# Migration Guide

## Contents In Brief

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About This Document</td>
<td>2</td>
</tr>
<tr>
<td>Overview to the FDM Migration</td>
<td>2</td>
</tr>
<tr>
<td>How Migration Works</td>
<td>2</td>
</tr>
<tr>
<td>Supported FDM Versions for Migration</td>
<td>5</td>
</tr>
<tr>
<td>Additional Migration Details</td>
<td>5</td>
</tr>
<tr>
<td>FDM Artifacts and their Equivalents in FDMEE</td>
<td>5</td>
</tr>
<tr>
<td>Installation Steps</td>
<td>7</td>
</tr>
<tr>
<td>Executing the Migration</td>
<td>12</td>
</tr>
<tr>
<td>Drill Region Post Migration Steps</td>
<td>24</td>
</tr>
<tr>
<td>Check Rules Post Migration Steps</td>
<td>26</td>
</tr>
<tr>
<td>Additional Post Migration Steps</td>
<td>29</td>
</tr>
<tr>
<td>Testing and Troubleshooting Tips</td>
<td>30</td>
</tr>
</tbody>
</table>
About This Document


Overview to the FDM Migration

In EPM Release 11.1.2.4, FDM is no longer available. All customers who wish to continue using the key FDM functionality can migrate to FDMEE.

FDMEE is offered as the full-featured successor to Oracle® Hyperion Financial Data Quality Management ERP Integration Adapter for Oracle Applications (ERP Integrator) and Oracle Hyperion Financial Data Quality Management (FDM). FDMEE offers combined features of both products. It captures key functionality from FDM, and preserves the ERP Integrator framework that enables users to load metadata and data, and to drill-through and write-back.

Due to significant differences between FDM and FDMEE, no automated upgrade process is provided.

Notable aspects of this migration:

- FDM supports multiple applications with separate schemas, whereas FDMEE does not support application concepts. All data is stored in a single schema.
- A large volume of data is stored in the staging table and all data may not have to be migrated to the new environment. You can selectively migrate required data and omit very old data.
- FDM supports fixed, delimited, script, and adapter import format types. FDMEE imports only fixed and delimited import format types.
- Migration from FDM releases 11.1.1.x and 11.1.2.x to FDMEE 11.1.2.4 is supported.
- The ERPI content is upgraded to FDMEE during an in place upgrade, and the FDM content can also be migrated. After migration, duplicate artifacts from both ERPI and FDM may occur. It is recommended that you delete the duplicated artifacts from ERPI. If you have an integration of ERPI and FDM in releases 11.1.1.3 or 11.1.1.4, then this combination is not supported for migration, only 11.1.2.x if you use both ERPI and FDM together.
- FDM export uses dimension alias names when writing to the data file for Oracle Hyperion Financial Management applications. For this reason, alias names should be the same as defined in Financial Management.

How Migration Works

The migration is implemented as a Extract Transform Load (ETL) process from one or more source FDM schema to a single target FDMEE schema. The process is implemented using Oracle Database Integrator (ODI). The migration process can be executed using ODI installed as part of the 11.1.2.4 FDMEE installation. The process is executed and monitored using ODI Studio so make sure you have identified the server to install ODI Studio.

There are two ODI Scenarios: one for the setup data and the other for the historical data.
You run the setup data process to migrate the setup information like the application registration, import format, location, and so on to FDMEE. Since multiple FDM applications are consolidated into a single FDMEE schema to avoid duplication, you can prefix the data from each application with a prefix. If you follow a naming convention and data does not overlap, then a prefix may not be required.

You run the data process to migrate historical data. You may have several years of historical data in your existing FDM applications. You may not need to migrate all the data to FDMEE environment. You can decide how much data should be migrated to the new system and run the migration process only for required range periods.

Here is what you can expect when you migrate from FDM application to FDMEE.

