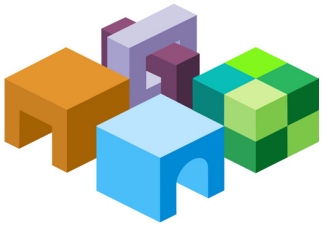


RELEASE 9.3.1.1

ORACLE DATA INTEGRATOR ADAPTER FOR HYPERION FINANCIAL MANAGEMENT GETTING STARTED

ORACLE | Hyperion

Contents In Brief

Sample Files for Practice

This document guides you through the use of the sample files that are included with Oracle® Data Integrator Adapter for Hyperion Financial Management. 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 consolidate data in Oracle's Hyperion® Financial Management – System 9 applications.

The sample files are delivered in the `odiafm_93110_samples.zip` file, which you can extract to any folder. The `odiafm_93110_samples.zip` file contains these folders:

- `data`, which contains these files:
  - `LoadAccounts.csv`
  - `LoadCurrencies.csv`
  - `LoadCustom1.csv`
  - `LoadCustom2.csv`
  - `LoadCustom3.csv`
  - `LoadCustom4.csv`
  - `LoadEntities.csv`
  - `LoadScenarios.csv`
  - `alldata.dat`
  - `extractData.dat`
- `work_repository`, which contains `financial_management_samples.zip`

Financial Management Sample Application Prerequisites

Using the sample files requires that you have a Financial Management application called Comma. If you do not have the Comma application, you must create it. For instructions on creating applications, see the *Hyperion Financial Management — System 9 Administrator's Guide*.

To create the Comma application, you require sample security, metadata, data, rules, and member list files. If you installed the Sample Applications component from the Typical or Custom installation process, the files are in the directory where you installed Financial Management. If you did not install the sample files, you can obtain them by reinstalling Financial Management and selecting the Sample Applications component during the installation.

When you create the Comma application, you can load sample security, metadata, data, rules, and member list files from the `Sample Apps/Comma` directory. Instructions for loading the sample files are included in the `Sample Apps/Comma/Documentation` directory.

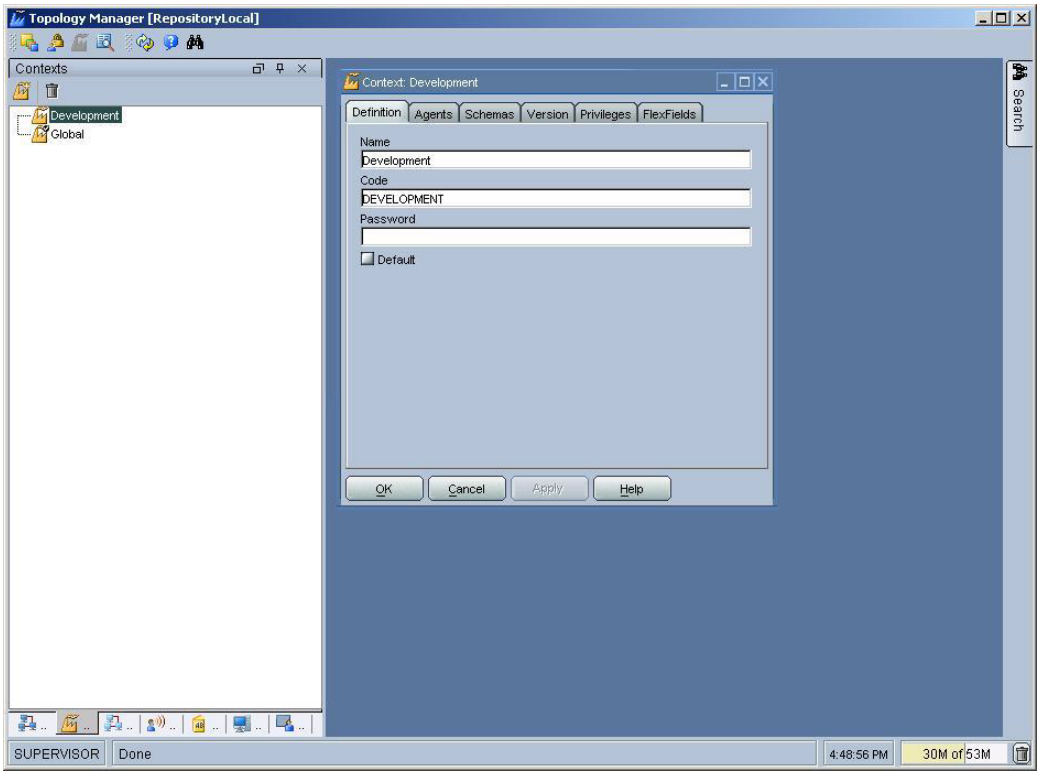
Setting Up an Environment

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 Hyperion Financial Management Data Server

Use Oracle Data Integrator to create a data server for the Hyperion Financial Management 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.

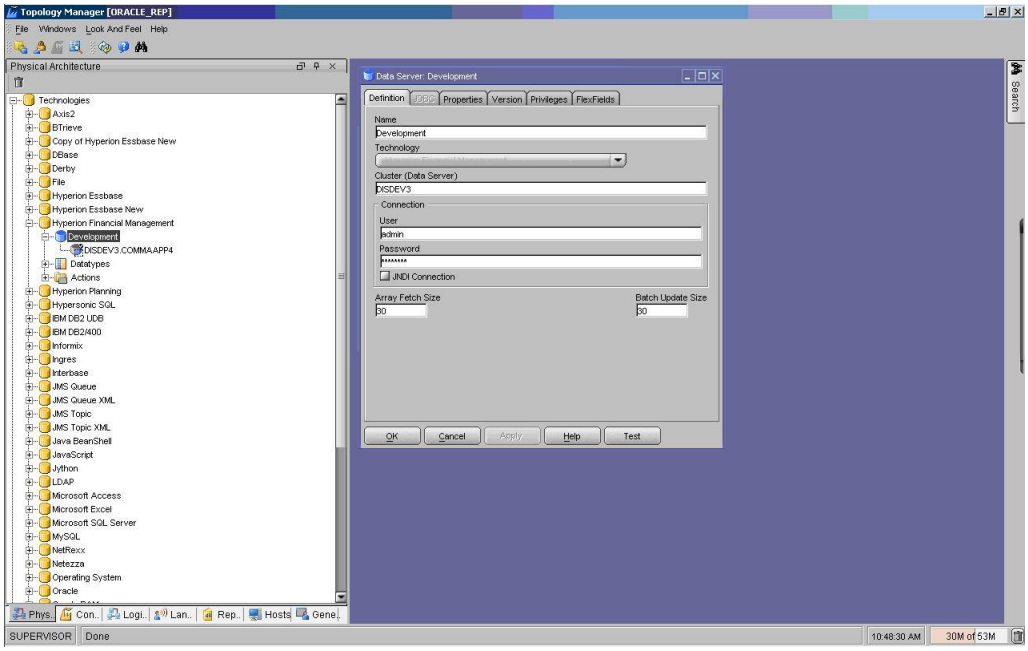
➤ To set up a Hyperion Financial Management data server:

1 **Using the Topology Manager, create a data server under the Hyperion Financial Management technology:**

Note:

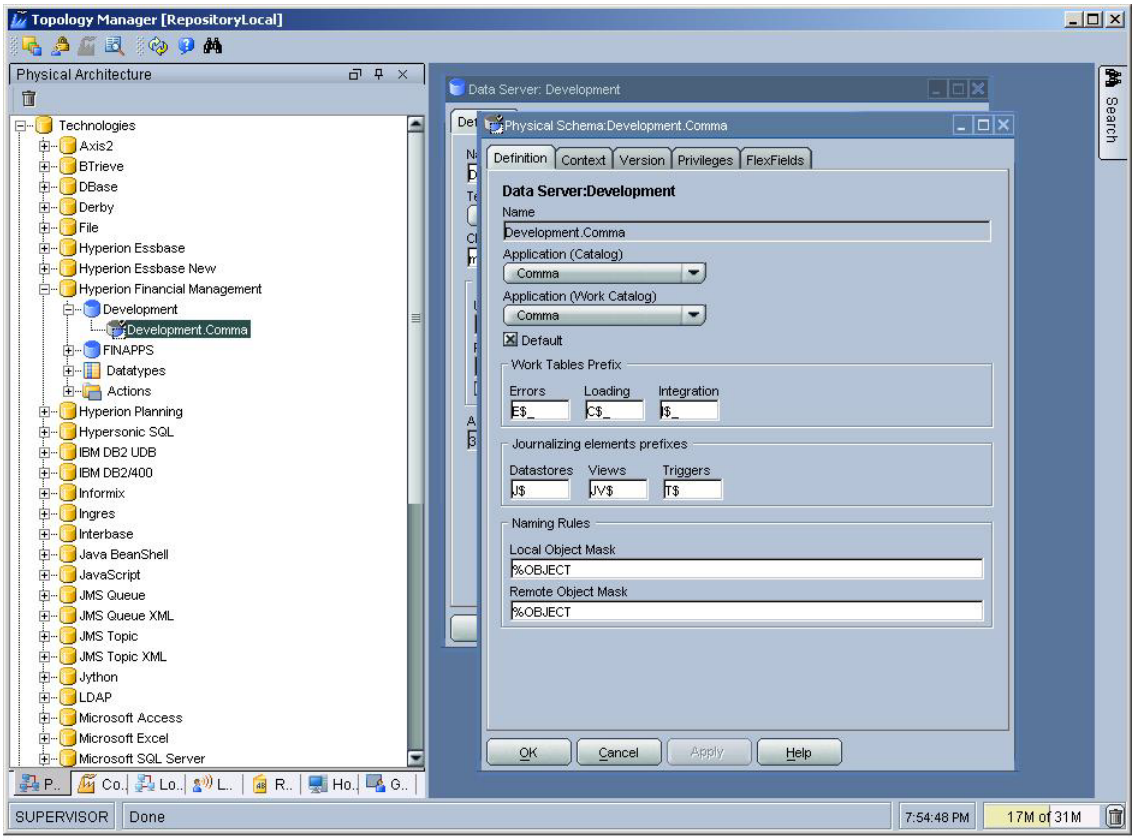
If the Hyperion Financial Management technology is not listed in the Topology Manager, you must import it. See the *Oracle Data Integrator User's Guide* for instructions.

