

# **Data and Metadata Integration with Planning & Essbase Using Oracle Data Integrator (ODI)**

**Trey Daniel**

**Financial Systems, Reporting, and Processes @ GameStop**

February 19, 2015



# About Me

---

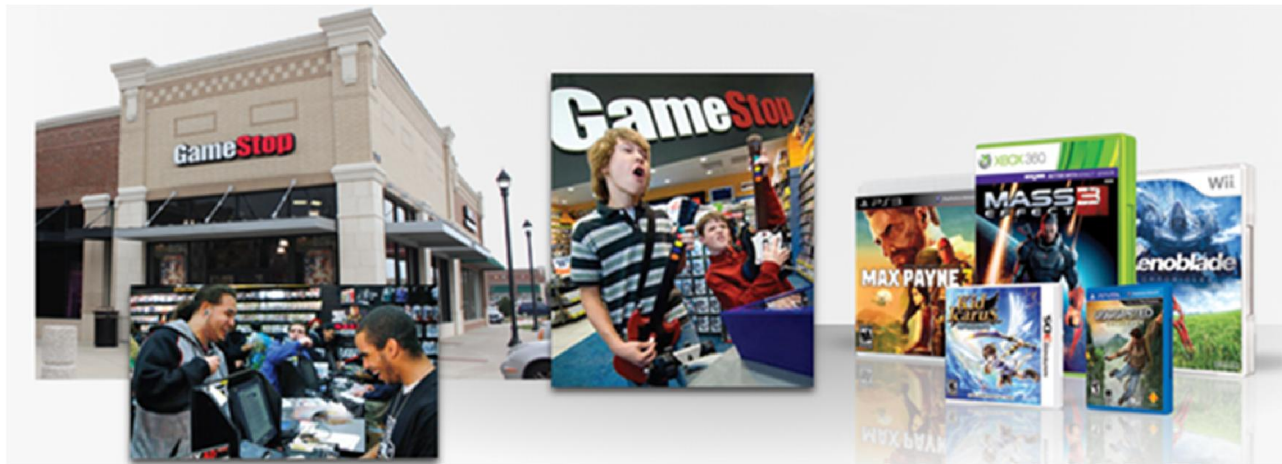
- B.S., Management Information Systems, Louisiana State University (LSU)
- MBA, Louisiana State University (LSU)
- Over 10 years working with Hyperion as both Consultant/Oracle Partner & Customer
  - PricewaterhouseCoopers, LLP (PwC), 2004 to 2010
  - Accenture, 2010 to 2014 (with 2 years as US Hyperion Data Integration Lead)
  - Financial Reporting, Systems, and Processes @ GameStop, 2014 to Present
- Oracle Certified Implementation Specialist
  - HFM 11 (2012)
  - Essbase 11 (2013)



# About GameStop

---

- “A global multichannel video game, consumer electronics and wireless services retailer with more than 6,600 stores worldwide, GameStop makes the most popular technologies affordable and simple”



- Besides GameStop stores and online, the GameStop family also includes:
  - **GameInformer** (world's leading print and digital video game publication)
  - **Kongregate** (browser-based game site)
  - **BuyMyTronics** (online consumer electronics trade-in platform)
  - **Simply Mac** (Apple authorized seller of all Apple products & certified warranty + repair services)
  - **Spring Mobile** (post-paid AT&T services and wireless products)
  - **Cricket Wireless** (AT&T-branded pre-paid wireless services, devices and accessories)