1. The FDM migration does not delete or modify any objects within FDMEE.

2. The following FDM application objects are migrated:
   - Target Application
   - Global Categories
   - Application Categories
   - Global Periods
   - Application Periods
   - Logic groups
   - Check Entity Group
   - Check Rule Group
   - Import Format
   - Locations
   - Data Mappings
   - Data along with its workflow status, mappings, and drill regions

   If you use any other objects, they are not migrated automatically. You need to create or move them manually. If you use FDM along with ERPI, the application selected for migration may already exist in FDMEE. In this case, the FDMEE application is not created or modified.

   If a FDM location does not have an import format assigned (for example, you performed multi-loads only), the location and its data are not migrated.

3. Unlike FDM, FDMEE needs a data rule to store data. To meet this requirement, during the data migration, new data rules are created by a unique combination of location and category associated with the application.

   Dummy rules are only created for single period load irrespective of where the data in FDM was imported - from a single period or multi-period files. For any locations requiring multi-period loads, recreate import formats and data load rules as required in FDMEE.

4. FDM supports file-based, script-based, and ERPI adapter-based integrations.

   The following describes how these integrations are migrated:
- File-based—FDMEE creates a source system named “file” and assigns the source system type of “file” automatically. When file-based application objects are migrated, they are associated with this source system.

- Script-based—During migration, FDMEE creates a new source system named "FDMC" and assigns the source system type “Others.” FDMEE also creates a source adapter named “Open Interface Adapter” and assigns the source system type “Others” automatically. When script-based application objects are migrated, they are associated with this source system and source adapter.

Script based import formats are copied from FDM to the open interface adapter based import formats in FDMEE. You must rewrite the script that was used in FDM to pull data from the source in Jython as an `befImport` event. You can also add it to the open interface adapter scenario in ODI to indicate where to pull the data, and how to load to the open interface table. See the *Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator’s Guide* to complete this setup.

All import data pump scripts (Visual Basic scripts) are not migrated. The script expressions are not removed during migration. In this case, rewrite scripts in Jython, and then switch expressions to point to new Jython scripts.

- ERPI adapter-based—ERPI adapter-based integrations refer to the adapter import format used in FDM to integrate with EPRI (the predecessor of FDMEE) to run rules and bring data back in FDM.

To extract data from EBS/PSFT source systems, you create a source system, import formats, locations, and data rules in ERPI. Next, in FDM, you create import formats of the adapter type, and then associate them with newly created locations. These locations, in turn are associated with ERPI data rules.

For all FDM locations that used the ERPI adapter, new import formats are created in FDMEE to match the original segment mappings that were used in ERPI for the rule.

5. FDMEE supports six types of delimiters for source files:

- comma—`
- pipe—`\|`
- exclamation—`!
- colon—`:`
- semi-colon—`;`
- tab

FDM supports more delimiters than FDMEE. When import formats are migrated, any delimiter, other than one of the above six delimiters, is converted into a comma (,). If needed, you can modify a delimiter after the migration.

6. FDMEE does not support Visual Basic (VB) and Excel-based expressions in import formats. It also does not support VB-based mapping scripts. If needed, you can modify and use Jython-based expressions and mapping scripts (as applicable) after the migration.
Supported FDM Versions for Migration

The following are the supported releases for the migration:

- Source Version—FDM: Releases 11.1.1.x and 11.1.2.x are supported. ERPI-FDM releases 11.1.2.x are supported. ERPI-FDM setups in release 11.1.1.3 & 11.1.1.4 are not migrated.
- Target Version—FDMEE releases: 11.1.2.4.000 or higher must be installed and configured before you can begin the migration.

Additional Migration Details

1. All mapping scripts are stored in the database and are copied as part of the FDM to FDMEE migration. However, the scripts in FDM use Visual Basic (VB), and FDM supports Jython only. All scripts must be updated to correct the syntax. You can also convert to the SQL script option in FDMEE. SQL scripts perform better than FDM Visual Basic scripts.

   Other scripts are stored in files and not copied.

2. FDM supported several source adapters to connect EBS, SAP, and ERPI. The details related to the EBS and SAP adapter supported directly by FDM are not migrated because there are separate and different adapters for these systems included as a licensed option for FDMEE. In addition, weblink source adapters defined in FDM are not migrated.