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



3 Create the physical schema to point to your Financial Management application.

In this figure, the physical schema points to the Comma application:



4 **On the Context tab:**

a. Set the **Context** to Development.

b. Enter the logical schema name SampleHFMApp.

> **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.

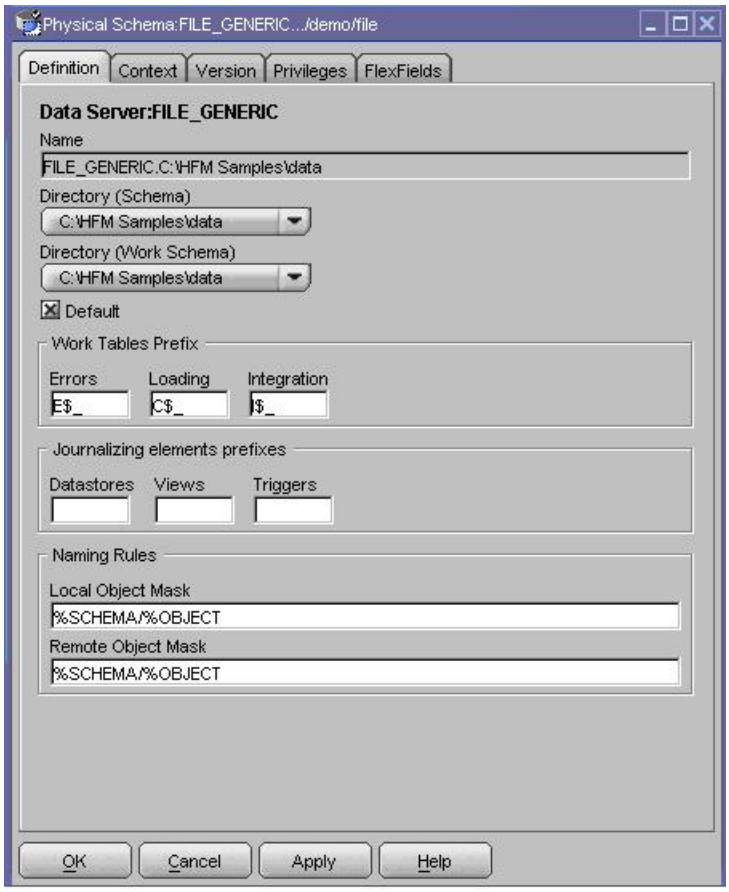
Note:

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 `odiafm_93110_samples.zip`), as shown:**

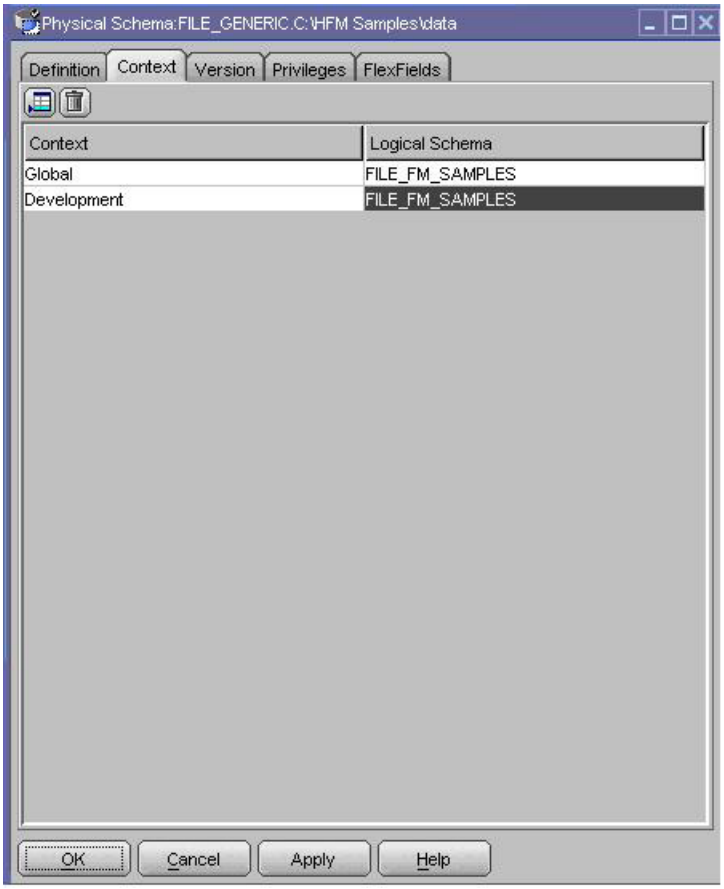


The data folder contains the samples source files.

3 **On the Context tab:**

a. Set the **Context** to Development.

b. For **Logical Schema**, enter FILE_FM_SAMPLES, as shown:



c. Click **OK**.

Setting Up a Work Repository

The `odifm_93110_samples.zip` that is delivered with Oracle Data Integrator Adapter for Hyperion Financial Management includes a work repository export file called `financial_management_samples.zip`. (For more information about `odifm_93110_samples.zip`, see "Sample Files for Practice" on page 2.)

The `financial_management_samples.zip` file contains the Oracle Data Integrator models, interfaces, packages, and KMs that are required for loading and extracting metadata and data in the sample Financial Management application. Use Oracle Data Integrator to create a work repository for your work with the Adapter for Hyperion Financial Management samples and import `financial_management_samples.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_FINANCIAL_MANAGEMENT_SAMPLE. See the *Oracle Data Integrator User's Guide* for instructions.
2. Launch Designer, and connect to the ODI_FINANCIAL_MANAGEMENT_SAMPLE work repository.
3. Select **File > Import > Work Repository**.
4. Select an import mode.

The INSERT_UPDATE mode is recommended.

5 **Navigate to the folder containing `financial_management_samples.zip`, and click OK.**

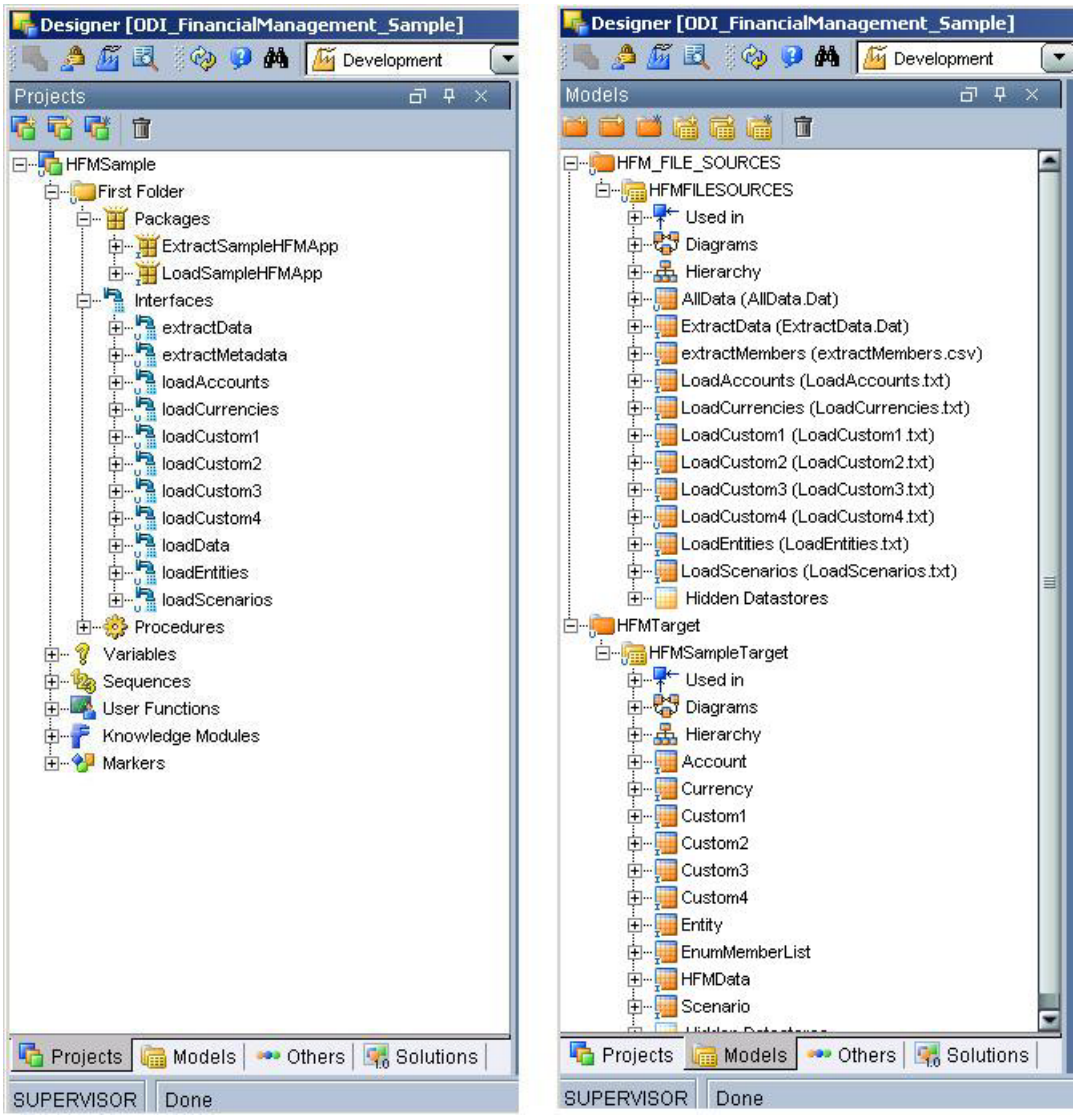
These interfaces are added to the work repository:

- loadAccounts
- loadCurrencies
- loadCustom1
- loadCustom2
- loadCustom3
- loadCustom4
- loadEntities
- loadScenarios
- loadData
- extractData
- extractMetadata

The import also adds two packages:

- LoadSampleHFMApp, which chains the interfaces for loading metadata and data
- ExtractSampleHFMApp, which chains the interfaces for extracting data and members

This figure shows how the Projects and Models trees look when the import succeeds:



Loading and Extracting Data Using Sample Interfaces

The work repository, which was set up as part of the environment, contains interfaces for loading and extracting Financial Management application data.

Loading Metadata into the Sample Application

Load theFinancial Management sample application with metadata following the procedures in the following topics.

Loading Account Dimension Metadata

The sample package includes an interface called loadAccounts, which loads metadata into the Account dimension.

Note:

For instructions on building this interface or others like it, see "Creating an Interface to Load Metadata" on page 15.

➤ To load metadata into the Account dimension:

1. Run the loadAccounts interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Account dimension from Financial Management Windows client.

Loading Currency Dimension Metadata

The sample package includes an interface called loadCurrencies, which loads metadata into the Currency dimension.

➤ To load metadata into the Currency dimension:

1. Run the loadCurrencies interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Currency dimension from Financial Management Windows client.

Loading Custom1 Dimension Metadata

The sample package includes an interface called loadCustom1, which loads metadata into the Custom1 dimension.

➤ To load metadata into the Custom1 dimension:

1. Run the loadCustom1 interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Custom1 dimension from Financial Management Windows client.

Loading Custom2 Dimension Metadata

The sample package includes an interface called loadCustom2, which loads metadata into the Custom2 dimension.

➤ To load metadata into the Custom2 dimension:

1. Run the loadCustom2 interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Custom2 dimension from Financial Management Windows client.

Loading Custom3 Dimension Metadata

The sample package includes an interface called loadCustom3, which loads metadata into the Custom3 dimension.

➤ To load metadata into the Custom3 dimension:

1. Run the loadCustom3 interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Custom3 dimension from Financial Management Windows client.

Loading Custom4 Dimension Metadata

The sample package includes an interface called loadCustom4, which loads metadata into the Custom4 dimension.

➤ To load metadata into the Custom4 dimension:

1. Run the loadCustom4 interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Custom4 dimension from Financial Management Windows client.

Loading Entity Dimension Metadata

The sample package includes an interface called loadEntities, which loads metadata into the Entity dimension.

➤ To load metadata into the Entity dimension:

1. Run the loadEntities interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Entity dimension from Financial Management Windows client.

Loading Scenario Dimension Metadata

The sample package includes an interface called loadScenarios, which loads metadata into the Scenario dimension.

➤ To load metadata into the Scenarios dimension:

1. Run the loadScenarios interface.
2. Check the Operator log to see whether the interface ran successfully.
3. Validate the Scenarios dimension from Financial Management Windows client.

Loading Data into the Sample Application

The sample package includes an interface called loadData, which loads data into the Financial Management application.

Note:

Before completing the following procedure, verify that the Financial Management application contains the required metadata. If it does not, load the metadata from Financial Management Windows client.

➤ To load data into a Financial Management application:

1. **Run the loadData interface.**
2. **Check the Operator log to see whether the interface ran successfully.**
3. **Validate the data load from Financial Management Windows client.**

Extracting Data from the Sample Application

The sample package includes an interface called extractData, which extracts data from the Financial Management application and writes the data to a file.

➤ To extract data from a Financial Management application:

1. **Run the extractData interface.**
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 `odiafm_93110_samples.zip`).**

Extracting Member Lists from the Sample Application

The sample package includes an interface called extractMetadata, which extracts member lists from the Financial Management application and writes them to a file.

➤ To extract member lists from a Financial Management application:

1. **Run the extractMetadata interface.**
2. **Check the Operator log to see whether the interface ran successfully.**
3. **Validate the extracted data in the file `extractMembers.csv` in the data directory (extracted from `odiafm_93110_samples.zip`).**

Creating Models

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

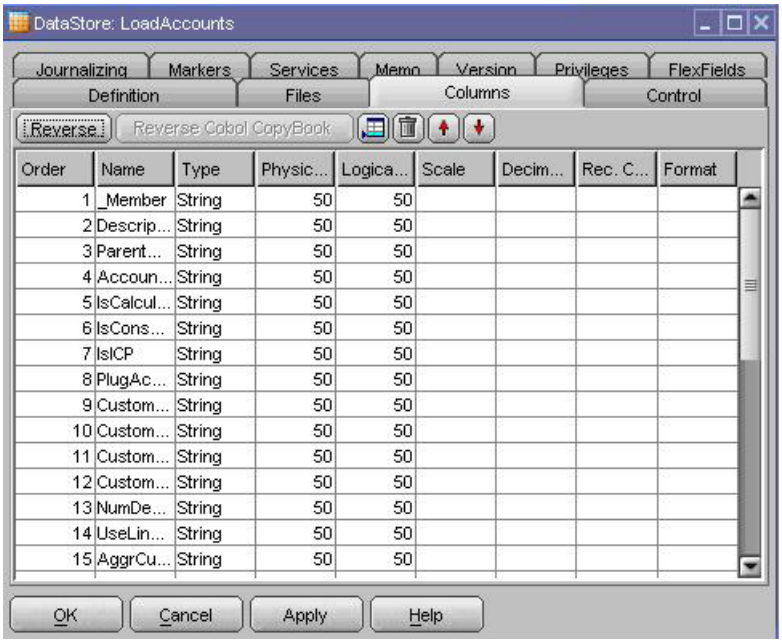
Creating and Reverse-Engineering the Sample Source Models

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

➤ To create and reverse-engineer the sample source models:

1. In the **Models** view, insert a new model folder called `HFM_File_Sources`.
2. Right-click the `HFM_File_Sources` model folder, and select **Insert Model**.
3. Name the model HFMFileSources, and set **Technology** to File and **Logical Schema** to FILE_FM_SAMPLES.
4. On the **Reverse** tab, set **Context** to Development, and click **OK**.
5. Right-click **HFMFileSources**, and select **Insert DataStore**.
6. Set **Name** to Account, Currency, Custom1, Custom2, Custom3, Custom4, Entity, Scenario, or DataLoad, 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 (Account, Currency, Custom1, Custom2, Custom3, Custom4, Entity, or Scenario).
8. Select the **Files** tab.
9. 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.
10. On the **Columns** tab, click **Reverse**.

This figure shows how the page should look:



11 Ensure that the fields corresponding to numeric fields in the Financial Management application are set as numeric, with the correct length and scale.

12 Repeat step 5 through step 11 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 Hyperion Financial Management target models.

