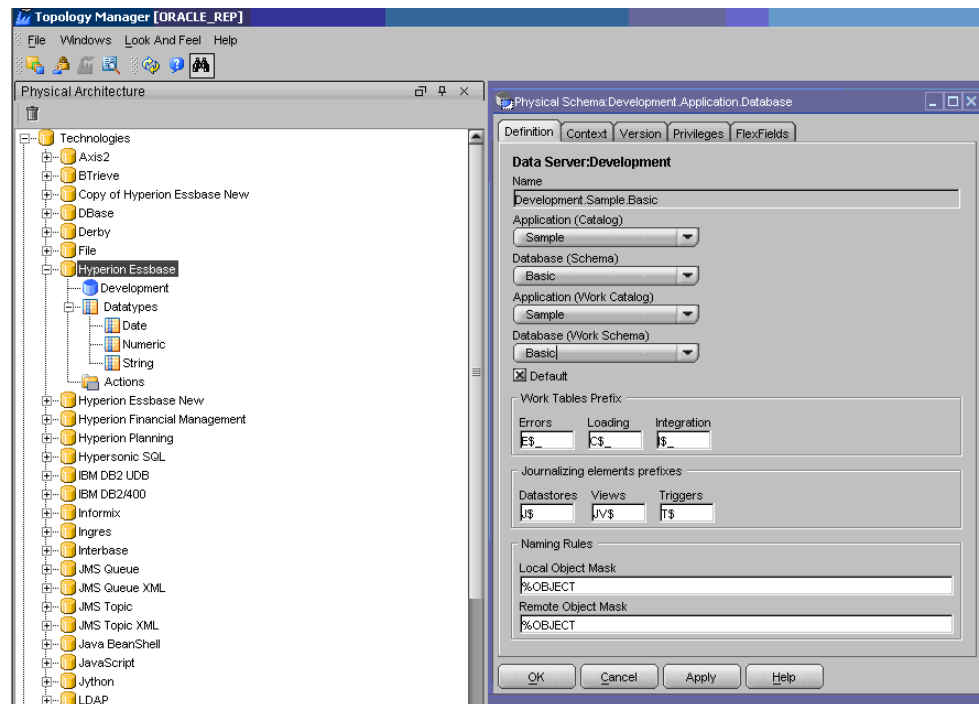
If the Hyperion Essbase technology is not listed in the Topology Manager, you must import it. See the *Oracle Data Integrator User's Guide* for instructions.

- 2 In **Name**, enter the name `Development` for the data server.



- 3 Provide information required to set up the data server and click **OK**.
- 4 Create the physical schema to point to the Essbase application.

In this figure, the physical schema points to the application—Sample and database—Basic:



- 5 On **Context**:
 - a. Set the Context to Development.
 - b. Enter the logical schema name SampleEssbase.

Caution!

If you give the logical schema a different name, update the models and interfaces to point to the name that you used. Otherwise, you might be unable to run the packages and interfaces after importing the work repository.

- c. Click OK.

See the *Oracle Data Integrator User's Guide* for more information about setting up a data server.

Setting Up a File Physical Schema

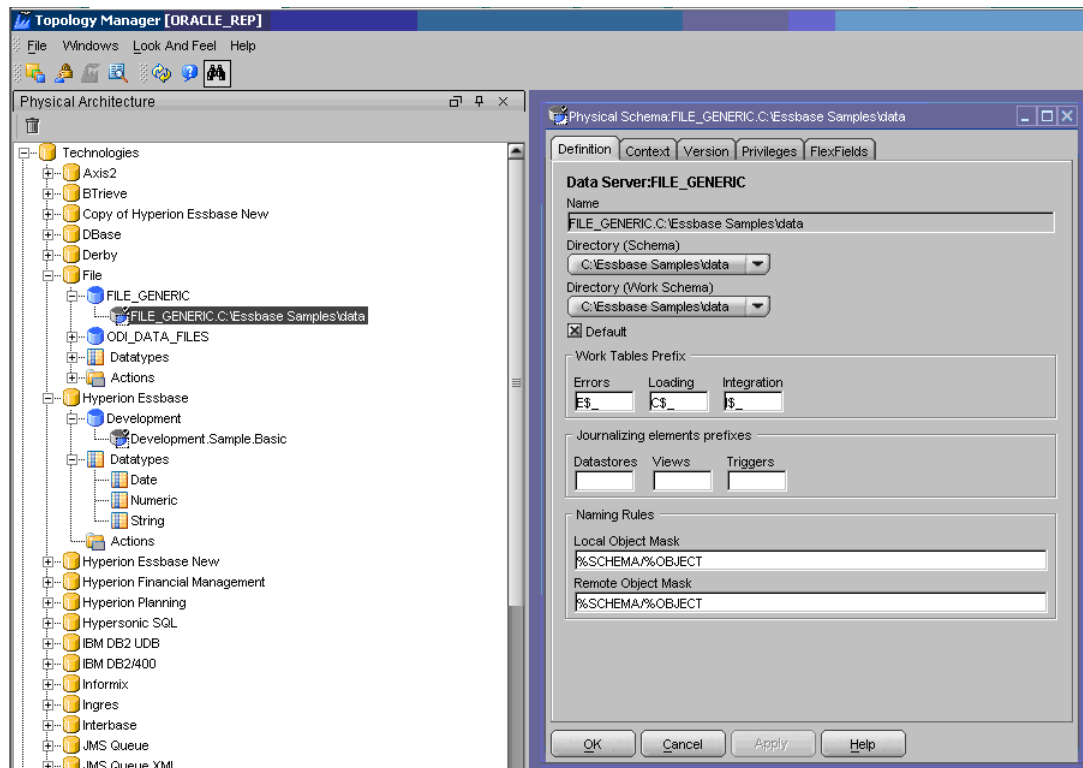
Use Oracle Data Integrator to create a physical schema for the File technology, and create a physical schema for the File data server.

Note:

This procedure is required because the sources for the samples are delimited flat files.

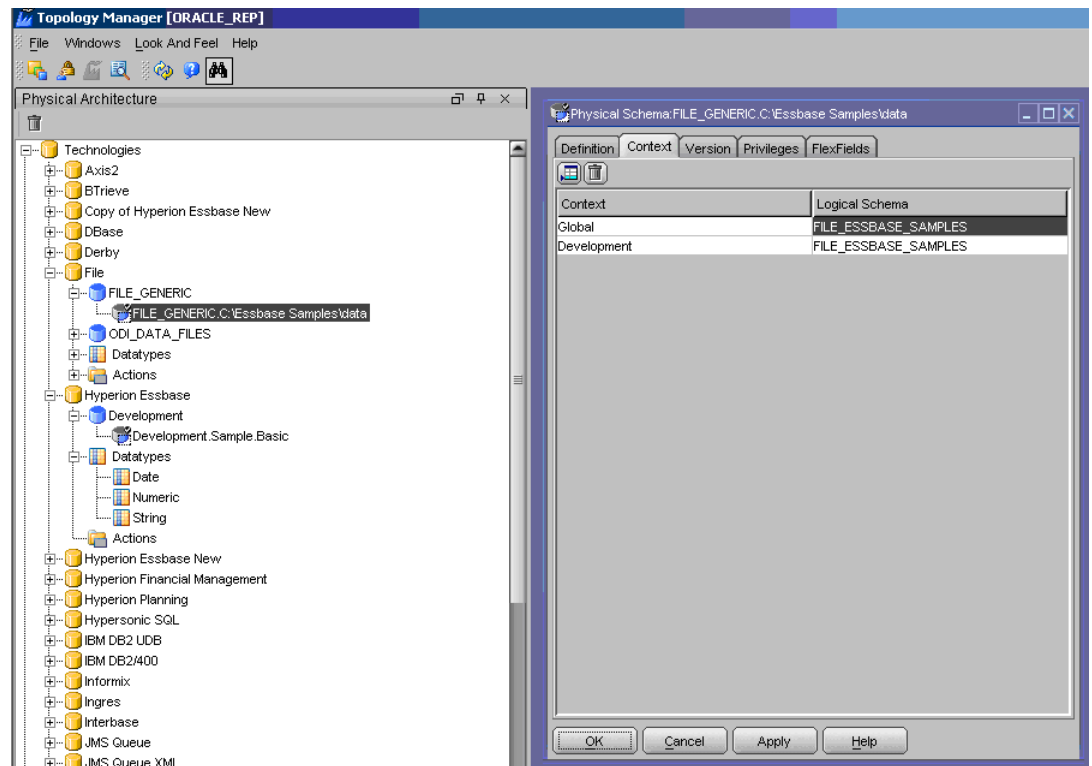
- To set up a File physical schema:

- 1 Using the Topology Manager, create a physical schema under the File technology for the FILE_GENERIC data server.
- 2 In **Directory (Schema)**, select the **data** directory (extracted from `odiaess_93110_samples.zip`), as shown:



The data folder contains the sample source files.

- 3 **On Context:**
 - a. Set the Context to Development.
 - b. For Logical Schema, enter `FILE_ESSBASE_SAMPLES`, as shown:



c. Click OK.

Configuring the Sunopsis Engine Physical Schema

When creating interfaces for sources and targets that do not store data in a relational database format, you must select a staging area other than the source or target; as an option, you can configure and use Oracle Data Integrator's staging area—Sunopsis Memory Engine.

- To configure the Sunopsis Memory Engine:
 - 1 From Topology Manager, under **Technologies**, traverse to **Sunopsis Engine > SUNOPSIS_MEMORY_ENGINE > SUNOPSIS_MEMORY_ENGINE_Default**, right-click **SUNOPSIS_MEMORY_ENGINE_Default** and click **Edit**.
 - 2 On **Context**, insert a new line, and set the **Context** to **Development** and Logical Schema to **SUNOPSIS_MEMORY_ENGINE**, and click **OK**.

Setting Up a Work Repository

The `odiaess_93110_samples.zip` that is delivered with Oracle Data Integrator Adapter for Essbase includes a work repository export file called `Work_Repository.zip`. (For more information about `odiaess_93110_samples.zip`, see [“Sample Files for Practice” on page 2.](#))

The `odiaess_93110_samples.zip` file contains the Oracle Data Integrator models, interfaces, packages, and Knowledge Modules (KMs) that are required for loading and extracting metadata

and data in the sample Essbase application. Use Oracle Data Integrator to create a work repository for your work with the Adapter for Essbase samples and import `Work_Repository.zip` into the work repository.

► To set up a work repository:

- 1 Using the Topology Manager, connect to a master repository and create a work repository named `ODI_ESSBASE_SAMPLE`. See the *Oracle Data Integrator User's Guide* for instructions.
- 2 Launch Designer, and connect to the `ODI_ESSBASE_SAMPLE` work repository.
- 3 Select **File > Import > Work Repository**.
- 4 Select an import mode.
The `INSERT_UPDATE` mode is recommended.
- 5 Select **Import From a ZIP File**, and navigate to the folder containing the file `Work_Repository.zip`, and click **OK**.

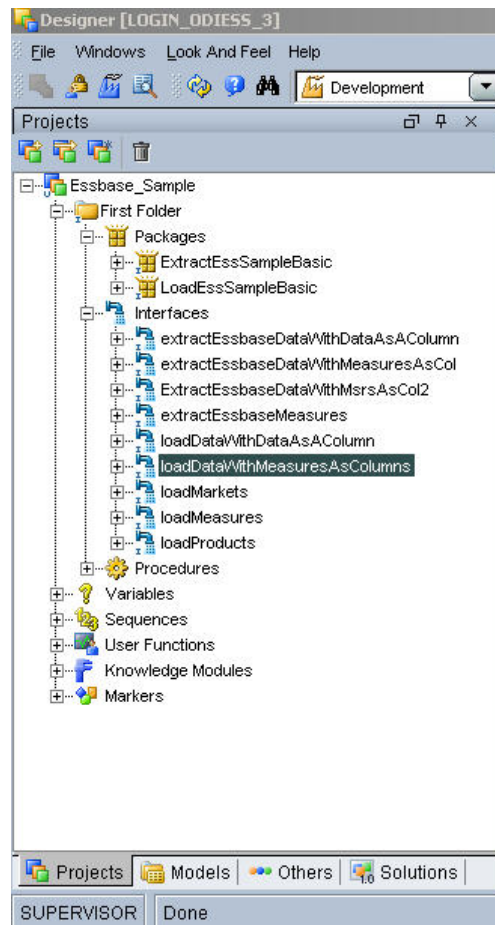
These interfaces are added to the work repository:

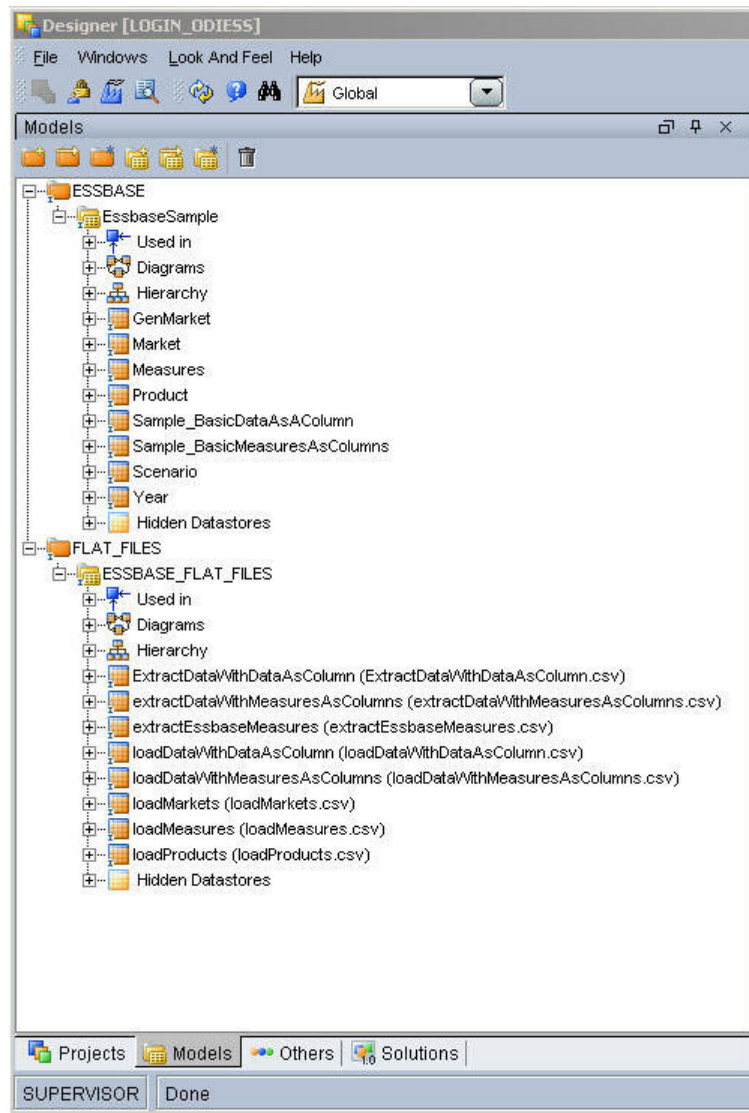
- `loadMeasures`
- `loadProducts`
- `loadMarkets`
- `loadDataWithDataAsAColumn`
- `loadDataWithMeasuresAsColumns`
- `extractEssbaseDataWithDataAsAColumn`
- `extractEssbaseDataWithMeasuresAsCol`
- `extractEssbaseDataWithMsrsAsCol2`
- `extractEssbaseMeasures`

The import also adds two packages:

- `LoadEssSampleBasic`, which chains the interfaces for loading metadata and data
- `ExtractEssSampleBasic`, which chains the interfaces for extracting data and members

These figures show how the Projects and Models trees look when the import succeeds:





Loading and Extracting Metadata or Data Using Sample Interfaces

The work repository, which was set up as part of the environment, contains interfaces for loading and extracting metadata and data from an Essbase application.

Loading Metadata into the Sample Application

Load the Essbase application—Sample with metadata following the procedures in the following topics.

Loading Measures Dimension Metadata

The sample files provided for practice includes an interface called loadMeasures, which loads metadata into the Measures dimension.

Note:

For instructions on building this interface or others like it, see [“Creating an Interface to Load Metadata” on page 20](#).

- To load metadata into the Measures dimension:
 - 1 Run the loadMeasures interface.
 - 2 Check the Operator log to see whether the interface ran successfully.
 - 3 Validate the Measures dimension from Oracle's Essbase® Administration Services Windows client.

Loading Products Dimension Metadata

The sample files provided for practice includes an interface called loadProducts, which loads metadata into the Products dimension.

- To load metadata into the Products dimension:
 - 1 Run the loadProducts interface.
 - 2 Check the Operator log to see whether the interface ran successfully.
 - 3 Validate the Product dimension from Administration Services Windows client.

Loading Markets Dimension Metadata

The sample files provided for practice includes an interface called loadMarkets, which loads metadata into the Markets dimension.

- To load metadata into the Markets dimension:
 - 1 Run the loadMarkets interface.
 - 2 Check the Operator log to see whether the interface ran successfully.
 - 3 Validate the Markets dimension from Administration Services Windows client.

Loading Data into the Sample Application

The sample files provided for practice includes two interfaces to load data:

- loadDataWithDataAsAColumn
- loadDataWithMeasuresAsColumns

Use these interfaces to load data into the Essbase application—Sample and database—Basic.

Note:

Before completing the following procedure, verify that the Essbase application—Sample and database—Basic contains the required metadata. If it does not, load the metadata as outlined in *Chapter 6. Setting Up Sample Applications* in the *Hyperion Essbase – System 9 Installation Guide for Windows*.

- To load data into an Essbase application—Sample and database—Basic:
 - 1 Run any interface—loadDataWithDataAsAColumn or loadDataWithMeasuresAsColumns.
 - 2 Check the Operator log to see whether the interface ran successfully.
 - 3 Validate the data load from Administration Services Windows client.

Extracting Data from the Sample Application

The sample files provided for practice includes two interfaces for extracting data from the Essbase application—Sample and database—Basic and writing the data to a file.

- extractEssbaseDataWithDataAsAColumn (Uses Essbase report script to extract data)
 - extractEssbaseDataWithMeasuresAsCol (Uses the Essbase report script query to extract data)
 - ExtractEssbaseDataWithMsrsAsCol2 (Uses the Essbase MDX query to extract data)
- To extract data from an Essbase application—Sample and database—Basic:
 - 1 Run any interface—extractEssbaseDataWithDataAsAColumn, extractEssbaseDataWithMeasuresAsCol, or ExtractEssbaseDataWithMsrsAsCol2.
 - 2 Check the Operator log to see whether the interface ran successfully.
 - 3 Validate the extracted data in the file `extractData.dat` in the data directory (extracted from `odiaess_93110_samples.zip`).

Extracting Metadata from the Sample Application

The sample files provided for practice includes an interface called `extractEssbaseMeasures`, which extracts metadata from the Measures dimension from the Essbase application—Sample and database—Basic and writes them to a file.

- To extract metadata from an Essbase application—Sample and database—Basic:
 - 1 Run the `extractEssbaseMeasures` interface.
 - 2 Check the Operator log to see whether the interface ran successfully.
 - 3 Validate the extracted data in the file `extractEssbaseMeasures.csv` in the data directory (extracted from `odiaess_93110_samples.zip`).

Creating Models

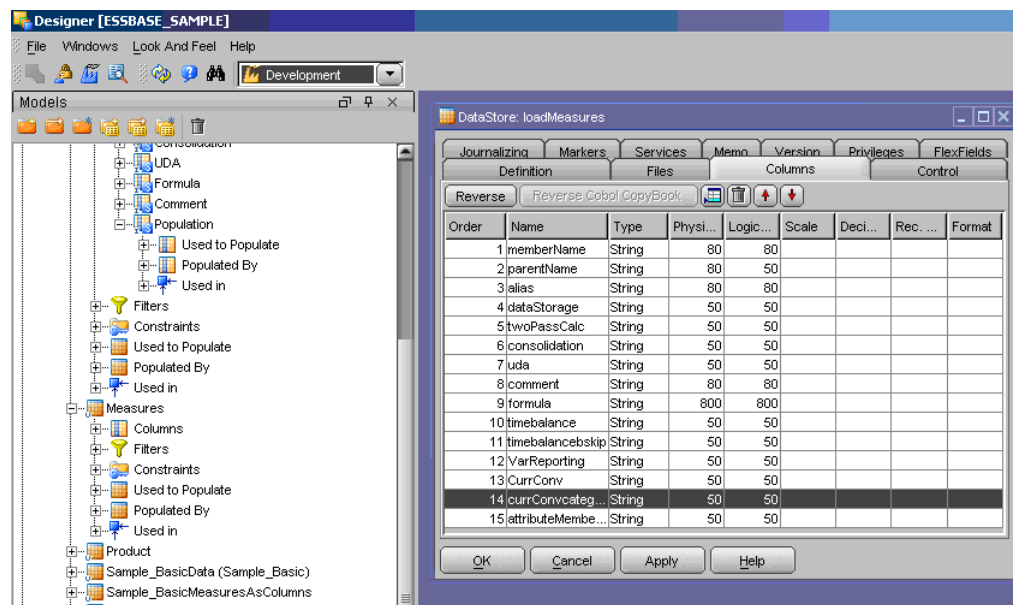
You must create source and target models before you create interfaces.

Creating and Reverse-Engineering the Sample Source Models

Use Oracle Data Integrator Designer to create and reverse-engineer the Adapter for Essbase sample source models.

- To create and reverse-engineer the sample source models:
 - 1 In the **Models** view, insert a new model folder called `FLAT_FILES`.
 - 2 Right-click the **FLAT_FILES** model folder, and select **Insert Model**.
 - 3 Name the model `ESSBASE_FLAT_FILES`, and set **Technology** to **File** and **Logical Schema** to `FILE_ESSBASE_SAMPLES`.
 - 4 On **Reverse**, set **Context** to **Development**, and click **OK**.
 - 5 Right-click **ESSBASE_FLAT_FILES**, and select **Insert Datastore**.
 - 6 Set **Name** to `loadMeasures`, `loadProducts`, `loadMarkets`, `loadDataWithDataAsColumn`, or `loadDataWithMeasuresAsColumns`, depending on which file source you are defining.
 - 7 Next to **Resource Name**, click **Browse**, and select the file for the source that you are defining (`loadMeasures`, `loadProducts`, `loadMarkets`, `loadDataWithDataAsColumn`, or `loadDataWithMeasuresAsColumns`).
 - 8 On **Files**, set **File Format** to **Delimited**, **Heading (Number of lines)** to **1**, and **Field Separator** to **,** (comma) for `.txt` and `.csv` files and **;** (semicolon) for `.dat` files.
 - 9 On **Columns**, click **Reverse**.

This figure shows how the page should look:

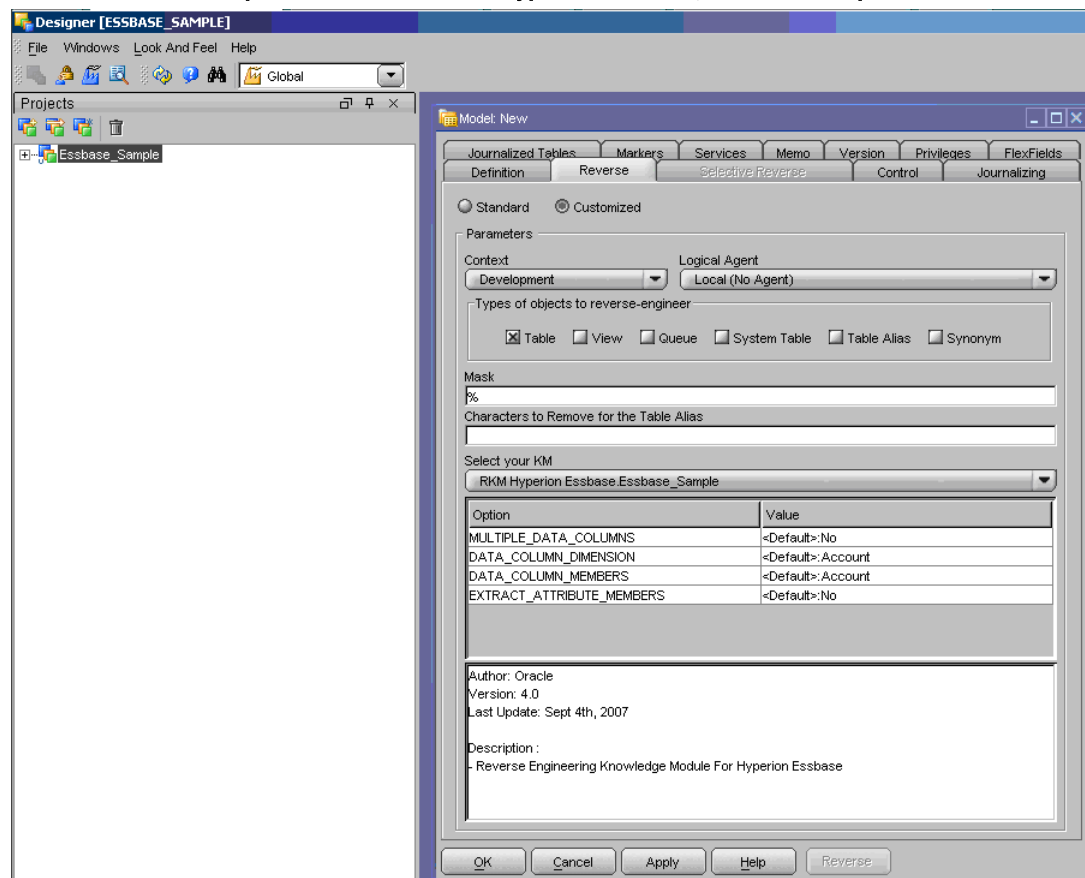


- 10 Ensure that the fields corresponding to numeric fields in the Essbase application are set as numeric, with the correct length and scale.
- 11 Repeat [step 5](#) through [step 10](#) for each remaining file source.