3. If the FDM schema has been modified from the default fifty segments, modify the definitions of the TDATASEG_ALL and TDATAMAPSEG_ALL views to add or delete accordingly in the SQL query.

   The SQL query is in aif_fdmc_create_common.sql.

4. An Oracle Essbase target application that gets migrated from FDM to FDMEE is migrated as a “classic” application. You can still edit the application in EPMA or Essbase Administration Services (EAS) as needed, but changes made in EPMA are not reflected unless the application is deployed to Essbase.

FDM Artifacts and their Equivalents in FDMEE

FDM artifacts and their equivalent or similar artifacts in FDMEE are shown below:

<table>
<thead>
<tr>
<th>S1 Number</th>
<th>Artifact</th>
<th>Description</th>
<th>Migrated (Yes/No/Partial/NA)</th>
<th>Equivalent Artifact in FDMEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Application Settings</td>
<td>These are systems settings in FDM that are not migrated.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Integration Settings</td>
<td></td>
<td>Yes</td>
<td>Target application</td>
</tr>
<tr>
<td>3</td>
<td>Configuration Settings</td>
<td></td>
<td>na</td>
<td></td>
</tr>
<tr>
<td>S1 Number</td>
<td>Artifact</td>
<td>Description</td>
<td>Migrated (Yes/No/Partial/NA)</td>
<td>Equivalent Artifact in FDMEE</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>Web Settings</td>
<td></td>
<td>na</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>User Maintenance</td>
<td>These are security settings in FDM that are not migrated.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Object Maintenance</td>
<td>These are security settings in FDM that are not migrated.</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

Table 2  FDM Metadata Artifacts and their Equivalents in FDMEE

<table>
<thead>
<tr>
<th>S1 Number</th>
<th>Artifact</th>
<th>Description</th>
<th>Migrated (Yes/No/Partial/NA)</th>
<th>Equivalent Artifact in FDMEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Locations</td>
<td></td>
<td>Yes</td>
<td>Locations</td>
</tr>
<tr>
<td>8</td>
<td>Import Formats</td>
<td></td>
<td>Yes</td>
<td>Import Formats</td>
</tr>
<tr>
<td>9</td>
<td>Logic Groups</td>
<td></td>
<td>Yes</td>
<td>Logic Groups</td>
</tr>
<tr>
<td>10</td>
<td>Validation Rules</td>
<td></td>
<td>Yes</td>
<td>Check Rules</td>
</tr>
<tr>
<td>11</td>
<td>Validation Entities</td>
<td></td>
<td>Yes</td>
<td>Check Entities</td>
</tr>
<tr>
<td>12</td>
<td>Control Groups and Process Explorer</td>
<td></td>
<td>na</td>
<td>Oracle Hyperion Financial Close Management This is not a FDMEE application, but a separate application.</td>
</tr>
<tr>
<td>13</td>
<td>Control Tables</td>
<td></td>
<td>Yes</td>
<td>Global Categories</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Application Categories</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Global Period</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Application Periods</td>
</tr>
<tr>
<td>14</td>
<td>Dimensions</td>
<td></td>
<td>Yes</td>
<td>Dimensions of target application</td>
</tr>
<tr>
<td>15</td>
<td>Dimension Labels</td>
<td></td>
<td>na</td>
<td></td>
</tr>
</tbody>
</table>