➤ To reverse-engineer the sample target models:

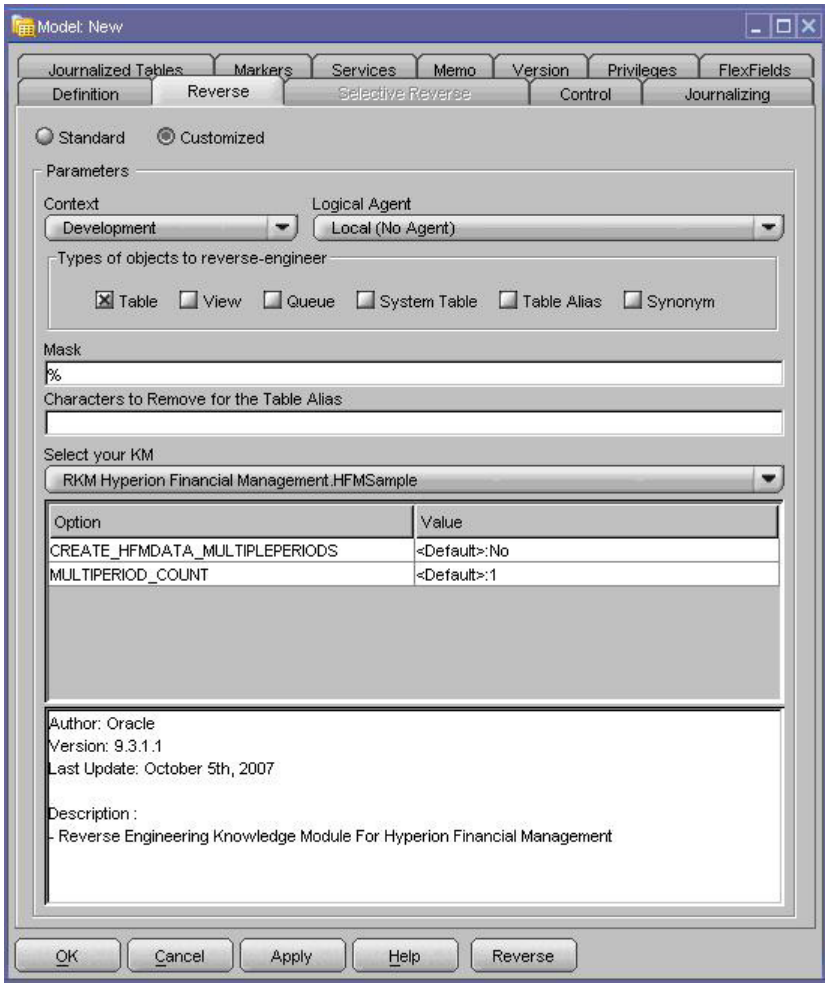
1 In the **Models** view, insert a new model folder called HFMTarget.

2 Right-click the **HFMTarget** model folder and select **Insert Model**.

3 Name the model HFMSampleTarget, set **Technology** to Hyperion Financial Management, and set **Logical Schema** to SampleHFMApp.

4 On the **Reverse** tab, select **Customized** (at the top of the page).

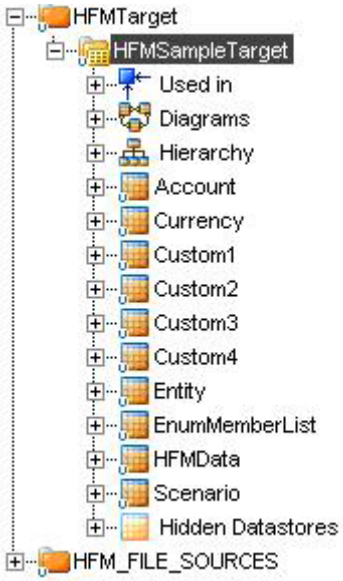
5 Set **Context** to Development, and select **RKM Hyperion Financial Management, Financial Management Sample**, as shown:



6 Set RKM options.

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

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



If the HFMSampleTarget 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 Account dimension into the sampleFinancial Management application. Using this interface as a model, you can create interfaces for loading the Currency, Custom1, Custom2, Custom3, Custom4, Entity, and Scenario 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 22.

➤ To create an interface for loading the Account dimension:

1 Launch Designer, and expand the **Interfaces** node under the **HFMSample** project.

2 Right-click and select **Insert Interface**.

3 Name the interface loadAccounts, and set **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.

5 Select the **Diagram** tab.

6 In the **Models** view, drag **Account DataStore** from the **HFMTarget/HFMSampleTarget** model to the **Target DataStore** pane.

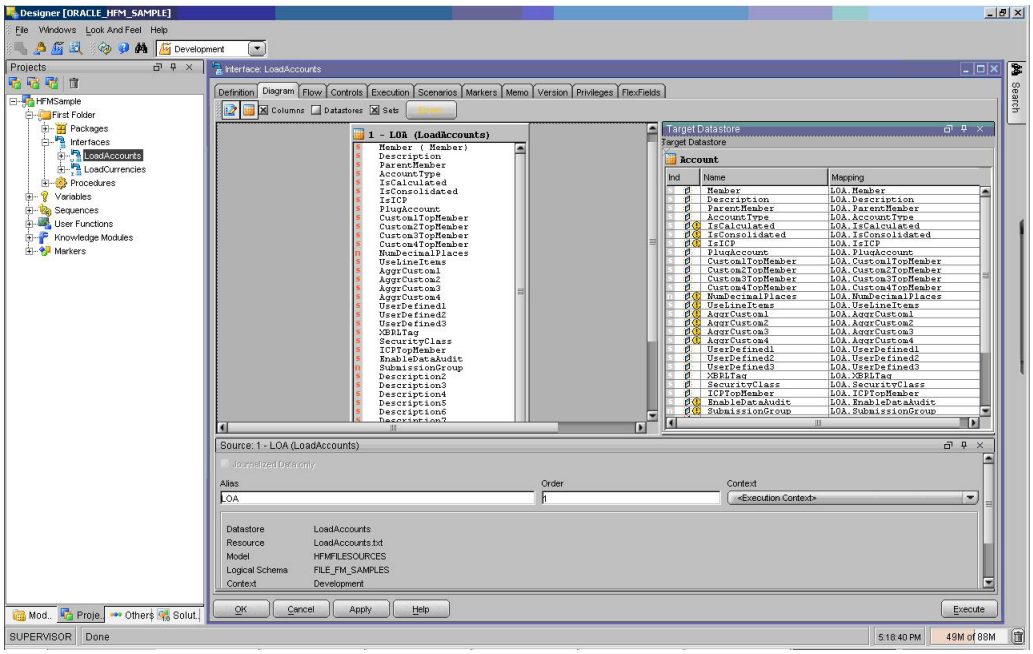
7 Drag the **Accounts** source from the **HFM_File_Sources/HFMFileSource** model to the **Sources** area.

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

8 Click **Yes**.

9 Manually map any columns that were not mapped automatically.

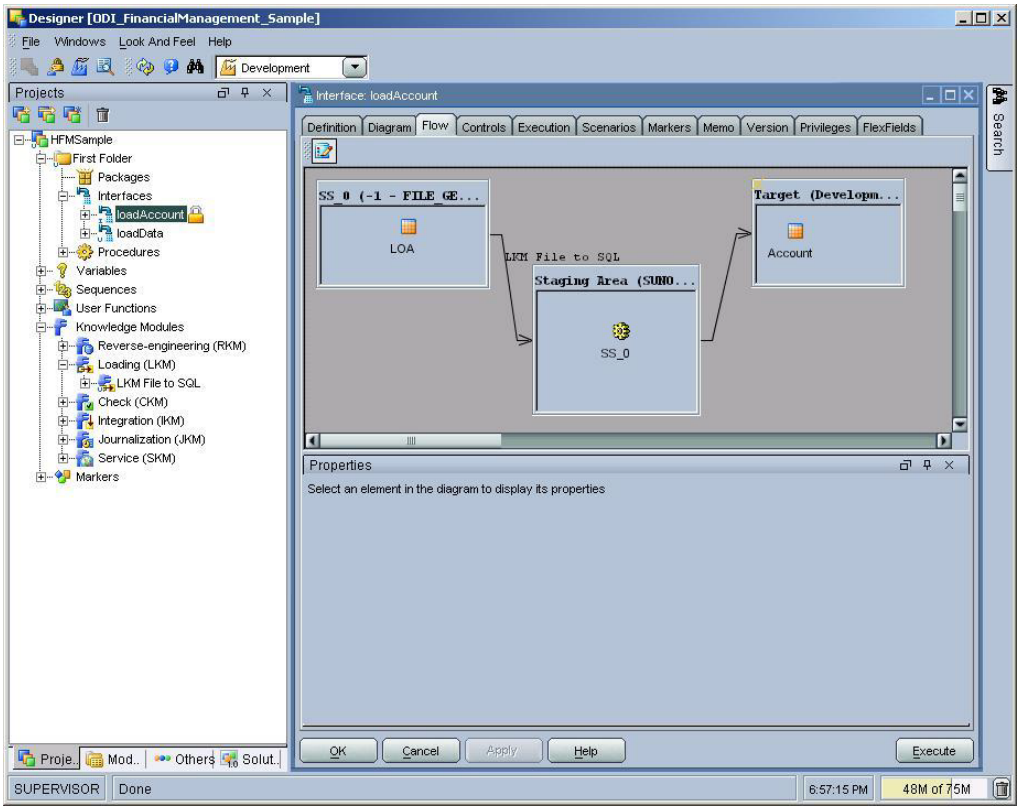
This figure shows how the page should look:



10 On the **Flow** tab, select the SS_0, and ensure that **LKM** is set to LKM File to SQL.

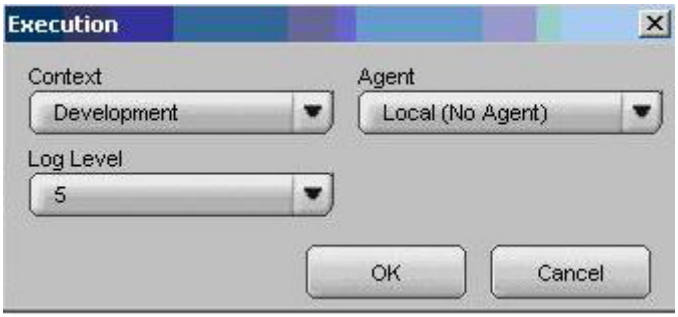
11 Select **Target**, and ensure that **IKM** is set to IKM SQL to Hyperion Financial Management Dimension.