Creating and Reverse-Engineering the Sample Target Models

Use Oracle Data Integrator Designer to create and reverse-engineer the Adapter for Essbase target models.

- To reverse-engineer the sample target models:
 - 1 In the **Models** view, insert a new model folder called **ESSBASE**.
 - 2 Right-click the **ESSBASE** model folder and select **Insert Model**.
 - 3 Name the model **EssbaseSample**, set **Technology** to **Hyperion Essbase**, and set **Logical Schema** to **SampleEssbase**.
 - 4 On **Reverse**, select **Customized** (at the top of the page).
 - 5 Set **Context** to **Development**, and select **RKM Hyperion Essbase, Essbase Sample**, as shown:

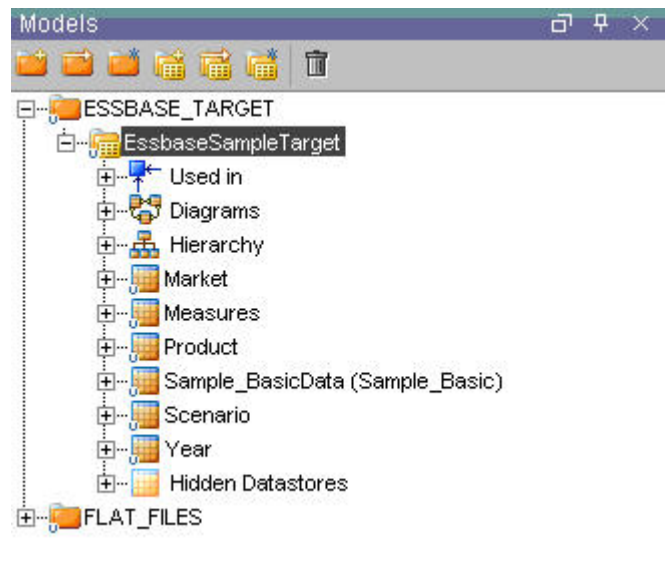


- 6 Set these Reverse-engineering Knowledge Module (RKM) options.

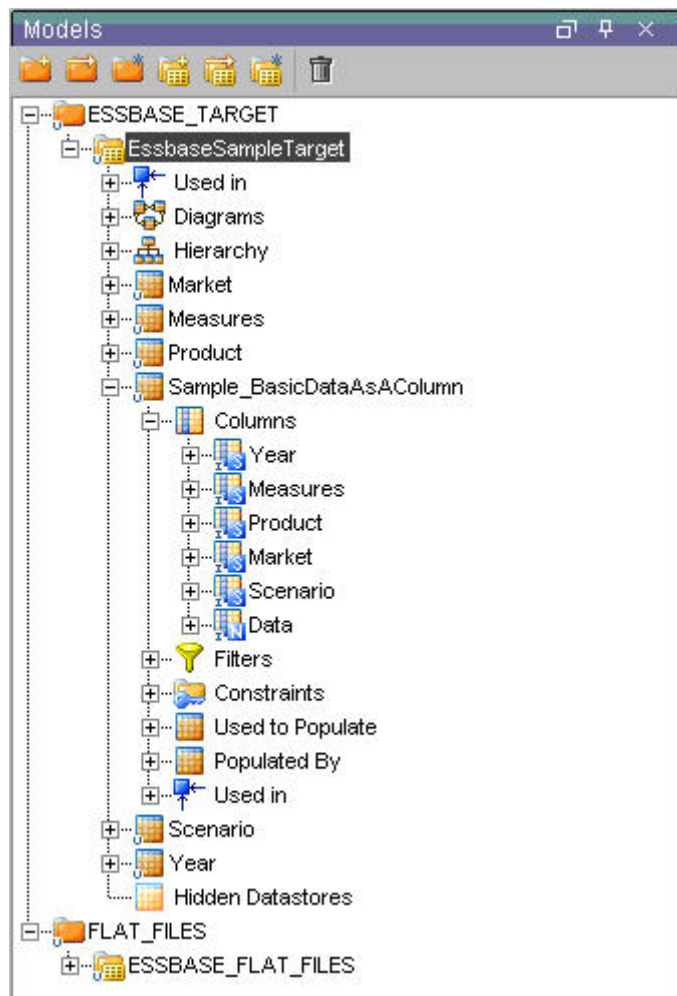
Option	Value	Description
MULTIPLE_DATA_COLUMNS	No (Default)	<p>If this option is set to No, then the datastore created for the data extract / load model contains one column for each of the standard dimensions and a single data column.</p> <p>If this option is set to Yes, then the datastore created for the data extract / load model contains one column for each of the standard dimensions excluding the dimension specified by the DATA_COLUMN_DIMENSION option and as many data columns as specified by the comma separated list for the DATA_COLUMN_MEMBERS option.</p>
DATA_COLUMN_DIMENSION	Account (Default)	<p>This option is only applicable if MULTIPLE_DATA_COLUMNS is set to Yes.</p> <p>Specify the data column dimension name.</p> <p>For example, data columns are spread across the dimension Account or Time, and so on.</p>
DATA_COLUMN_MEMBERS	Account (Default)	<p>This option is only applicable if MULTIPLE_DATA_COLUMNS is set to Yes.</p> <p>Separate the required data column members with , (Comma).</p> <p>For example, if the data column dimension is set to Account and members are set to "Sales,COGS", then the datastore for data extract/load contains one column for each of the dimension except the data column dimension and one column for each of the data column member specified in the comma separated value. For example. Assuming that the dimensions in the Essbase application are Account, Scenario, Product, Market, and Year and the data column dimension is specified as Account and Data Column Members as "Sales, COGS", the datastore will have the following columns:</p> <ul style="list-style-type: none"> ● Scenario (String) ● Product (String) ● Market (String) ● Year (String) ● Sales (Numeric) ● COGS (Numeric)
EXTRACT_ATTRIBUTE_MEMBERS	No	<p>If this option is set to No, then the datastore created for the data extract / load model contains one column for each of the standard dimensions and a single data column. Attribute dimensions are not included.</p> <p>If this option is set to Yes, then the data model contains these columns.</p> <ul style="list-style-type: none"> ● One column is created for each of the standard dimensions ● One or more data column(s) are created depending upon the value of the MULTIPLE_DATA_COLUMN option ● One column is created for each of the associated attribute dimension

- 7 Click **Reverse**, click **Yes** to validate the changes, and click **OK**.

This figure shows how the EssbaseSample models are displayed when reverse-engineering has succeeded:



- 8 From the sample, another data load target will be reversed to demonstrate loading with measures as the data load column. First Rename the existing **Sample_BasicData** target to **Sample_BasicDataAsAColumn** by right-clicking on the **Sample_BasicData** target clicking **Edit** and change the fields, name, alias and resource to **Sample_BasicDataAsAColumn**. The **EssbaseSample** model should contain the following dimensions:



- 9 To reverse the new data load target right-click **EssbaseSample** and click **Edit**.
- 10 Click **Reverse** and change the RKM options to values as shown in the following figure:

Model: EssbaseSampleTarget

Journalized Tables Markers Services Memo Version Privileges FlexFields

Definition Reverse **Selective Reverse** Control Journalizing

☐ Standard ☒ Customized

Parameters

Context: Development Logical Agent: Local (No Agent)

Types of objects to reverse-engineer

☒ Table ☐ View ☐ Queue ☐ System Table ☐ Table Alias ☐ Synonym

Mask: %

Characters to Remove for the Table Alias:

Table Alias maximum length: 35

RKM Hyperion Essbase.Essbase_Sample

Option	Value
MULTIPLE_DATA_COLUMNS	Yes
DATA_COLUMN_DIMENSION	measures
DATA_COLUMN_MEMBERS	sales,cogs
EXTRACT_ATTRIBUTE_MEMBERS	<Default>:No

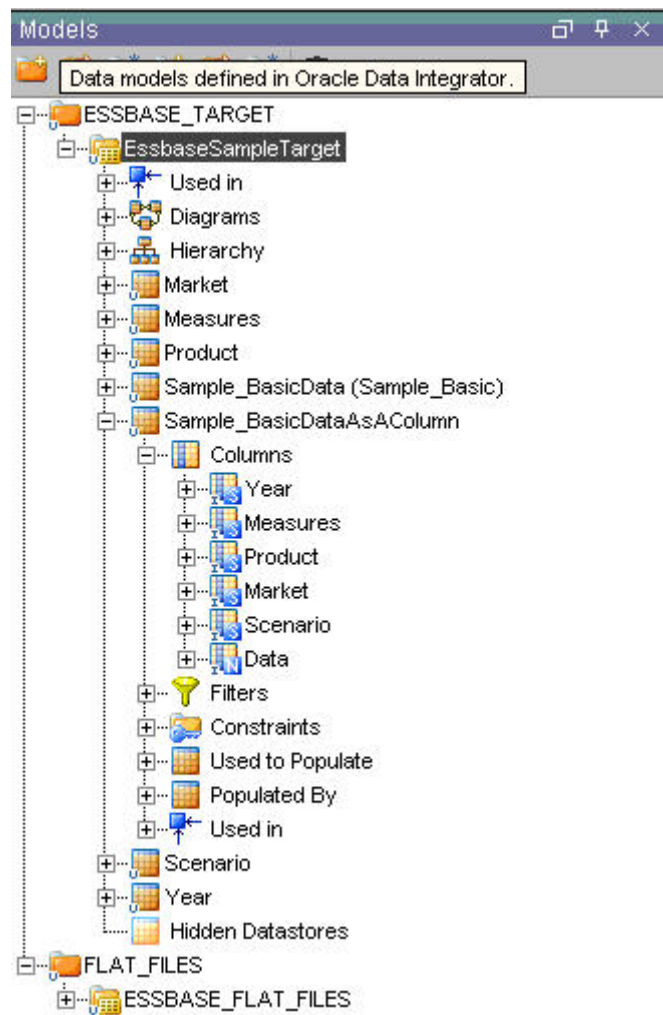
If this option is set to No, then the datastore created for the Data extract / load model would contain one column for each of the standard dimensiona and a single Data column.

If this option is set to Yes, then the datastore would contain one column for each of the standard dimensions excluding the dimension specified by DATA_COLUMN_DIMENSION option and as many data columns as specified by the comma separated list for the DATA_COLUMN_MEMBERS option.

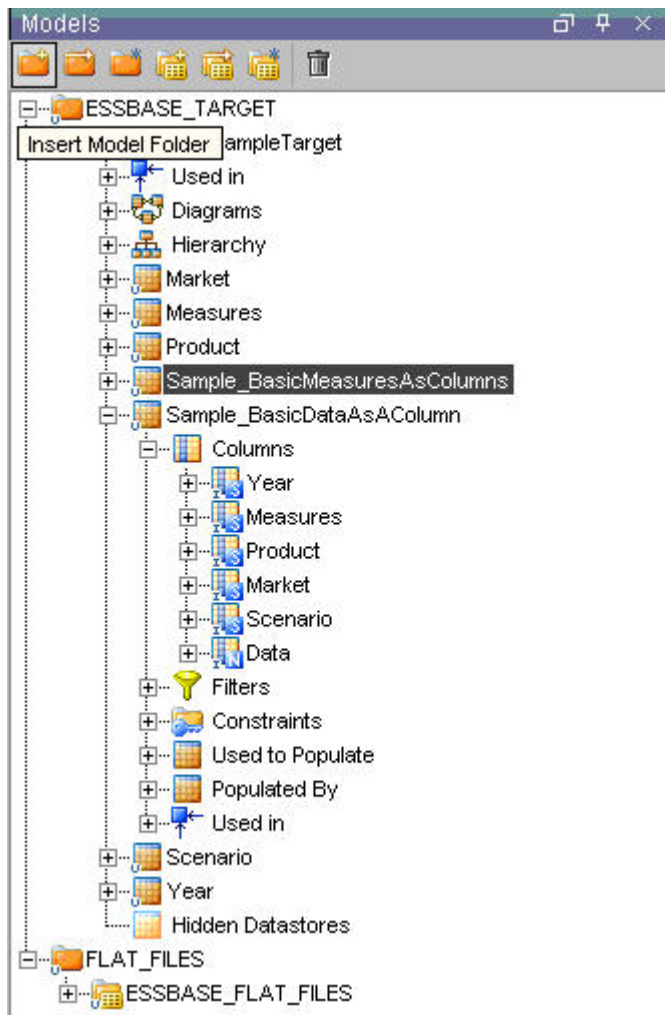
OK Cancel Apply Help Reverse

- 11 Click **Reverse**, click **Yes** to validate the changes, and click **OK**.

This figure shows how the EssbaseSample models are displayed when reverse-engineering has succeeded:



- 12 First Rename the existing **Sample_BasicData** target to **Sample_BasicMeasuresAsColumns** by right-clicking on the **Sample_BasicData** target clicking **Edit** and change the fields, name, alias and resource to **Sample_BasicMeasuresAsColumns**. The **EssbaseSample** model should contain the following dimensions:



If the EssbaseSample models are not displayed, check the Operator log to determine why reverse-engineering failed.