Table 3  FDM Other Artifacts and their Equivalents in FDMEE

<table>
<thead>
<tr>
<th>S1 Number</th>
<th>Artifact</th>
<th>Description</th>
<th>Migrated (Yes/No/Partial/NA)</th>
<th>Equivalent Artifact in FDMEE</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Maps</td>
<td></td>
<td>Yes</td>
<td>Data load mapping (extracted as part of the Setup migration)</td>
</tr>
<tr>
<td>17</td>
<td>Historical Data</td>
<td></td>
<td>Yes</td>
<td>Data load rules and data (including the workflow status). Data load rules are not a concept in FDM, they are automatically created during the migration process, based on the FDM POVs (Location, Category, and Period).</td>
</tr>
<tr>
<td>S1 Number</td>
<td>Artifact</td>
<td>Description</td>
<td>Migrated (Yes/No/Partial/NA)</td>
<td>Equivalent Artifact in FDMEE</td>
</tr>
<tr>
<td>-----------</td>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>18</td>
<td>Process State</td>
<td>Yes</td>
<td>Process State</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>POV Lock Status</td>
<td>Yes</td>
<td>POV Lock Status</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Scripts</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Reports</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Menu Make and Task Flow</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Process Logs</td>
<td>No</td>
<td>Only the log associated with the fish icon display is migrated.</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Error Logs</td>
<td>No</td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Map Converter</td>
<td>na</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Data Files</td>
<td>No files from the FDM folder structure are migrated. This includes data files, mapping files and so on, which were loaded to FDM. You need to copy them manually to the FDMEE application folders.</td>
<td>No</td>
<td>All rules created through the migration point to a dummy file. You need to replace them manually with the actual files copied from the FDM folder structure.</td>
</tr>
</tbody>
</table>

### Installation Steps

**Subtopics**

- **Prerequisites**
- **Install the Patch**
- **Install DB Schema**
- **Install and Configure the ODI Components**

Complete the following steps to migrate FDM setup artifacts and data to FDMEE:

### Prerequisites

Before beginning the migration, make sure you have:

- a running version of FDM
- an installed version of FDMEE 11.1.2.4.x
**Note:** FDMEE does not support parent locations across different target applications. In FDMEE since a new parent has to be created, before running the migration, copy maps between locations in FDM. You copy maps and set correct parent within the same target application, and then migrate with no parent locations across different target applications. In this way you don’t have to recreate maps for the second application.

**Install the Patch**

1. Create a working directory, for example, C:\Oracle\FDM.
2. From My Oracle Support (https://support.oracle.com), download patch 21379349 - FDM to FDMEE Migration Utility.
3. Unzip the patch to the working directory.

**Install DB Schema**

**Update the FDMEE Schema**

1. Using a tool like SQL Developer, log in to the FDMEE schema.
2. When the FDMEE schema is on an Oracle Database, run: aif_fdme_create_oracle.sql.
   When FDMEE schema is on a SQL Server, run: aif_fdme_create_sql_server.sql.

**Update the FDM Schema**

1. Using a tool like SQL Developer, log in to the FDM schema.
2. For the FDM schema, run: aif_fdmc_create_common.sql.
   The aif_fdmc_create_common.sql command works for both an Oracle and a SQL Server database.

**Install and Configure the ODI Components**

The following steps needs to be performed using ODI Studio on your work repository:

1. Start the Oracle Data Integrator Studio (ODI).
2. From Topology, then Logical Architecture, and then Technologies, select Generic SQL.
3. Right click Generic SQL, and then select New Logical Schema.
4. Create a logical schema named **FDMC_SOURCE**.

5. From **Topology**, then **Physical Architecture**, and then **Technologies**, select **Oracle** or **Microsoft SQL Server**.
   
   a. Right click the Oracle or Microsoft SQL Server, and then select **New Data Server**.
   
   b. Create a data server named **FDMC_DATA_SERVER**.
      
      The name can also be based on any specific client requirements.
   
   c. Right click the data server created in step b, and then select **New Physical Schema**.
   
   d. Create a physical schema that points to the FDM database.

6. Edit the **GLOBAL** context (or the name of the context that you created for the FDM source), and then map the **FDMC_SOURCE** logical schema (created in step 3) to its physical schema (created in step d).
For information on setting up contexts, see “Setting up the ODI Context Code” in the *Oracle Hyperion Financial Data Quality Management, Enterprise Edition Administrator’s Guide*.

AIF_TGT designates the target FDMEE database schema. If needed, you can create a new context to designate another source or target schema.