12 Set **IKM** options.



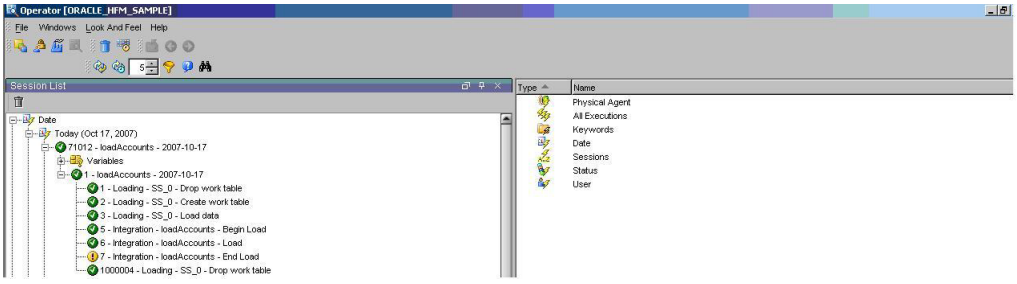
13 Click **Apply**.

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



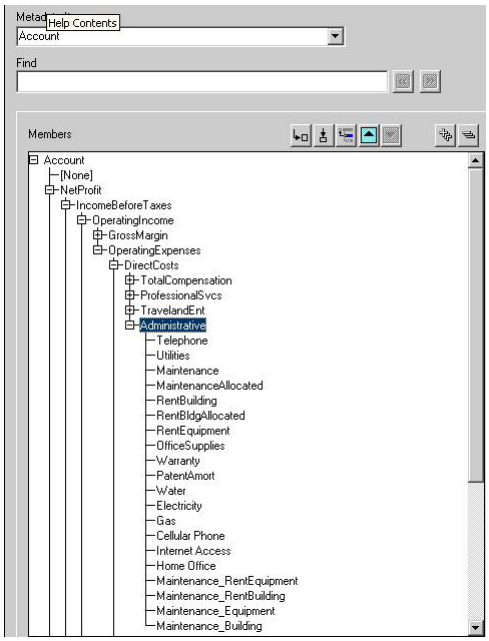
15 View the results of running the interface in Oracle Data Integrator Operator.

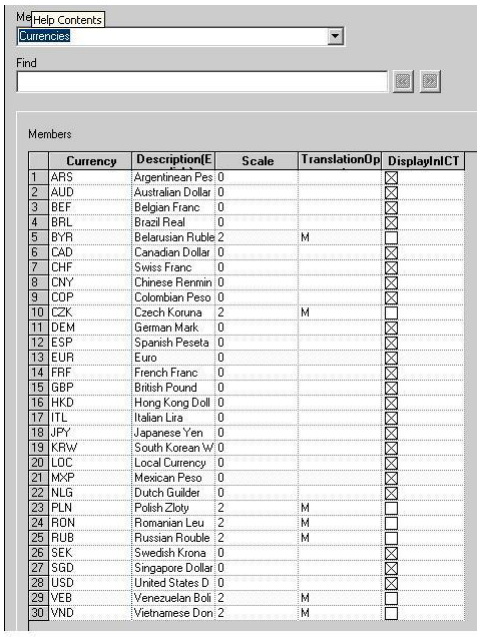
This figure shows how the results should look:

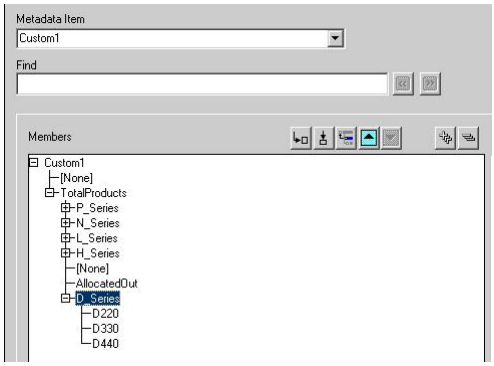


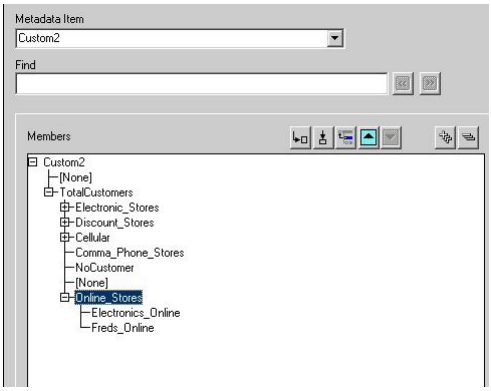
16 Validate the Account dimension from Financial Management Windows client.

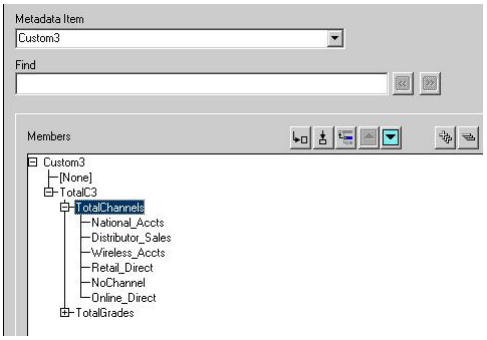
These figures show the hierarchies created in Financial Management:

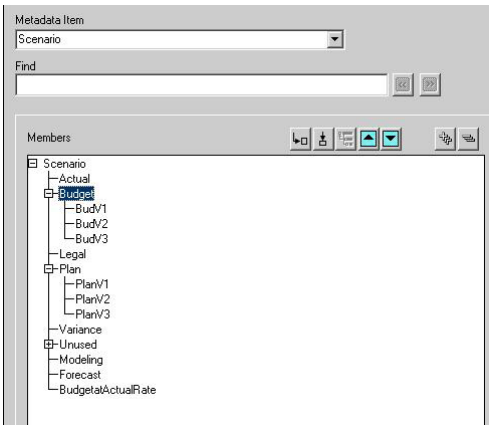

Help Contents
Account
Find
Members
Account
[None]
NetProfit
IncomeBeforeTaxes
OperatingIncome
GrossMargin
OperatingExpenses
DirectCosts
TotalCompensation
ProfessionalSvcs
TravelandEnt
Administrative
Telephone
Utilities
Maintenance
MaintenanceAllocated
RentBuilding
RentBldgAllocated
RentEquipment
OfficeSupplies
Warranty
PatentAmort
Water
Electricity
Gas
Cellular Phone
Internet Access
Home Office
Maintenance_RentEquipment
Maintenance_RentBuilding
Maintenance_Equipment
Maintenance_Building


Help Contents
Currencies
Find
Members
Currency
Description(E
Scale
TranslationOp
DisplayInICT
1 ARS Argentinean Pes 0
2 AUD Australian Dollar 0
3 BEF Belgian Franc 0
4 BRL Brazil Real 0
5 BYR Belarusian Ruble 2 M
6 CAD Canadian Dollar 0
7 CHF Swiss Franc 0
8 CNY Chinese Renmin 0
9 COP Colombian Peso 0
10 CZK Czech Koruna 2 M
11 DEM German Mark 0
12 ESP Spanish Peseta 0
13 EUR Euro 0
14 FRF French Franc 0
15 GBP British Pound 0
16 HKD Hong Kong Doll 0
17 ITL Italian Lira 0
18 JPY Japanese Yen 0
19 KRW South Korean W 0
20 LOC Local Currency 0
21 MXP Mexican Peso 0
22 NLG Dutch Guilder 0
23 PLN Polish Zloty 2 M
24 RON Romanian Leu 2 M
25 RUB Russian Rouble 2 M
26 SEK Swedish Krona 0
27 SGD Singapore Dollar 0
28 USD United States D 0
29 VEB Venezuelan Boli 2 M
30 VND Vietnamese Don 2 M


Metadata Item
Custom1
Find
Members
Custom1
[None]
TotalProducts
P_Series
N_Series
L_Series
H_Series
[None]
AllocatedOut
D_Series
D220
D330
D440







Creating an Interface to Load and Consolidate Data

➤ To create an interface for loading and consolidating data:

1. Launch Designer, and expand the Interfaces node under the **HFMSample** project.
2. Right-click, and select **Insert Interface**.
3. Name the interface loadData, and set **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.

5 Select the **Diagram** tab.

6 In the **Models** view, drag **HFMData DataStore** from the **HFMTarget/HFMSampleTarget** model to the **Target DataStore** pane.