Creating Interfaces and Packages

Creating an Interface to Load Metadata

You can create an interface for loading the Measures dimension into the Essbase application—Sample and database—Basic. Using this interface as a model, you can create interfaces for loading the Products and Markets dimensions with corresponding sources and targets.

You can also chain the interfaces into a package so that you can run them in a single process. See [“Creating a Package to Load Metadata and Data” on page 27](#).

- To create an interface for loading the Measures dimension:
 - 1 Launch Designer, and expand the **Interfaces** node under the **Essbase_Sample** project.
 - 2 Right-click **Interfaces** and select **Insert Interface**.

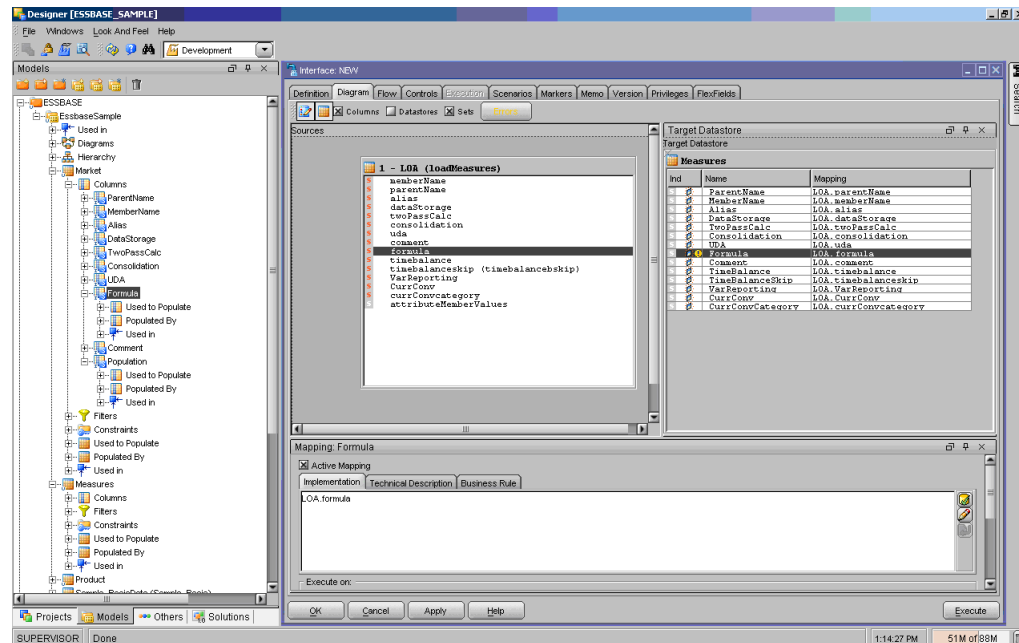
- 3 Name the interface `loadMeasures`, and set **Optimization Context** to **Development**.
- 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema”](#) on page 7.

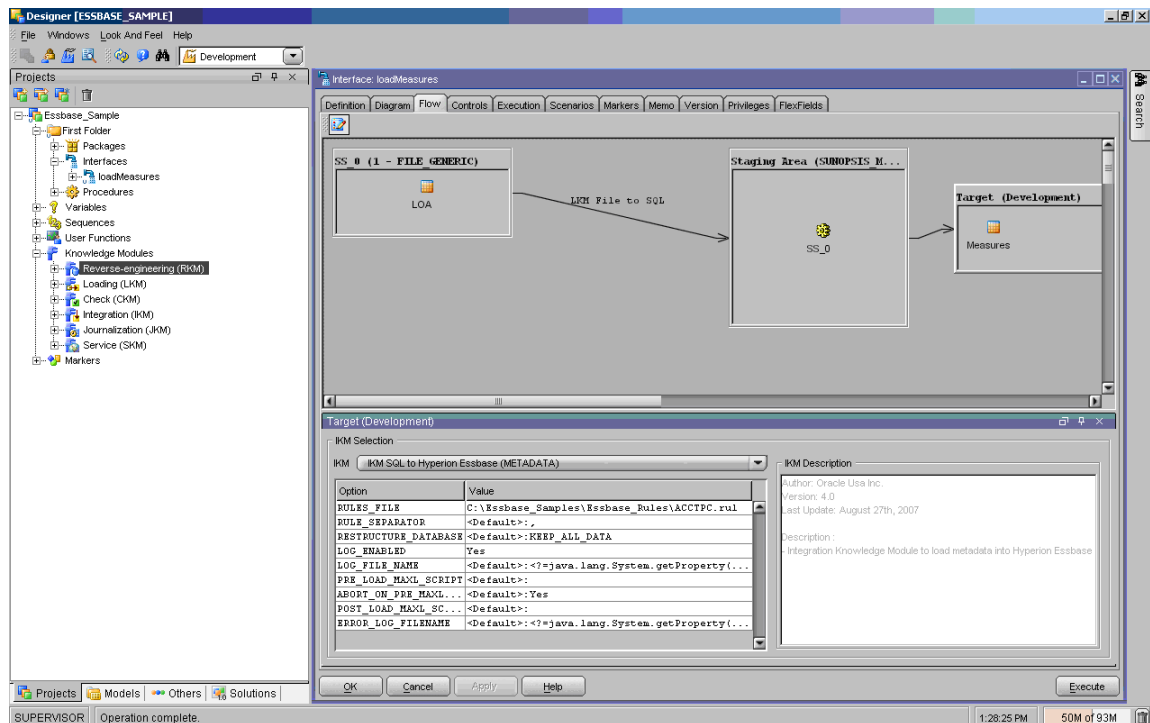
- 5 Select **Diagram**.
- 6 In the **Models** view, drag **Measures** datastore from the **Essbase/EssbaseSample** model to the **Target Datastore** pane.
- 7 Drag the **loadMeasures** source from the **FLAT_FILES/ESSBASE_FLAT_FILES** model to the **Sources** area.
A message that prompts you to use automatic mapping is displayed.
- 8 Click **Yes**.
- 9 Manually map any columns that were not mapped automatically.

This figure shows how the page should look:



- 10 On **Flow**, select the **SS_0**, and ensure that **LKM** is set to **LKM File to SQL**.
- 11 Select **Target**, and ensure that **IKM** is set to **IKM SQL to Hyperion Essbase (METADATA)**.
- 12 Set these Integration Knowledge Module (IKM) options.

```
RULES_FILE C:\Essbase_Samples\Essbase_Rules\ACCTPC.rul
PRE_LOAD_MAXL_SCRIPT C:\Essbase_Samples\MAXL\premaxl.mxl
```

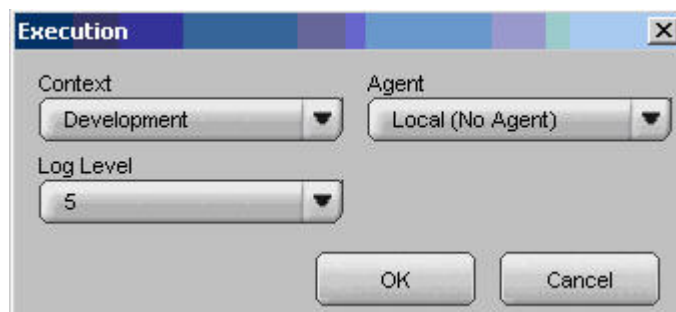


Note:

The rules file for each metadata load is included in the rules directory of the `odiaess_93110_samples.zip` file. The MAXL file is also included the MAXL directory of the `odiaess_93110_samples.zip` file.

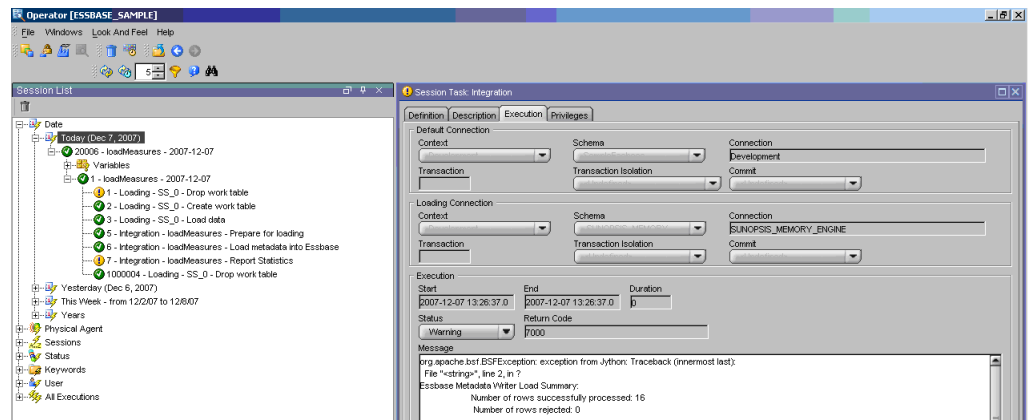
13 Click **Apply**.

14 Click **Execute** to run the LoadMeasures Interface, and, under **Context**, select **Development**.



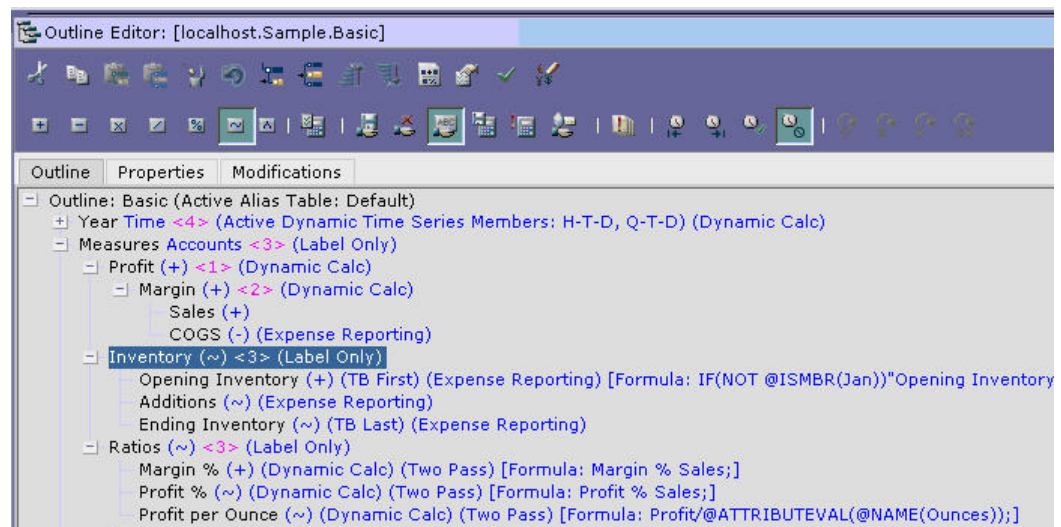
15 View the results of running the interface in Oracle Data Integrator Operator.

This figure shows how the results should look:



16 Validate the Measures dimension from Oracle's Essbase® Administration Services client.

These figure shows the hierarchies created in Essbase.



Creating an Interface to Load and Calculate Data (Load File format—Data as a column)

- To create an interface for loading and consolidating data:
 - 1 Launch Designer, and expand the Interfaces node under the **EssbaseSample** project.
 - 2 Right-click, and select **Insert Interface**.
 - 3 Name the interface `loadDataWithDataAsAColumn`, and set **Optimization Context** to **Development**.
 - 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

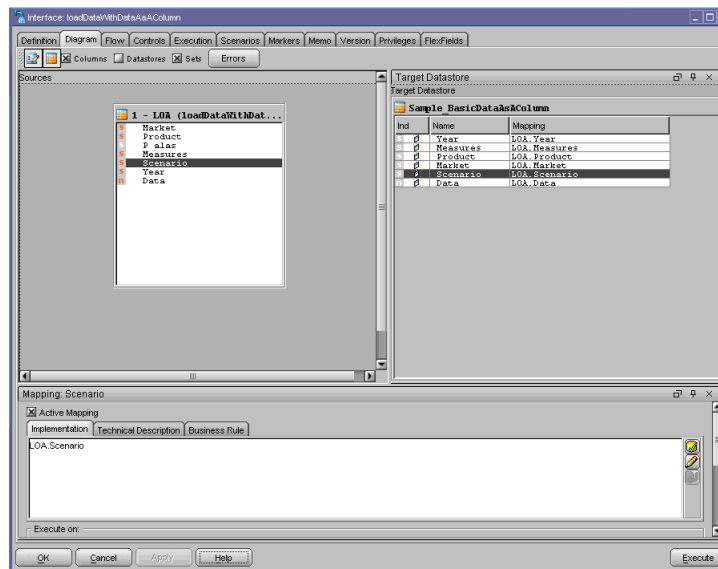
If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema”](#) on page 7.