**Note:** Under **context**, you can have one context for each FDM application, or you can overwrite the existing context after each migration. It is recommended to use one context and physical schema for each FDM application (typically in scenarios where there are multiple FDM applications, there are multiple contexts and physical schemas.)

7. From **Designer**, then **Load Plans**, and then **Scenarios**, right click and select **Import Scenarios**.
Note: From the patch, copy the XML files (SCEN_FDMC_EXTRACT_SETUP_001.xml and SCEN_FDMC_EXTRACT_DATA_001.xml) to the work repository directory:

<Oracle_Middleware_Home>EPMSystem11R1\products\FinancialDataQuality\odi\11.1.2.4.0\workrepo>

8. In IMPORT_UPDATE mode, import the FDMC_EXTRACT_SETUP scenario and FDMC_EXTRACT_DATA scenario into the work repository.

Make sure that the Import Type shows Synonym Mode INSERT_UPDATE.

The FDM migration uses the FDMC_EXTRACT_SETUP scenario to migrate setup data, and uses the FDMC_EXTRACT_DATA scenario to migrate application data.
Executing the Migration

Subtopics
- Extracting Setups
- Migrated Setup Examples
- Extracting Data
- Migrated Data Load Mapping Examples

Migration enables you to extract setups and data from FDM.

Extracting Setups

During the migration, various setup artifacts from FDM are migrated and created as artifacts in FDMEE. For details about the migrated artifacts, see Table 1, “FDM Administration Artifacts and their Equivalents in FDMEE”.
Note: If the target application is an EPMA deployed application, register the application in the FDMEE target application user interface before extracting the setups from FDM. For more information, see the “Registering Target Applications” topic in Oracle Hyperion Financial Data Quality Management Administrator’s Guide.

The following steps are performed from ODI Studio in your work repository.

To execute the migration:

1. Select the ODI Scenario FDMC_EXTRACT_SETUP, and then right click and select Execute.

2. From the Execute screen, select the context, and click OK.
From the Variable Values screen, specify the following values for each variable:

- **FDMC.p_application_name**—Specify the target application name, for example, enter TotConsol.
  
  For data mart adapters, specify a name for the custom target application.

- **vp_application_type**—Specify the application type.
  
  Supported application types are:
  - CUSTOM
  - ESSBASE
  - HPL
  - HFM

  Application types must be entered in uppercase characters.

- **FDMC.p_application_db_name**—Specify the following:
  
  For Essbase applications, specify the Essbase database name. If multiple databases were used in FDM, execute the migration multiple times by specifying each database.
  
  For Oracle Hyperion Planning applications, specify a comma separated list of six plan names in the same order as created in Planning.
  - Example 1: If the Planning application has three plan names: Plan1 as the first plan name, Plan3 as the third plan name, and PFP as the sixth plan name, enter Plan1,,Plan3,,,PFP
  - Example 2: If the Planning application has Plan1 only as the first plan name, enter Plan1,,,,

  **Note:** If you have a Planning or Essbase application in FDM with the same name as an Oracle Hyperion Enterprise® or Oracle Hyperion Strategic Finance application, the migration fails. To resolve this, change the name of the Hyperion Enterprise or Strategic Finance application in their adapters to a dummy value, and then run the Planning or Essbase application migration. After the migration is complete, the Hyperion Enterprise or Strategic Finance adapter setting can be restored.

  For custom applications, specify the system key of the PULL/DATAMART adapter.
  
  For Financial Management applications, this parameter is ignored.

- **FDMC.p_prefix**—(optional) When a prefix is specified, it is prefixed to all setup data. When you run the scenario again, specify the same prefix to avoid duplicate data or a program failure. You can specify a value of up to 10 characters.
The following shows how to set variable values when extracting setups for a Planning application from FDM.

4 Click **OK** to initiate the execution process.

From **ODI Studio**, select the **Operator** tab, and monitor the progress of the migration.
After executing the FDMC_EXTRACT_SETUP process, review the processes and ensure that all sub-processes for the ODI execution are successful.
Migrated Setup Examples

Subtopics

- Migrated Artifacts in Import Format
- Migrated Artifacts in Location
- Migrated Artifacts in Logic Group
- Migrated Artifacts in Check Rule Group

Below are examples setup artifacts migrated from FDM to FDMEE.