# Hyperion Data & Metadata Integration Products

---

## Over The Years:

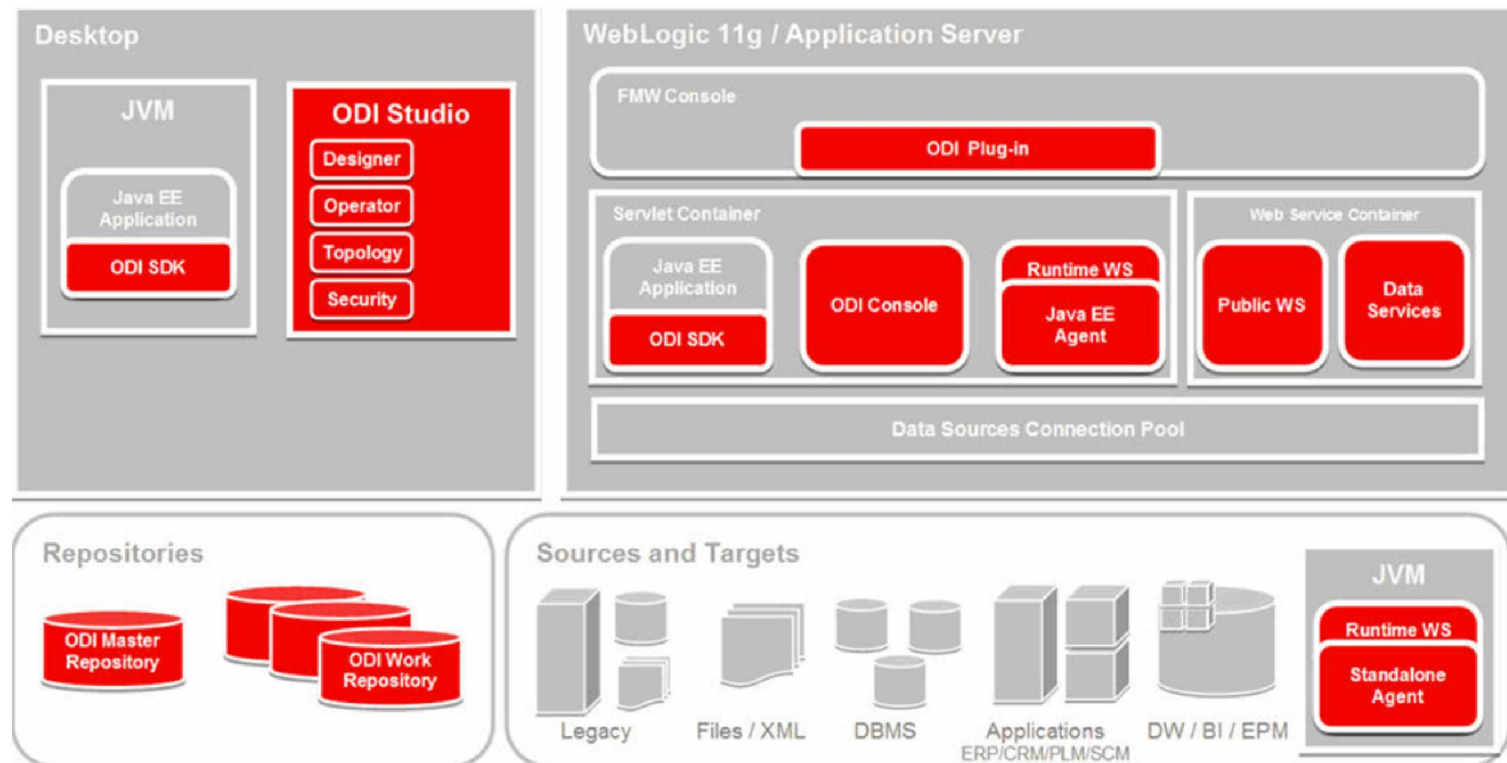
- HAL
- LedgerLink
- DIM
- Enterprise Conversion Tables
- Upstream / FDM
- EAS
- EIS
- EAL
- Essbase Studio
- ODI
- EPMA
- ERPi
- FDMEE
- Razza / DRM
- EA