- 5 Select **Diagram**.
- 6 In the **Models** view, drag **Sample_BasicDataAsAColumn** datastore from the **ESSBASE/EssbaseSample** model to the **Target Datastore** pane.
- 7 Drag the **loadDataWithDataAsColumn** source from the **FLAT_FILES/ ESSBASE_FLAT_FILES** model to the **Sources** area.

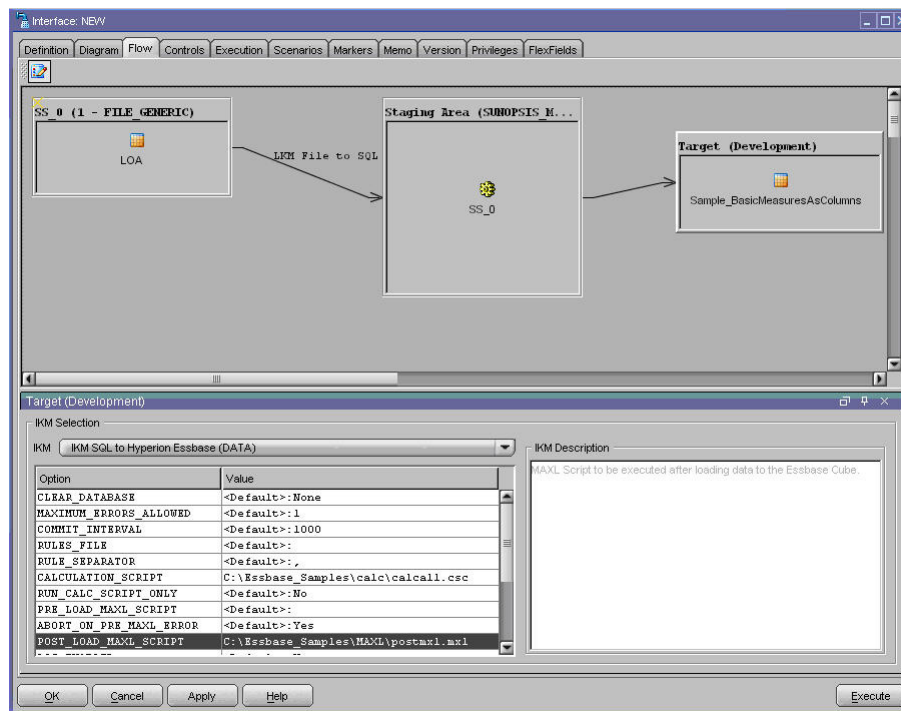
A message that prompts you to use automatic mapping is displayed.

- 8 Click **Yes**.
- 9 Manually map any columns that were not mapped automatically.

This figure shows how the page should look when you finish:

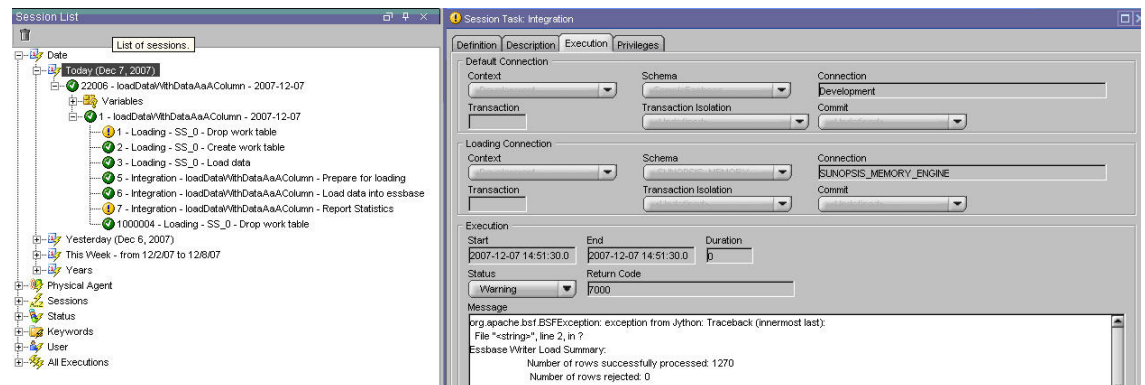


- 10 On **Flow**, select the **SS_0**, and ensure that **LKM** is set to **LKM File to SQL**.
- 11 Select **Target**, and ensure that **IKM** is set to **IKM SQL to Hyperion Essbase (Data)**.
- 12 Set these **IKM** options:
 - CALCULATION_SCRIPT C:\Essbase_Samples\Calc_Scripts\calcall.csc
 - POST_LOAD_MAXL_SCRIPT C:\Essbase_Samples\MAXL\postmxl.mxl



- 13 Click **Apply**.
- 14 Click **Execute** to run the loadDataWithDataAsAColumn interface, and select **Development** as the context.
- 15 View the results of running the interface in Oracle Data Integrator Operator.

This figure shows how the results should look:



Creating an Interface to Load and Calculate data (Load File Format—Measures as columns)

- To create an interface for loading and consolidating data:
 - 1 Launch Designer, and expand the **Interfaces** node under the **EssbaseSample** project.
 - 2 Right-click, and select **Insert Interface**.

- 3 Name the interface `loadDataWithMeasuresAsColumns`, and set **Optimization Context** to **Development**.
- 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

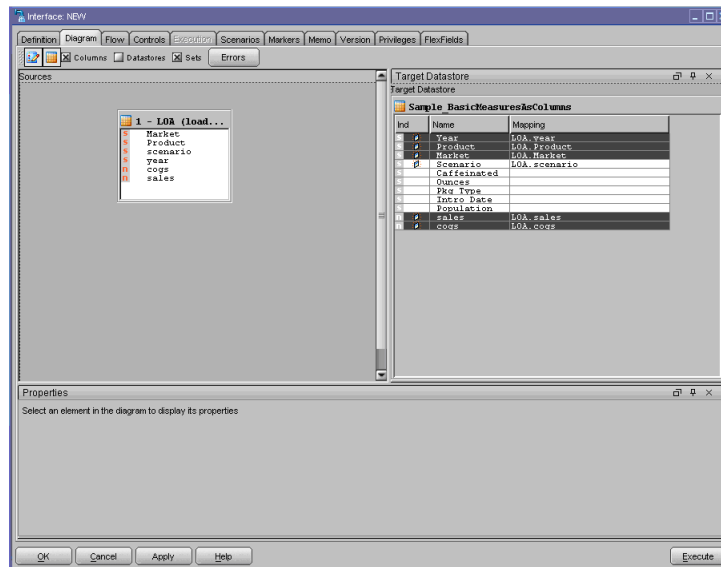
If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema” on page 7](#).

- 5 Select **Diagram**.
- 6 In the **Models** view, drag **Sample_BasicMeasuresAsColumns** datastore from the **ESSBASE/ EssbaseSample** model to the **Sources** pane.

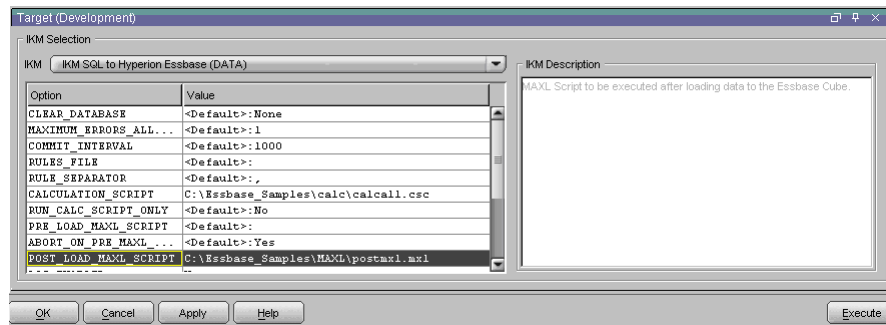
A message that prompts you to use automatic mapping is displayed.

- 7 Click **Yes**.
- 8 Manually map any columns that were not mapped automatically.

This figure shows how the page should look when you finish:

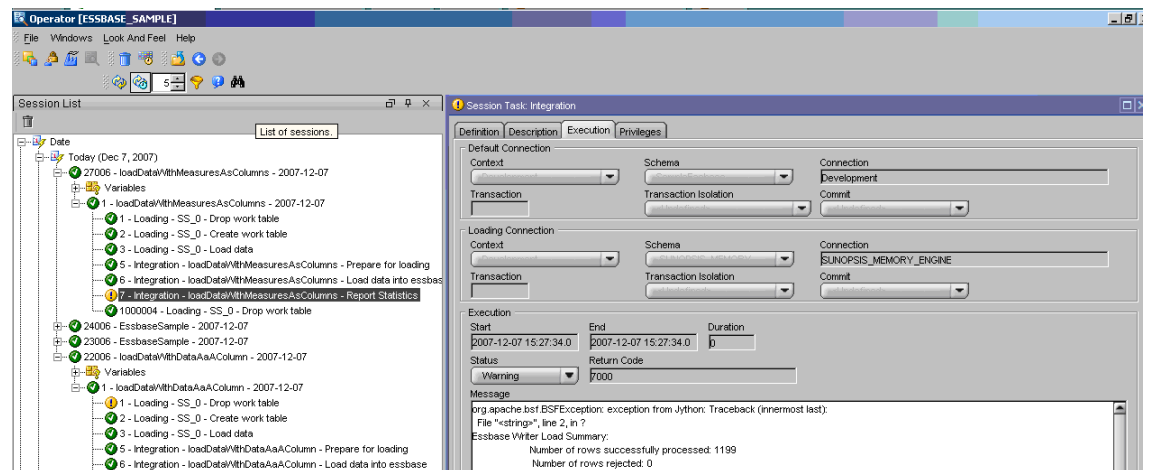


- 9 On **Flow**, select the **SS_0**, and ensure that **LKM** is set to **LKM File to SQL**.
- 10 Select **Target**, and ensure that **IKM** is set to **IKM SQL to Hyperion Essbase (Data)**.
- 11 Set these IKM options:
 - CALCULATION_SCRIPT C:\Essbase_Samples\Calc_Scripts\calcall.csc
 - POST_LOAD_MAXL_SCRIPT C:\Essbase_Samples\MAXL\postmxl.mxl



- 12 Click **Apply**.
- 13 Click **Execute** to run the `loadDataWithMeasuresAsColumns` interface, and select **Development** as the context.
- 14 View the results of running the interface in Oracle Data Integrator Operator.

This figure shows how the results should look:

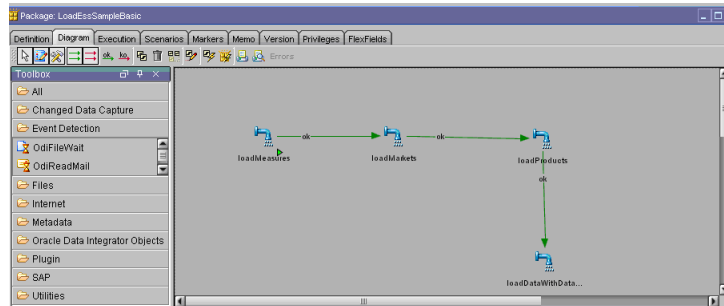


Creating a Package to Load Metadata and Data

You can chain interfaces into a package so that you can run them in a single process.

- To create a package for loading metadata and data:
 - 1 Launch Designer.
 - 2 Right-click **Packages**, and select **Insert Package**.
 - 3 Name the package `LoadEssSampleBasic` (or any other name).
 - 4 Select **Diagram**.
 - 5 Drag the `loadMeasures`, `loadMarkets`, `loadProducts`, and `loadDataWithDataAsAColumn` interfaces into the diagram area.
 - 6 Connect the interfaces in sequence, using the **ok→** green arrows.
 - 7 Click **Apply**.

This figure shows how the page should look:



- 8 Click **Execute** to run LoadEssSampleBasic Package, and select **Development** as the context.
- 9 View the results of running the LoadEssSampleBasic package in Oracle Data Integrator Operator.