Migrated Artifacts in Import Format

The first example shows artifacts migrated to import formats in FDMEE.

Migrated Artifacts in Location

The second example shows artifacts migrated to a location in FDMEE.
Migrated Artifacts in Logic Group

The third example shows artifacts migrated to a Logic Group in FDMEE.

Migrated Artifacts in Check Rule Group

The fourth example shows artifacts migrated to a Check Rule Group in FDMEE.
Extracting Data

During the migration, historical data is available to extract from FDM to FDMEE. Typically, you may not need to extract very old data. You can select the Period range and Category from which data is required.

To execute the migration to extract data:

1. Select the ODI Scenario **FDMC_EXTRACT_Data**, and then right click and click Execute.
2. From the Variable Values screen, specify the following values for each variable:
   - **FDMC.p_application_name**—Specify the target application name, for example, enter TotConsol.
   For data mart adapters, specify a name for the custom target application.
   - **FDMC.p_application_type**—Specify the application type.
     Supported application types are:
     - CUSTOM
     - ESSBASE
     - HPL
     - HFM
     Application types must be entered in uppercase characters.
   - **FDMC.p_application_db_name**—Specify the following:
For Essbase applications, specify the Essbase database name. If multiple databases were used in FDM, execute the migration multiple times by specifying each of the databases.

For Planning applications, specify a comma separated list of six plan names in the same order as created in Planning.

- Example 1: When the Planning application has three plan names: Plan1 as the first plan name, Plan3 as the third plan name, and PFP as the sixth plan name, enter `Plan1,,Plan3,,,PFP`.
- Example 2: When the Planning application has Plan1 only as the first plan name, enter `Plan1,,,,,`.

If you have a Planning or Essbase application in FDM with same name as a Hyperion Enterprise or Strategic Finance application, the migration fails.

To resolve this, change the name of Hyperion Enterprise or Strategic Finance application in their adapters to a dummy value and run the Planning or Essbase application migration. After the migration is complete, the Oracle Hyperion Enterprise® or Oracle Hyperion Strategic Finance adapter setting can be restored.

For custom applications, specify the system key of the PULL/DATAMART adapter.

For Financial Management applications, this parameter is ignored.

- `FDMC.p_data_migration_category_name`—Specify the POV Category from which to extract. For example, you might specify `ACTUAL`.
- `FDMC.p_data_migration_start_periodkey`—Specify the Start date Period Key from which to extract data. For example, you might specify `2001-01-31`.
- `FDMC.p_data_migration_end_periodkey`—Specify the End date Period Key from which to extract data. For example, you might specify `2015-12-31`.

The following shows how to set variable values when extracting data for a Financial Management application from FDM.
The following shows how to set variable values when extracting data for a Planning application from FDM.

3 Click **OK** to initiate the execution process.

From **ODI Studio**, and then the **Operator** tab, monitor the progress of the migration.
After executing the FDMC_EXTRACT_DATA process, review the processes and ensure that all the sub-processes for the ODI execution are successful.
Migrated Data Load Mapping Examples

Subtopics

- Migrated Data in a Data Load Rule in FDMEE
- Migrated Data in Data Load Workbench

Below are examples of data load mapping when data is migrated from FDM to FDMEE.

Migrated Data in a Data Load Rule in FDMEE

The first example shows how data is migrated to a data load rule.

![Data Load Rule in FDMEE](image1)

Migrated Data in Data Load Workbench

The second example shows how migrated data appears in the Data Load Workbench.

![Data Load Workbench](image2)
Drill Region Post Migration Steps

After migration, drill regions are handled in several ways depending on the target application.

Drill Region for Essbase

For Essbase, you must edit the drill URL using the Essbase Administration Services (EAS) Console and change the URL to point to FDMEE.