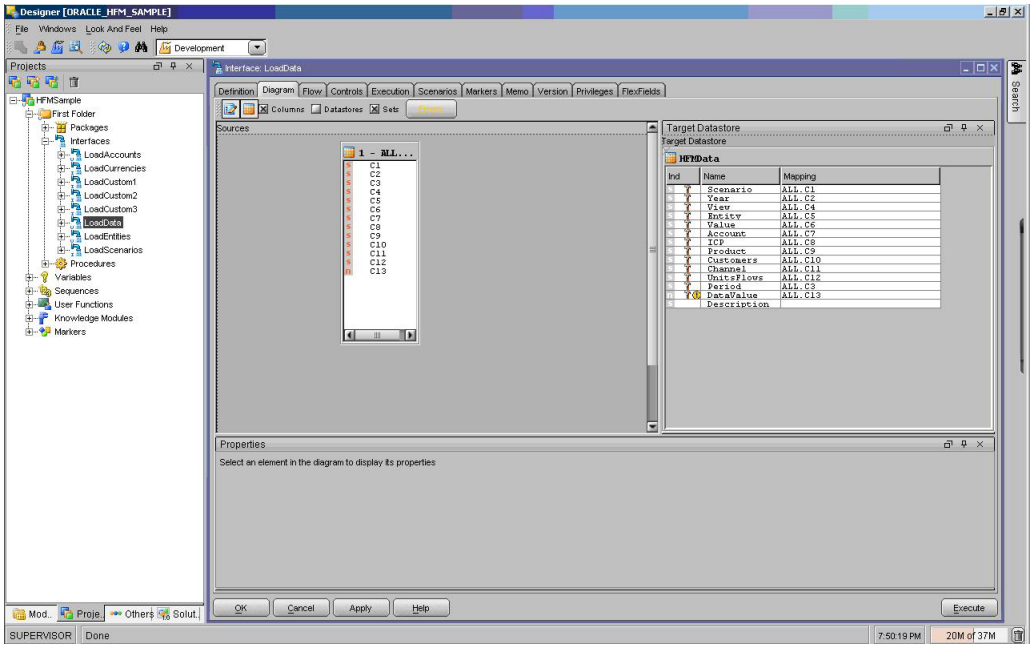
7 Drag the **alldata** source from the **HFM_File_Sources/HFMFileSource** model to the **Sources** area.

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:

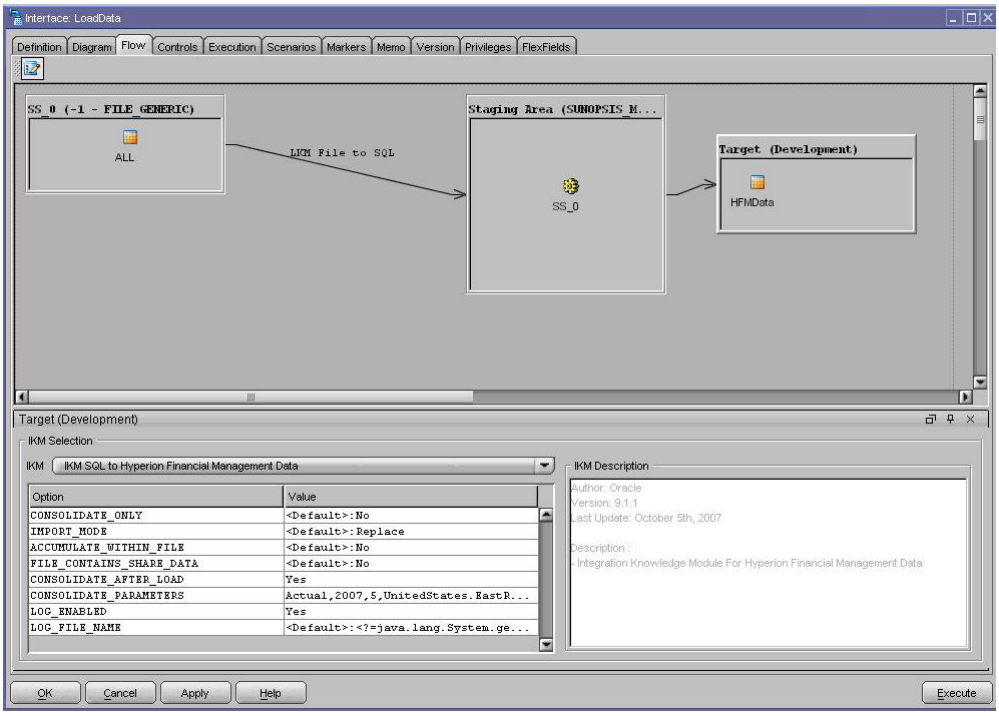


10 On the **Flow** tab, select the SS_0, and ensure that **LKM** is set to LKM File to SQL.

11 Select **Target**, and ensure that **IKM** is set to IKM SQL to Hyperion Financial Management Data.

12 Set these IKM options:

- CONSOLIDATE_AFTER_LOAD=YES
- CONSOLIDATE_PARAMETERS='ACTUAL,2007,5,United States.EastRegion,A'

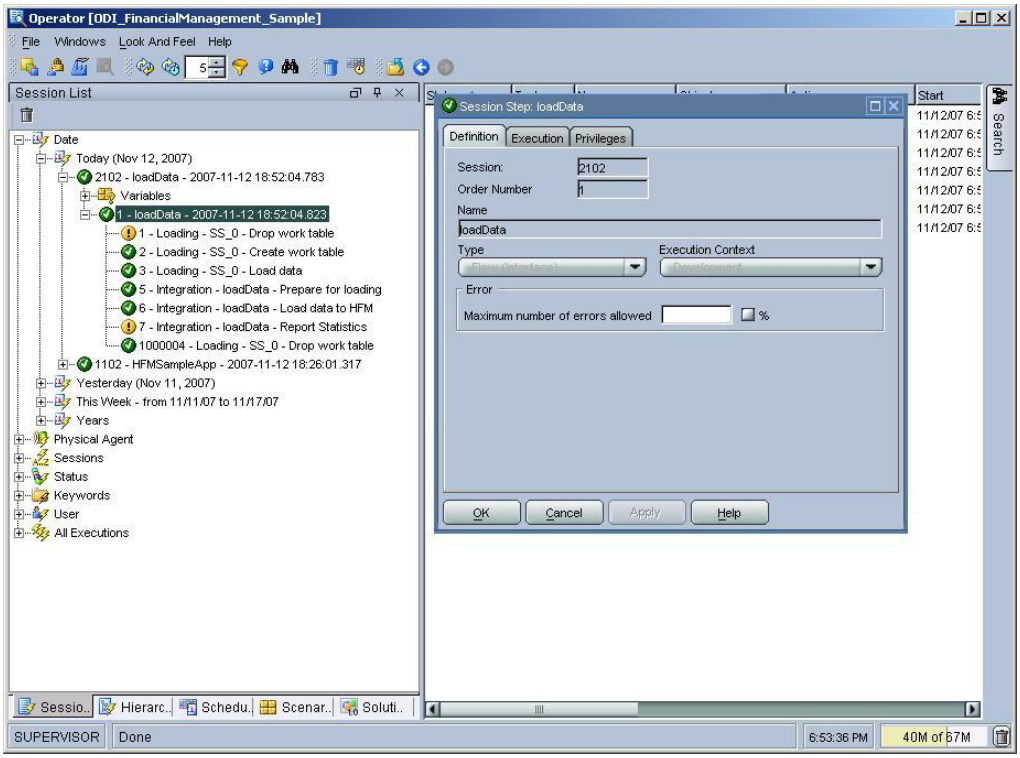


13 Click **Apply**.

14 Click **Execute** to run the loadData 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:



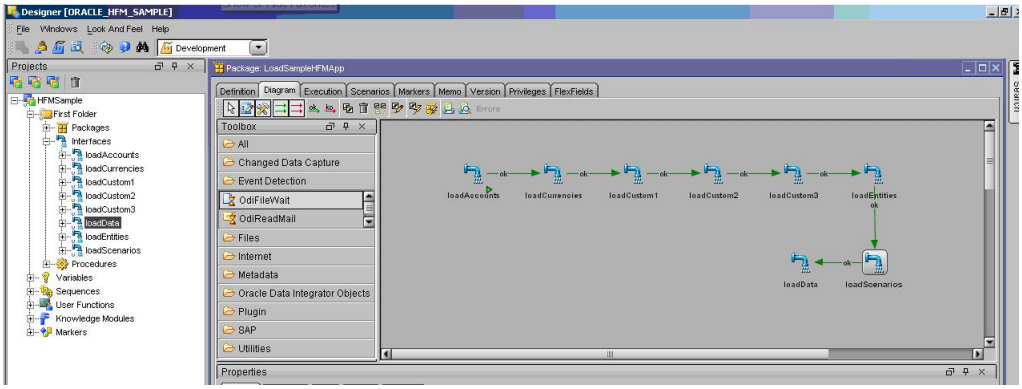
Creating a Package to Load Metadata and Data