This figure shows how the results should look in Operator:

Status	Step number	Step name	Start	Duration	Insert
✓	1	extractEssbaseMeasures	2007-12-10	1	17

The Session List on the left shows the following steps:

- Today (Dec 10, 2007)
 - 86006 - LoadEssSampleBasic - 2007-12-10
 - Variables
 - 1 - loadMeasures - 2007-12-10
 - 2 - loadMarkets - 2007-12-10
 - 3 - loadProducts - 2007-12-10
 - 4 - loadDataWithDataAsColumn - 2007-12-10
 - 22 - Loading - SS_0 - Drop work table
 - 23 - Loading - SS_0 - Create work table
 - 24 - Loading - SS_0 - Load data
 - 25 - Integration - loadDataWithDataAsColumn - Prepare for loading
 - 27 - Integration - loadDataWithDataAsColumn - Load data into essbase
 - 28 - Integration - loadDataWithDataAsColumn - Report Statistics
 - 1000025 - Loading - SS_0 - Drop work table

Creating an Interface to Extract Data (Using a Report Script)

- To create an interface for extracting data:
 - 1 Launch Designer, and expand the **Interfaces** node under the **EssbaseSample** project.
 - 2 Right-click, and select **Insert Interface**.
 - 3 Name the interface **extractEssbaseDataWithDataAsAColumn**, and set **Optimization Context** to **Development**.
 - 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

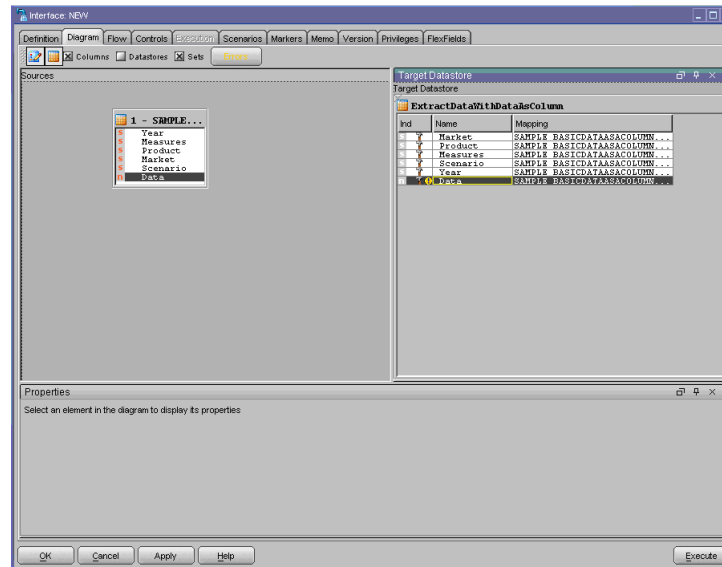
If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema”](#) on page 7.

- 5 Select **Diagram**.
- 6 In the **Models** view, drag **ExtractDataWithDataAsColumn** datastore from the **FLAT_FILES / ESSBASE_FLAT_FILES** model to the **Target Datastore** pane.
- 7 Drag the **Sample_BasicDataAsAColumn** source from the **Essbase/EssbaseSample** model to the **Sources** area.

A message that prompts you to use automatic mapping is displayed.

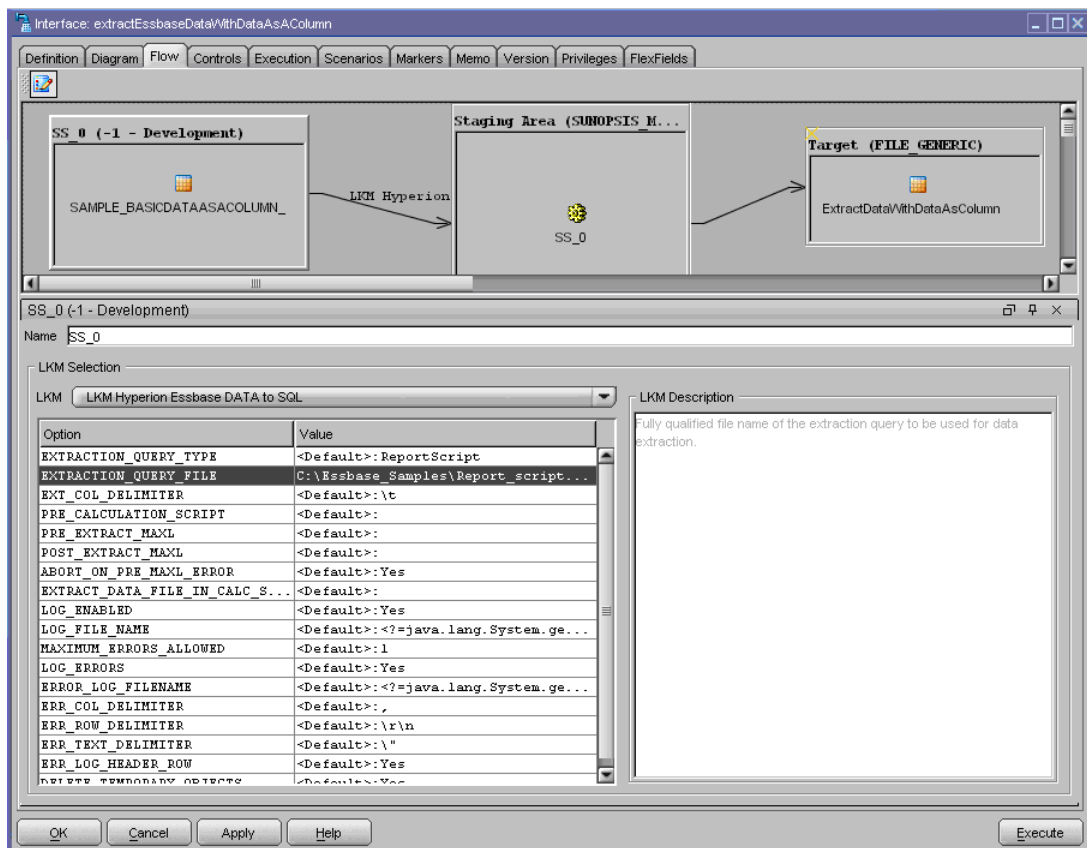
- 8 Click **Yes**.
- 9 Manually map any columns that were not mapped automatically.

This figure shows how the page should look:



- 10 On **Flow**, select the **SS_0**, and ensure that **LKM** is set to **LKM Hyperion Essbase Data to SQL**.
- 11 Select **Target**, and ensure that **IKM** is set to **IKM SQL to File**.
- 12 Set these Loading Knowledge Module (LKM) options.

EXTRACTION_QUERY_FILE C:\Essbase_Samples\Report_scripts\extract.rep



Note:

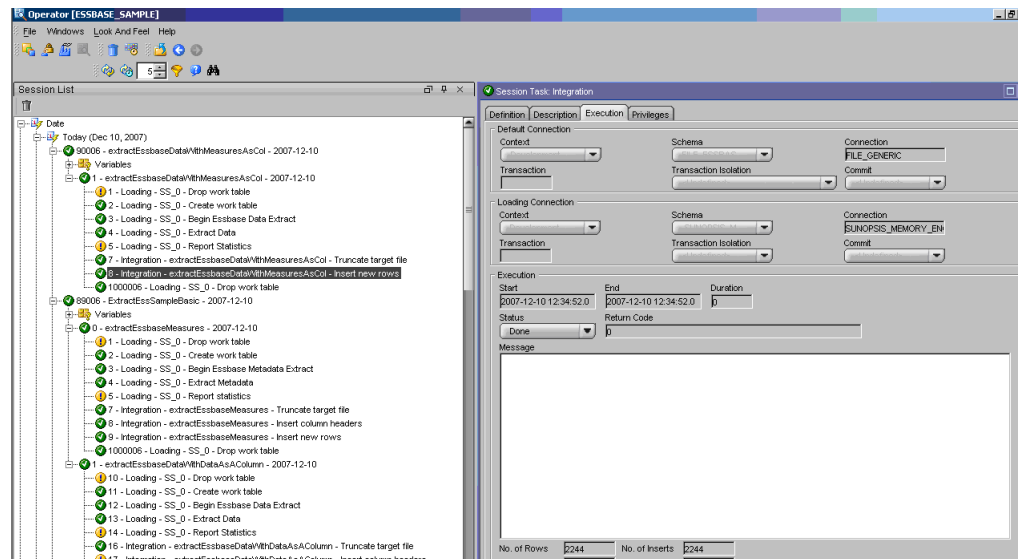
The `extract.rep` rules file is included in the rules directory within the `odiaess_93110_samples.zip`. Modify the report script to further limit what data is extracted.

13 Click **Apply**.

14 Click **Execute** to run the `extractEssbaseDataWithDataAsAColumn` interface, and select **Development** under **Context**.

View the results of running the `extractEssbaseDataWithDataAsAColumn` interface in Oracle Data Integrator Operator.

This figure shows how the results should look in Operator:



- 15 Verify that the `extractDataWithDataAsColumn.csv` file contains the data extracted from the Essbase application—Sample and database—Basic.

Creating an Interface to Extract Data (Using a MDX Query script)

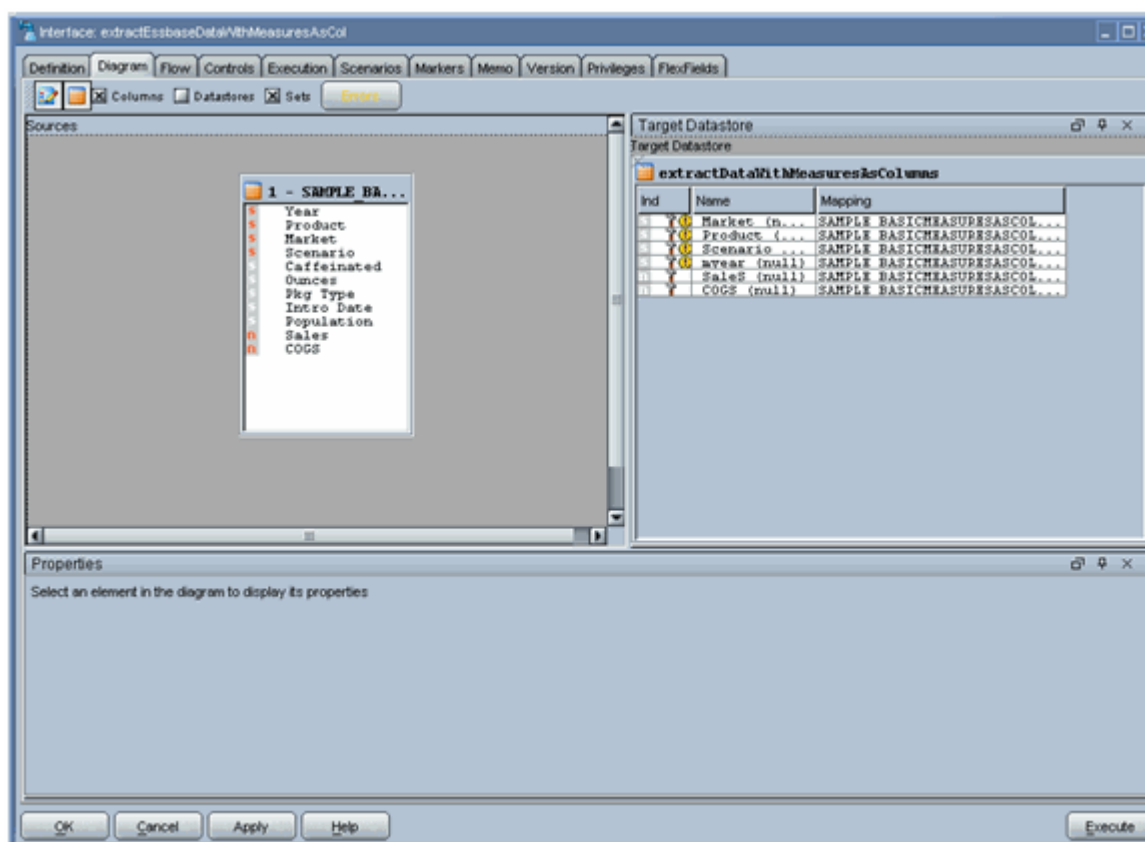
- To create an interface for extracting data:
 - 1 Launch Designer, and expand the **Interfaces** node under the **EssbaseSample** project.
 - 2 Right-click, and select **Insert Interface**.
 - 3 Name the interface `ExtractEssbaseDataWithMsrsAsCol2`, and set **Optimization Context** to **Development**.
 - 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema” on page 7](#).

- 5 Select **Diagram**.
- 6 In the **Models** view, drag `extractDataWithMeasuresAsColumns` datastore from the **FLAT_FILES / ESSBASE_FLAT_FILES** model to the **Target Datastore** pane.
- 7 Drag the `Sample_BasicMeasuresAsColumns` source from the **Essbase/EssbaseSample** model to the **Sources** area. A message that prompts you to use automatic mapping is displayed.
- 8 Click **Yes**.
- 9 Manually map any columns that were not mapped automatically.