1. Define a drill region for old data from the Essbase and Planning applications.
2. Launch the Essbase Administration Services (EAS) Console.
3. From the left pane, select the <application> and then the <database>.
4. Right click the <database>, select Edit and then select Drill through Definitions.
5. In the Drill through Definition window, enter the drill through definition.
   For example, enter FDM_DRILL_REGION.
6. Import the drill through XML content by clicking Import XML and then selecting DrillRegionXML.xml.
   The DrillRegionXML.xml file is included in the migration patch.
7. In Region Definition, enter the Member Set Functions using the Year, Period, and Scenario dimensions.
   For example, enter @DESCENDANTS ("Period") @LIST ("FY10", "FY11") @LIST ("Actual").
8. Save the definition.
9. After migrating an Essbase application, enable drill through for the application in FDMEE on the Application Options screen in the Target Applications option. This step includes checking the dimension check boxes and setting the application option for drill to Yes. The migration does not set the drill through option automatically for Essbase applications.

**Drill Region for Financial Management**

For Financial Management, a file with the extension .DRL is created and exported when data is loaded and consolidated. FDMEE loads this file to use with the target application.

However; to use the drill URL in FDMEE after the migration, you must run the `aif_fdmee_misc.sql` script. To do this, copy the script from the #SQL1 section.

Then, select the Level 0 members only check box.

**Drill Down for Old Data in an Oracle Database**

Drill down for old data may not work in an Oracle database.

FDM converts all target member values in mappings to uppercase characters. And during drill down, FDM converted all the values to uppercase characters before querying the matching intersections.

Conversely, FDMEE is case sensitive and handles all values in the format entered by the user.
As a result, during the drill down the member names passed from the target application do not match the values in the staging table.

This issue applies to an Oracle database implementation only.

**Check Rules Post Migration Steps**

After the migration, you need to recreate any check rules in Jython script for the target application. Validation check rule functions in FDM were written in Visual Basic (VB).

**Check Rules Not Compatible with Essbase and Planning**

Check Rules defined in FDM for Essbase and Planning applications are not compatible with the FDMEE format. All rules need to be converted from FDM format to FDMEE format either manually or using the member selector in FDMEE.

Also, when FDM is on an Oracle database, review all rules and fix cases for all members so that they match the correct case of members in the source FDMEE tdataseg table (for converted), or target application depending on the type of intersection.

**FDM Formats**

**FDM Target System Data**

| Scenario, Period, Year, View, Value, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20 |

**FDM Source System Data**

- Scenario, Period, Year, View, Value, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20

**FDMEE Formats**

FDMEE has different formats for various target and source intersections. Convert rules to the appropriate FDMEE format.

**Essbase Target System Data**

| Scenario, Period, Year, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20 |
Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20

**Essbase Source System Data**

~Scenario, Period, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20~

**Essbase Converted System Data**

`Scenario, Period, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20`

**Planning Target System Data**

|Scenario, Period, Year, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20|

**Planning Source System Data**

~Scenario, Period, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20~

**Planning Converted System Data**

`Scenario, Period, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20`

**Planning Multi-Currency Target System Data**

|Scenario, Period, Year, View, Currency, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20|
Planning Multi-Currency Source System Data

~Scenario, Period, View, Currency, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20~

Planning Multi-Currency Converted System Data

`Scenario, Period, View, Currency, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20`

Check Rules Not Compatible with Financial Management

Source and converted intersection check rules defined in FDM for Financial Management applications are not compatible with the FDMEE format. All source/converted rules need to be converted from FDM format to FDMEE format either manually or using the member selector in FDMEE.

When FDM is on Oracle database, review all rules, and fix the case for all members to match the correct case of members in the source or FDMEE tdataseg table (for converted). All Oracle Hyperion Financial Management target intersections should work without any need of modifications to the format or members case.