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

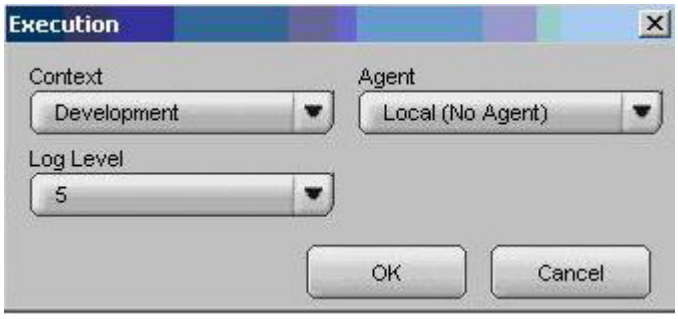
➤ To create a package for loading metadata and data:

1. Launch Designer.
2. Right-click **Packages**, and select **Insert Package**.
3. Name the package LoadSampleHFMApp (or any other name).
4. Select the **Diagram** tab.
5. Drag the **loadAccounts**, **loadCurrencies**, **loadCustom1**, **loadCustom2**, **loadCustom3**, **loadEntities**, **loadScenarios**, and **loadData** 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:

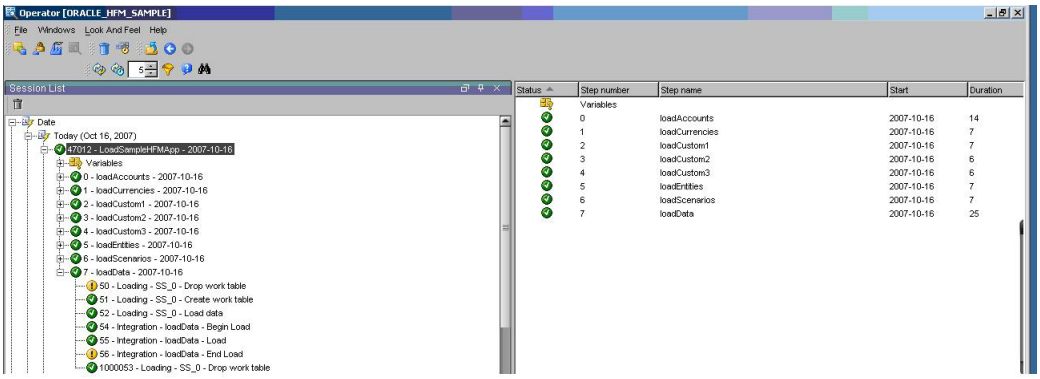


8. Click **Execute** to run LoadSampleHFMApp Package, and select Development as the context.



9. View the results of running the LoadSampleHFMApp package in Oracle Data Integrator Operator.

   This figure shows how the results should look in Operator:



Creating an Interface to Extract Data

➤ To create an interface for extracting data:

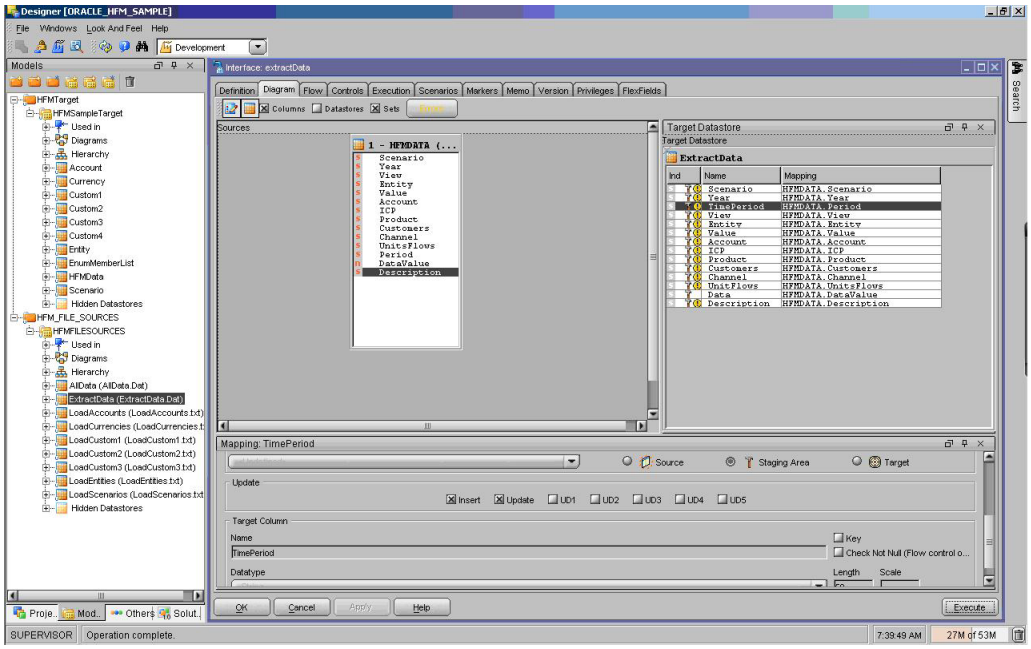
1. Launch Designer, and expand the **Interfaces** node under the **HFMSample** project.
2. Right-click, and select **Insert Interface**.
3. Name the interface extractData, and set **Context** to Development.
4. Select **Staging Area Different from Target**, and select a staging area that is appropriate 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.
5. Select the **Diagram** tab.
6. In the **Models** view, drag **extractData DataStore** from the **HFM_FILE_SOURCES/HFMFILESOURCES** model to the **Target DataStore** pane.
7. Drag the HFMData source from the HFMTarget/HFMSampleTarget 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** tab, select the SS_0, and ensure that **LKM** is set to LKM Hyperion Financial Management Data to SQL.

11 Select **Target**, and ensure that **IKM** is set to IKM SQL to File.

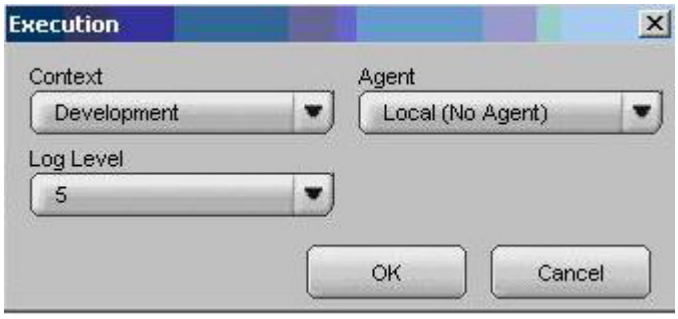
12 Set the LKM option SCENARIO_FILTER to Actual.

Note:

You can set other LKM options to further limit what data is extracted.

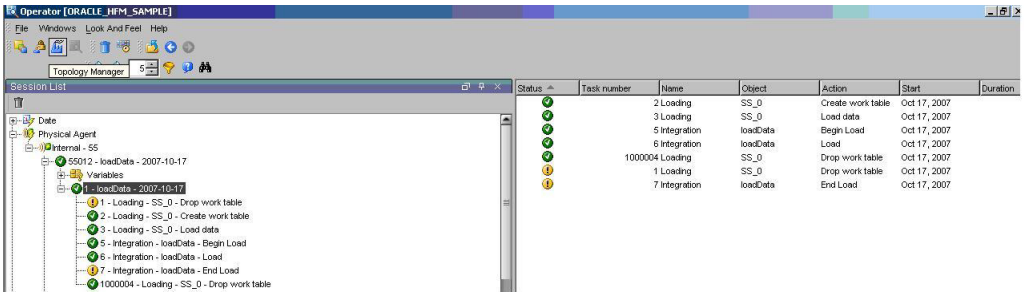
13 Click **Apply**.

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



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

This figure shows how the results should look in Operator:



16 **Verify that the `extractData.dat` file contains the data extracted from the Financial Management application.**

Creating an Interface to Extract Member Lists

You can create an interface for extracting member lists from the sample Financial Management application.

➤ To create an interface for extracting member lists:

1 **Launch Designer, and expand the Interfaces node under the HFMSample project.**

2 **Right-click, and select Insert Interface.**

3 **Name the interface extractMetadata, and set 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.

5 **Select the Diagram tab.**

6 **In the Models view, drag extractMembers DataStore from the HFM_FILE_SOURCES/ HFMFILESOURCES model to the Target DataStore pane.**

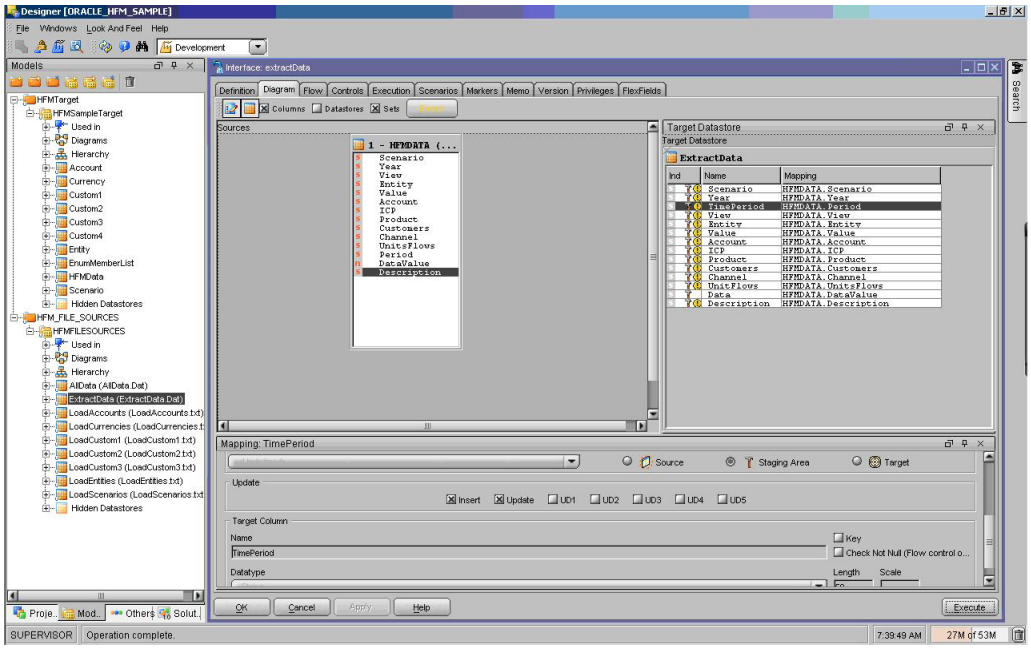
7 **Drag the EnumMemberList source from the HFMTarget/HFMSampleTarget model to the Sources area.**

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** tab, select the SS_0, and ensure that **LKM** is set to LKM Hyperion Financial Management Members to SQL.