This figure shows how the page should look:

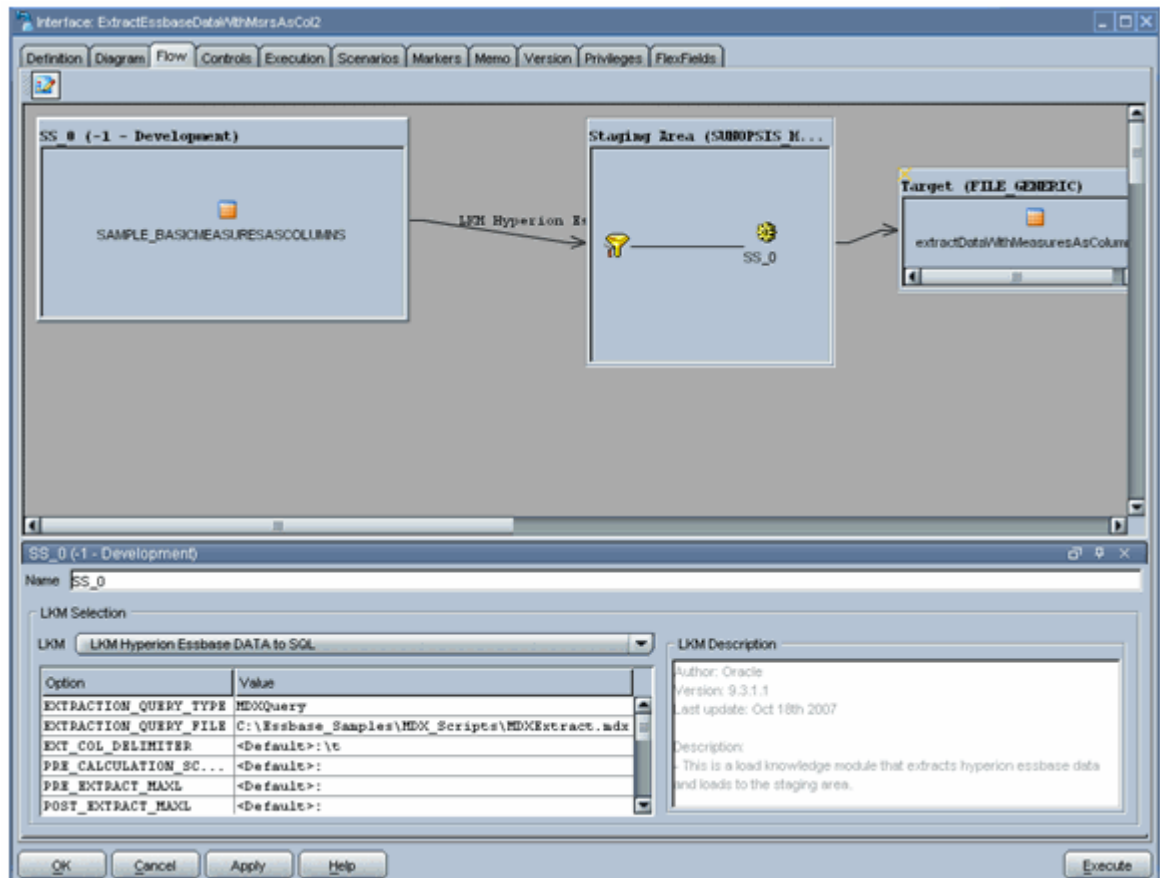


- 10 On **Flow**, select the **SS_0**, and ensure that **LKM** is set to **LKM Hyperion Essbase Data to SQL**.
- 11 Select **Target**, and ensure that **IKM** is set to **IKM SQL to File**.
- 12 Set the LKM options.

Option	Values	Description
PRE_CALCULATION_SCRIPT	Blank (Default)	(Optional) Specify the calculation script that you want to run before extracting data from the Essbase cube.
EXTRACTION_QUERY_TYPE	MDXQuery	Specify an extraction query type—report script, MDX query, or calculation script. Provide a valid extraction query, which fetches all the data to fill the output columns. The first record (first two records in case of calculation script) contains the meta information of the extracted data.
EXTRACTION_QUERY_FILE	C:\Essbase_Samples \MDX_Scripts\MDXExtract.mdx	Specify a fully qualified file name of the extraction query.

Option	Values	Description
EXT_COL_DELIMITER	\t (Default)	Specify the column delimiter for the extraction query. If no value is specified for this option, then space (" ") is considered as column delimiter.
EXTRACT_DATA_FILE_IN_CALC_SCRIPT	Blank (Default)	This option is only applicable if the query type in the EXTRACTION_QUERY_TYPE option is specified as CalcScript. Specify a fully qualified file location where the data is extracted through the calculation script.
PRE_EXTRACT_MAXL	Blank (Default)	Enable this option to execute a MAXL script before extracting data from the Essbase cube.
POST_EXTRACT_MAXL	Blank (Default)	Enable this option to execute a MAXL script after extracting data from the Essbase cube.
ABORT_ON_PRE_MAXL_ERROR	Yes	This option is only applicable if the PRE_EXTRACT_MAXL option is enabled. If the ABORT_ON_PRE_MAXL_ERROR option is set to Yes, while executing pre-MAXL script, the load process is aborted on encountering any error.
LOG_ENABLED	Yes	If this option is set to Yes, during the LKM process, logging is done to the file specified in the LOG_FILE_NAME option.
LOG_FILE_NAME	<?=java.lang.System.getProperty(java.io.tmpdir)?>/Extract_<%=snpRef.getFrom()%>.log (Default)	Specify a file name to log events of the LKM process.
MAXIMUM_ERRORS_ALLOWED	1 (Default)	Enable this option to set the maximum number of errors to be ignored before stopping extract.
LOG_ERRORS	No (Default)	If this option is set to Yes, during the LKM process, details of error records are logged to the file specified in the ERROR_LOG_FILENAME option.
ERROR_LOG_FILENAME	<?=java.lang.System.getProperty(java.io.tmpdir)?>/Extract_<%=snpRef.getFrom()%>.err	Specify a file name to log error record details of the LKM process.

Option	Values	Description
ERR_LOG_HEADER_ROW	No (Default)	If this option is set to Yes, then the header row containing the column names are logged to the error records file.
ERR_COL_DELIMITER	, (Default)	Specify the column delimiter to be used for the error records file.
ERR_ROW_DELIMITER	\r\n (Default)	Specify the row delimiter to be used for the error records file.
ERR_TEXT_DELIMITER	\\" (Default)	Specify the text delimiter to be used for the column data in the error records file. For example, if the text delimiter is set as ' \" ' (double quote), then all the columns in the error records file are delimited by double quotes.
DELETE_TEMPORARY_OBJECTS	No (Default)	Set this option to No, in order to retain temporary objects (tables, files, and scripts) after integration. This option is useful for debugging.

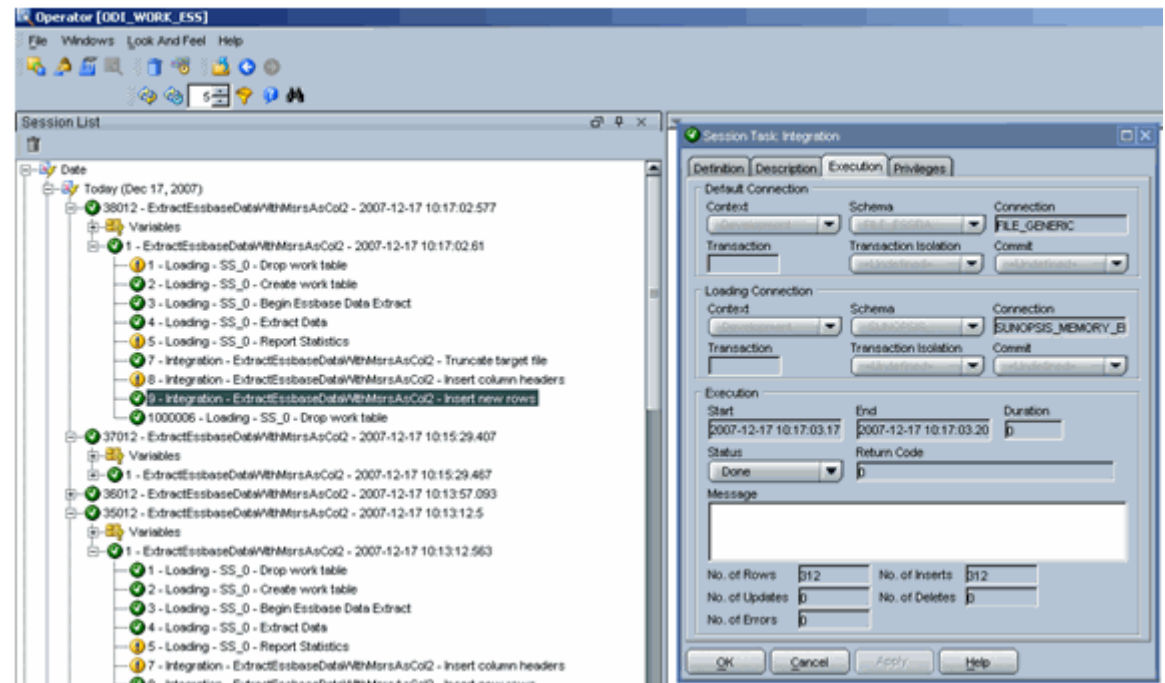


Note:

The MDXExtract.mdx rules file is included in the MDX_Scripts directory within the odiaess_93110_samples.zip.

- 13 Click **Apply**.
- 14 Click **Execute** to run the ExtractEssbaseDataWithMsrAsCol2 interface, and select **Development** under **Context**.
- 15 View the results of running the ExtractEssbaseDataWithMsrAsCol2 interface in Oracle Data Integrator Operator.

This figure shows how the results should look in Operator:



- 16 Verify that the extractDataWithDataAsColumn.csv file contains the data extracted from the Essbase application—Sample and database—Basic.

Creating an Interface to Extract Data (Using a Calculation Script)

Note:

For extracting data using calculation script, the Essbase server and the Oracle Data Integrator Agent must be running on the same computer.

- To create an interface for extracting data:
 - 1 Launch Designer, and expand the **Interfaces** node under the **EssbaseSample** project.
 - 2 Right-click, and select **Insert Interface**.

- 3 Name the interface `extractEssbaseDataWithMeasuresAsCol`, and set **Optimization Context** to **Development**.
- 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

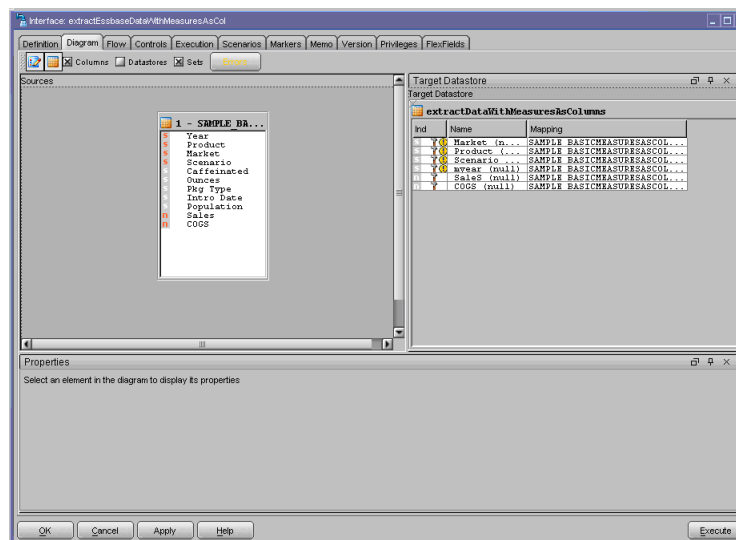
If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema” on page 7](#).

- 5 Select **Diagram**.
- 6 In the **Models** view, drag `extractDataWithMeasuresAsColumns` datastore from the **FLAT_FILES / ESSBASE_FLAT_FILES** model to the **Target Datastore** pane.
- 7 Drag the `Sample_BasicDataAsAColumn` source from the **Essbase/EssbaseSample** model to the **Sources** area.

A message that prompts you to use automatic mapping is displayed.

- 8 Click **Yes**.
- 9 Manually map any columns that were not mapped automatically.

This figure shows how the page should look:



- 10 On **Flow**, select the **SS_0**, and ensure that **LKM** is set to **LKM Hyperion Essbase Data to SQL**.
- 11 Select **Target**, and ensure that **IKM** is set to **IKM SQL to File**.
- 12 Set these LKM options.

Option	Values	Description
PRE_CALCULATION_SCRIPT	Blank (Default)	(Optional) Specify the calculation script that you want to run before extracting data from the Essbase cube.

Option	Values	Description
EXTRACTION_QUERY_TYPE	CalcScript	<p>Specify an extraction query type—report script, MDX query, or calculation script.</p> <p>Provide a valid extraction query, which fetches all the data to fill the output columns.</p> <p>The first record (first two records in case of calculation script) contains the meta information of the extracted data.</p>
EXTRACTION_QUERY_FILE	C:\Essbase_Samples\Calc_Scripts\ExtData.csc	Specify a fully qualified file name of the extraction query.
EXT_COL_DELIMITER	,	<p>Specify the column delimiter for the extraction query.</p> <p>If no value is specified for this option, then space (" ") is considered as column delimiter.</p>
EXTRACT_DATA_FILE_IN_CALC_SCRIPT	C:\Essbase_Samples\Data\Calcextract.csv	<p>This option is only applicable if the query type in the EXTRACTION_QUERY_TYPE option is specified as CalcScript.</p> <p>Specify a fully qualified file location where the data is extracted through the calculation script.</p>
PRE_EXTRACT_MAXL	Blank (Default)	Enable this option to execute a MAXL script before extracting data from the Essbase cube.
POST_EXTRACT_MAXL	Blank (Default)	Enable this option to execute a MAXL script after extracting data from the Essbase cube.
ABORT_ON_PRE_MAXL_ERROR	Yes	<p>This option is only applicable if the PRE_EXTRACT_MAXL option is enabled.</p> <p>If the ABORT_ON_PRE_MAXL_ERROR option is set to Yes, while executing pre-MAXL script, the load process is aborted on encountering any error.</p>
LOG_ENABLED	Yes	If this option is set to Yes, during the LKM process, logging is done to the file specified in the LOG_FILE_NAME option.
LOG_FILE_NAME	<?=java.lang.System.getProperty(java.io.tmpdir)?>/Extract_<%=snpRef.getFrom()%>.log (Default)	Specify a file name to log events of the LKM process.