FDM Common Format for Intersections

FDM Target Intersections

|Scenario, Period, Year, View, Value, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20|

FDM Source Intersections

~Scenario, Period, Year, View, Value, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20~

FDMEE Common Format for Intersections:

FDMEE has different format for source/converted intersections. Convert rules source/converted intersections to the FDMEE format.
Financial Management Source Intersections
~Scenario, Period, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20~

Financial Management Converted Intersections
`Scenario, Period, View, Entity, Account, ICP, Custom1, Custom2, Custom3, Custom4, Custom5, Custom6, Custom7, Custom8, Custom9, Custom10, Custom11, Custom12, Custom13, Custom14, Custom15, Custom16, Custom17, Custom18, Custom19, Custom20`

Additional Post Migration Steps
When you have successfully migrated all applications from FDM to FDMEE, and performed the drill region and check rule post migration steps, you may need to complete these additional tasks:

1. **Fix mappings in the Oracle Database**—When FDM is installed on an Oracle database, it stores target application members in uppercase characters. For example, it stores a Oracle Hyperion Planning member, such as Cash as CASH. However; FDMEE stores members exactly as they are on the target application.

   To address this, run the refresh members process from the target application.

   When this is done, run the SQL from aif_fdmee_misc.sql to update the data mappings. Use the SQL under the #SQL2 section.

   All category mappings cases have to be fixed manually in the Category Mappings option.

2. **Delete any duplicate and old setup ERPI artifacts**—If you have an ERPI adaptor-based integration, delete any duplicate and old setup ERPI artifacts from FDMEE.

   You complete this step because there are artifacts from ERPI that get upgraded to FDMEE, and artifacts that get migrated from FDM to FDMEE. It is recommended that the ERPI artifacts that were upgraded are deleted.

3. **Upload and associate your data files**—After the migration, data rules appear as if they were run. That is, the check marks and fish indicate that the rules were run and completed. However; the rules have not been run after the migration. If you try to execute a file-based data load rule, it fails since the data file is never migrated. When the data rule is created, it is associated with the "Sample.txt." You must upload and associate your data files. Data files are not migrated.

   To run the data load rules after the migration, note the following:

   a. File-based data loads need to have the data file loaded into the rule. After the migration it shows as "Sample.txt."

   b. Validate mappings (particularly to validate upper case issues).
c. If an application was created or imported to Oracle Essbase without data, and the rule was not run, data does not appear in Oracle Smart View for Office. (It may appear that data was loaded in the FDMEE user interface.)

4. **Update functional currency**—After the migration of a target application, the functional currency field for locations shows as [NONE]. You must manually update the functional currency field to the expected value.

5. **Remove additional data models**—When all FDM data has been migrated and you have completed all migration scenarios, remove the additional data model created during the migration.

   When FDMEE is on an Oracle server, run: `aif_fdmee_drop_oracle.sql`.

   When FDMEE is on a SQL Server, run: `aif_fdmee_drop_sql_server.sql`.

6. **Update target application**—When the calculation sequence for all dimensions is “0” in Oracle Hyperion Financial Data Quality Management, then after the upgrade, you cannot modify the target application settings.

   To fix this, run the **update calc sequence SQL** from the `aif_fdmeeMisc.sql` file. Use the SQL under the `#SQL3` section.

### Testing and Troubleshooting Tips

This section provides solutions to common migration issues.

- Specify a prefix to avoid duplicate key issues.
  - Avoid the use of special characters like dash (-) in the prefix.

- Each time you execute a scenario, a log file named `fdmClassicUpgrade.log` is created or updated in the operating system’s default temporary folder. Use this log to review and troubleshoot any scenario issues.

  In Windows, to find the location of the temporary folder:

  1. Click **Start**.
  2. In **Search**, type **CMD**.
  3. At the command prompt, type `ECHO %Temp%`.

    The full path of the temporary folder is displayed.

   Other operating systems may have the temporary folder located elsewhere. In this case, consult your operating system documentation.

- You can rerun the migration process to resolve any issues. If you wish to rerun the migration process from scratch, first delete the migrated application in Oracle Hyperion Financial Data Quality Management, Enterprise Edition from the Target Applications option, and run the migration again.