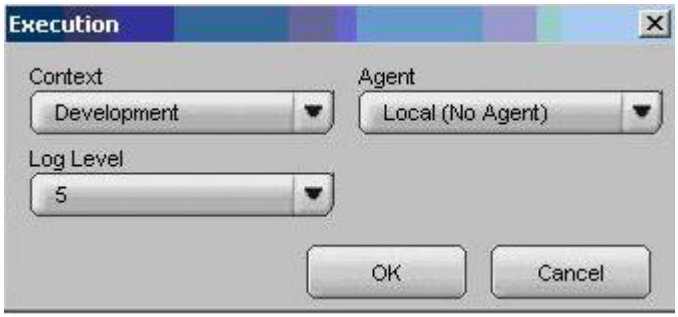
11 Select **Target**, and ensure that **IKM** is set to IKM SQL to File.

12 Set these IKM options:

- MEMBER_LIST_NAME=[Base]
- DIMENSION_NAME=Account

13 Click **Apply**.

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



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

16 Verify that the `extractMembers.csv` file contains the dimension members extracted from the Financial Management application.

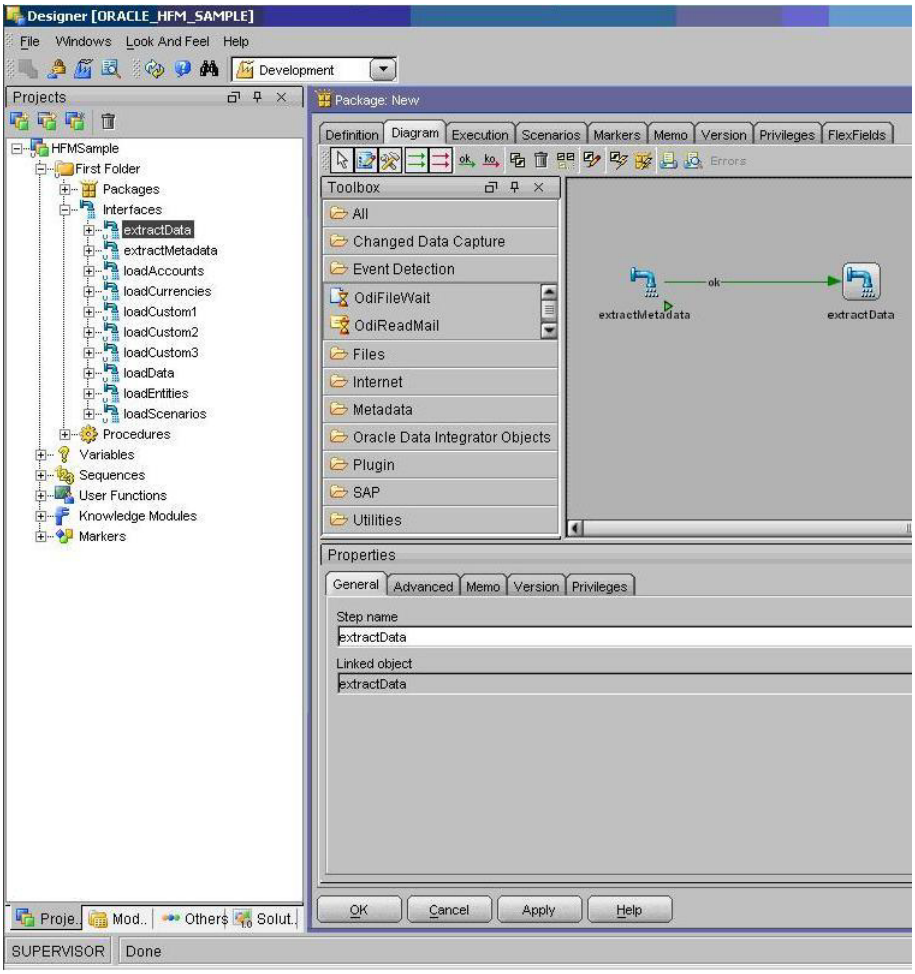
Creating a Package to Extract Metadata and Data

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

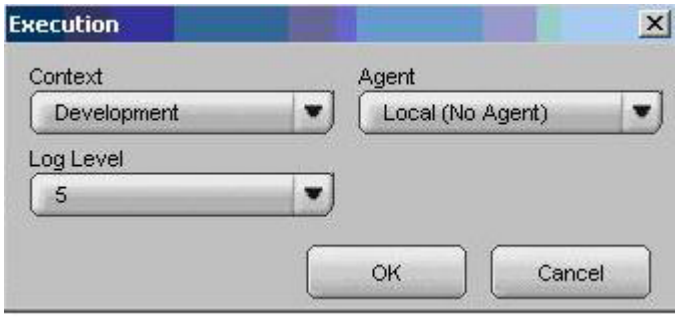
➤ To create a package for extracting metadata and data:

1. Launch Designer.
2. Right-click **Packages**, and select **Insert Package**.
3. Enter a name for the package, such as ExtractSampleHFMApp.
4. Select the **Diagram** tab.
5. Drag the **extractMetadata** and **extractData** 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:

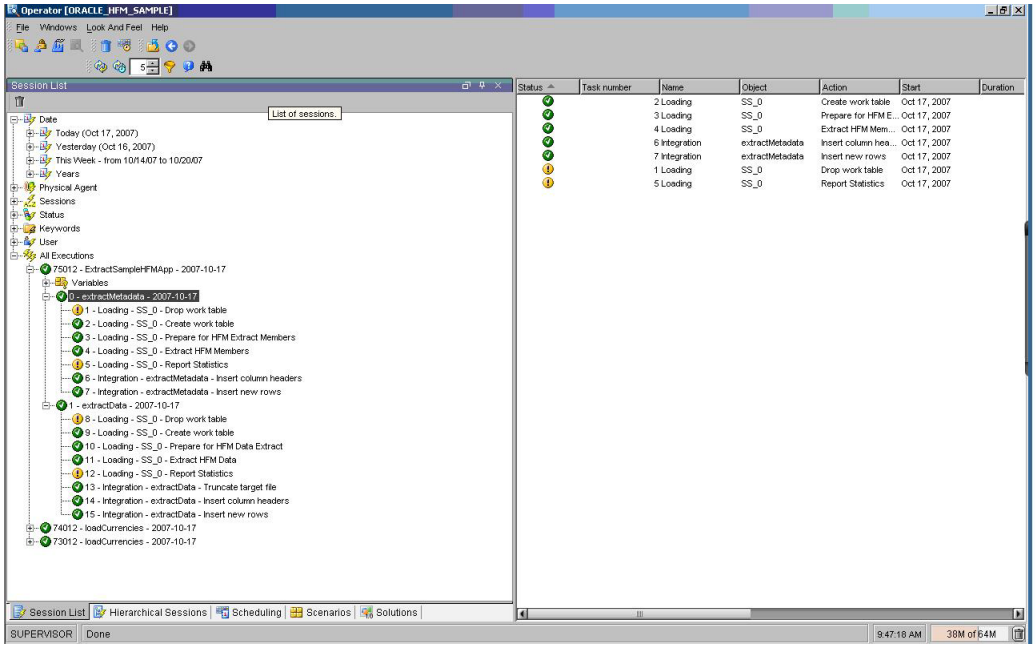


8. Click **Execute** to run the ExtractSampleHFMApp package, and select Development under **Context**.



9 **View the results of running the ExtractSampleHFMApp package in Oracle Data Integrator Operator.**

This figure shows how the results should look in Operator:



10 **Verify that the `extractData.dat` and `extractMembers.csv` files contain the data and members, respectively, that were extracted from the Oracle's Hyperion® Financial Management – System 9 application.**

COPYRIGHT NOTICE

Oracle Data Integrator Adapter for Hyperion Financial Management Getting Started, 9.3.1.1

Copyright © 2007, Oracle and/or its affiliates. All rights reserved.

Authors: Cheryl Morrison

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.