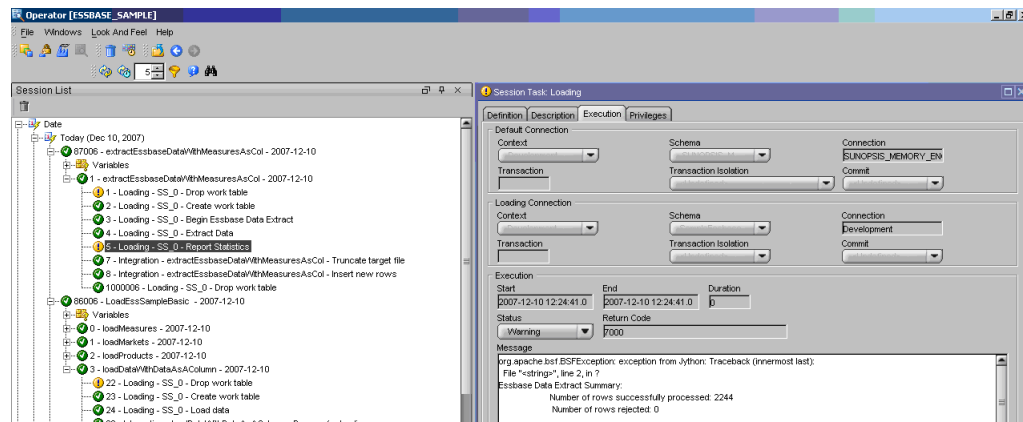
Option	Values	Description
MAXIMUM_ERRORS_ALLOWED	1 (Default)	Enable this option to set the maximum number of errors to be ignored before stopping extract.
LOG_ERRORS	Yes	If this option is set to Yes, during the LKM process, details of error records are logged to the file specified in the ERROR_LOG_FILENAME option.
ERROR_LOG_FILENAME	<?=java.lang.System.getProperty(java.io.tmpdir"?>/Extract_<%=snpRef.getFrom()"%>.err	Specify a file name to log error record details of the LKM process.
ERR_LOG_HEADER_ROW	No (Default)	If this option is set to Yes, then the header row containing the column names are logged to the error records file.
ERR_COL_DELIMITER	, (Default)	Specify the column delimiter to be used for the error records file.
ERR_ROW_DELIMITER	\r\n (Default)	Specify the row delimiter to be used for the error records file.
ERR_TEXT_DELIMITER	\\" (Default)	Specify the text delimiter to be used for the column data in the error records file. For example, if the text delimiter is set as ' \" ' (double quote), then all the columns in the error records file are delimited by double quotes.
DELETE_TEMPORARY_OBJECTS	No (Default)	Set this option to No, in order to retain temporary objects (tables, files, and scripts) after integration. This option is useful for debugging.

Note:

The `ExtData.csc` rules file is included in the `Calc_Scripts` directory within the `odiaess_93110_samples.zip`.

- 13 Click **Apply**.
- 14 Click **Execute** to run the `extractEssbaseDataWithMeasuresAsColumns` interface, and select **Development** under **Context**.
- 15 View the results of running the `extractEssbaseDataWithMeasuresAsColumns` interface in Oracle Data Integrator Operator.

This figure shows how the results should look in Operator:



- 16 Verify that the `extractDataWithDataAsColumn.csv` file contains the data extracted from the Essbase application—Sample and database—Basic.

Creating an Interface to Extract Metadata

You can create an interface for extracting metadata from the Essbase application—Sample and database—Basic.

- To create an interface for extracting metadata:
 - 1 Launch Designer, and expand the **Interfaces** node under the **EssbaseSample** project.
 - 2 Right-click, and select **Insert Interface**.
 - 3 Name the interface `extractEssbaseMeasures`, and set **Optimization Context** to **Development**.
 - 4 Select **Staging Area Different from Target**, and select a staging area that is appropriate to your environment.

Note:

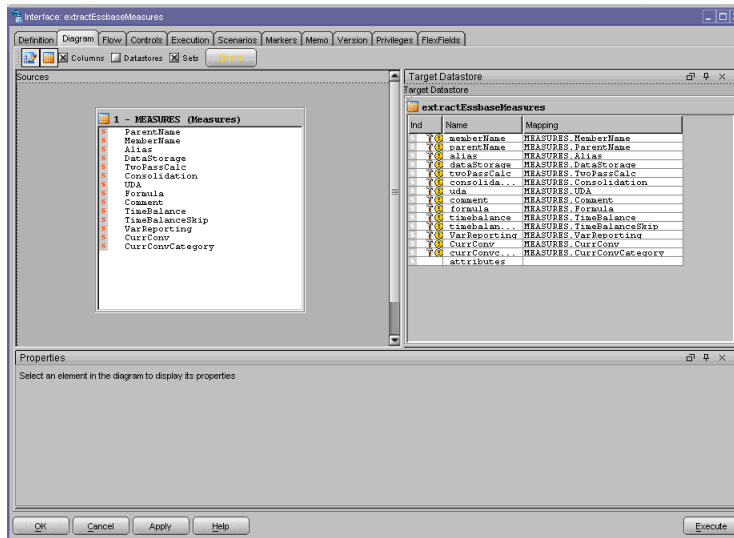
If no data server defined in your topology can be used as a staging area, use Sunopsis Memory Engine as the staging area, see [“Configuring the Sunopsis Engine Physical Schema”](#) on page 7.

- 5 Select **Diagram**.
- 6 In the **Models** view, drag `extractEssbaseMeasures` datastore from the **FLAT_FILES / ESSBASE_FLAT_FILES** model to the **Target Datastore** pane.
- 7 Drag the **Measures** source from the **Essbase/EssbaseSample** model to the **Sources** area.

A message that prompts you to use automatic mapping is displayed.

- 8 Click **Yes**.
- 9 Manually map any columns that were not mapped automatically.

This figure shows how the page should look:

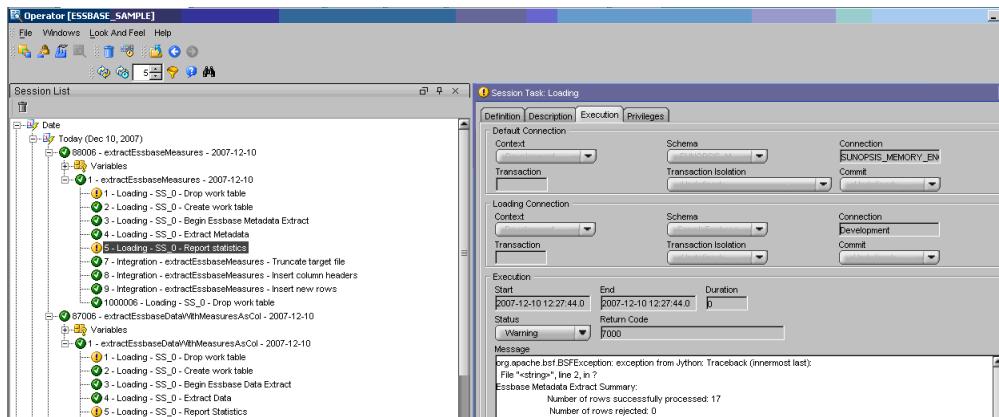


- 10 On the **Flow** , select the **SS_0**, and ensure that **LKM** is set to **LKM Hyperion Essbase (Metadata) to SQL**.
- 11 Select **Target**, and ensure that **IKM** is set to **IKM SQL to File Append**.
- 12 Set these IKM options.

Options	Value	Description
INSERT	Yes	Enable this option to automatically insert data into the Target Datastore of the interface.
TRUNCATE	Yes	If this option is set to Yes, the target datastore is truncated and/or the target file is created.
GENERATE_HEADER	Yes	If this option is set to Yes, header row is inserted into the target file that lists the names of the columns.

- 13 Click **Apply**.
- 14 Click **Execute** to run the extractEssbaseMeasures interface, and select **Development** under **Context**.
- 15 View the results of running the extractEssbaseMeasures interface in Oracle Data Integrator Operator.

This figure shows how the results should look in Operator:



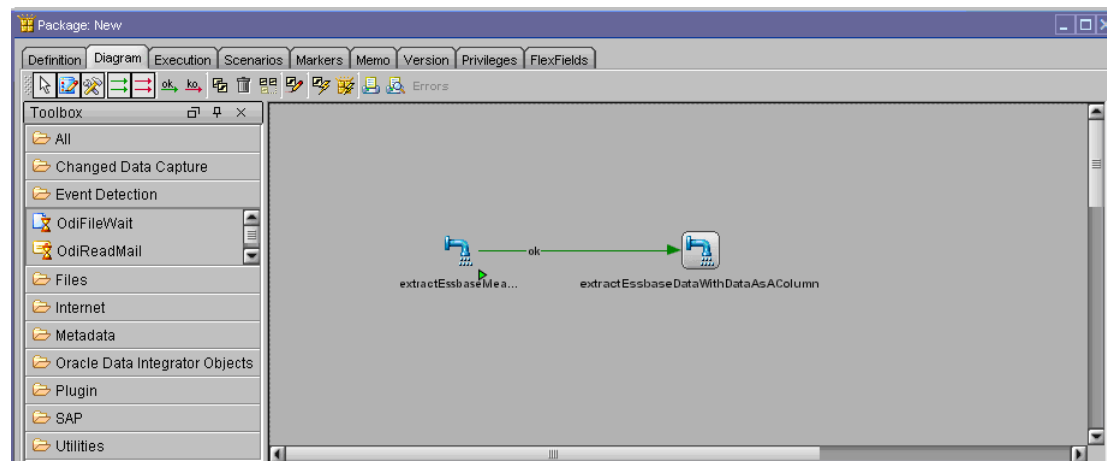
- 16 Verify that the `MeasuresExtract.csv` file contains the data extracted from the Essbase application—Sample and database—Basic.

Creating a Package to Extract Metadata and Data

You can chain interfaces into a package so that you can run them in a single process.

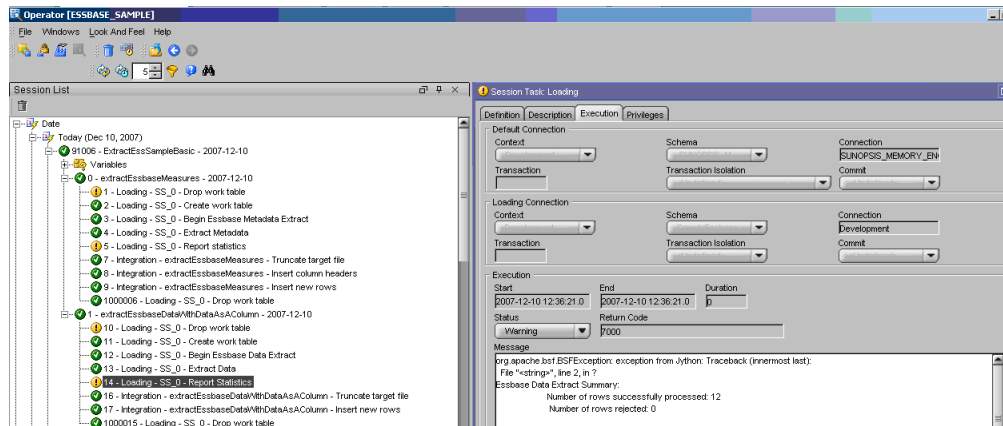
- To create a package for extracting metadata and data:
 - 1 Launch Designer.
 - 2 Right-click **Packages**, and select **Insert Package**.
 - 3 Enter a name for the package, such as `ExtractEssSampleBasic`.
 - 4 Select **Diagram**.
 - 5 Drag the **extractEssbaseMeasures** and **extractEssbaseDataWithDataAsAColumn** interfaces into the diagram area.
 - 6 Connect the interfaces in sequence, using the ok→ green arrows.
 - 7 Click **Apply**.

This figure shows how the page should look:



- 8 Click **Execute** to run the `ExtractEssSampleBasic` package, and select **Development** under **Context**.
- 9 View the results of running the `ExtractEssSampleBasic` package in Oracle Data Integrator Operator.

This figure shows how the results should look in Operator:



- 10 Verify that the `extractEssbaseDataWithDataAsAColumn.csv` and `MeasuresExtract.csv` files contain the data and members, respectively, that were extracted from the Oracle's Hyperion® Essbase® – System 9 application.

COPYRIGHT NOTICE

Oracle Data Integrator Adapter for Hyperion Essbase Getting Started, 9.3.1.1.0

Copyright © 2008, Oracle and/or its affiliates. All rights reserved.

Authors: Data Integration Management writing team

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.