## 2015:

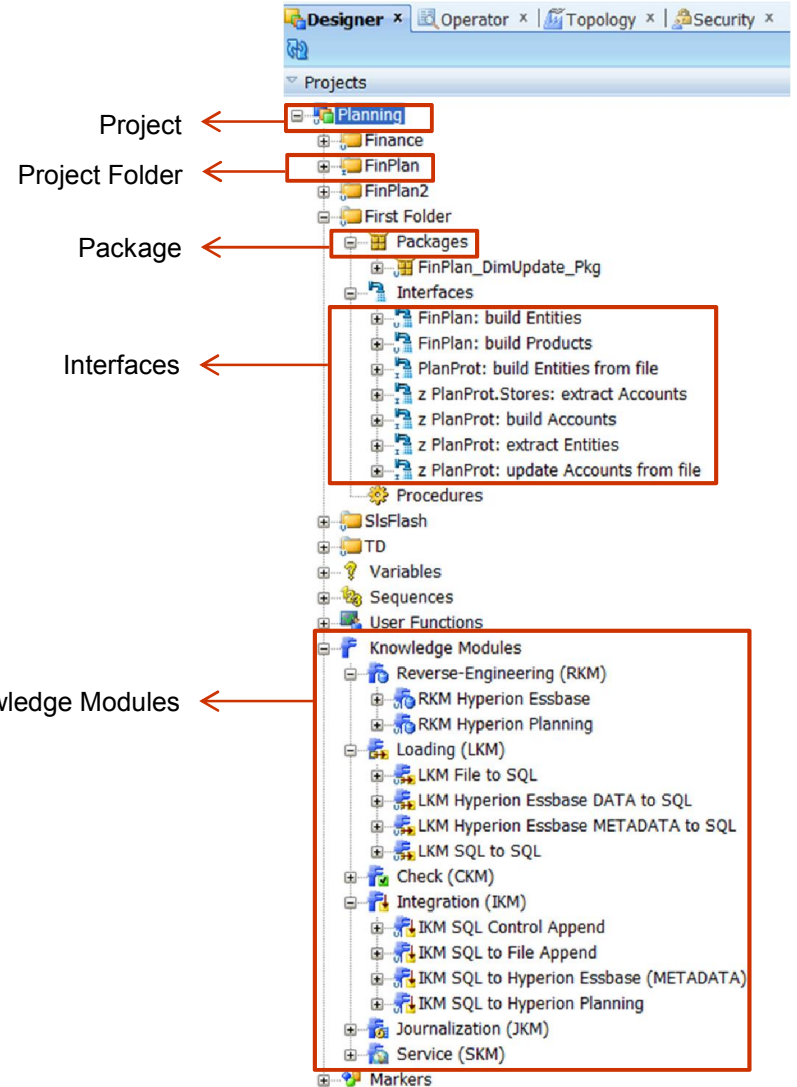
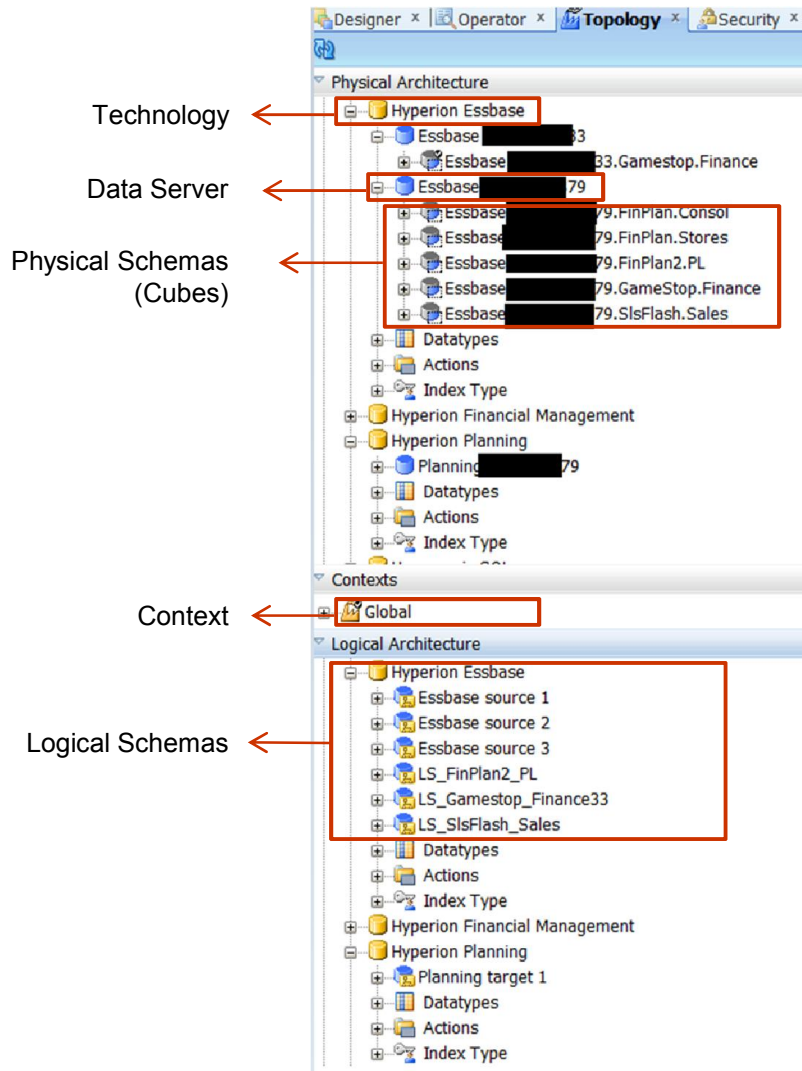
- HAL
- LedgerLink
- DIM
- ~~Enterprise Conversion Tables~~
- ~~Upstream / FDM~~
- **EAS**
- ~~EIS~~
- ~~EAL~~
- Essbase Studio (*low market share vs. EAS*)
- **ODI**
- EPMA (*primarily metadata*)
- ~~ERPi~~
- **FDMEE**
- ~~Razza / DRM~~ (*metadata only*)
- EA (*data only*)

# About ODI

- Oracle's Fusion Middleware ELT (Extract – Load – Transform) application for integrating data between almost any type of application (RDMBS, OLAP, Big Data, MS Office, Flat File, etc.)
- Oracle's approach to defining metadata/data transformation and integration processes through a unified infrastructure & utilizing RDBMS engines



# Key ODI Elements





# Key ODI Elements (Continued)

The screenshot displays the Oracle Data Integrator (ODI) interface with four tabs: Designer, Operator, Topology, and Security. The Security tab is active, showing a list of profiles and users. The Session List on the left shows various sessions, with a red box highlighting the 'Sessions' section. The Profiles section on the right lists various roles, with a red box highlighting the 'CONNECT' profile. The Users section on the right shows a list of users, with a red box highlighting the 'Users' section. The Objects section on the right lists various ODI objects, with a red box highlighting the 'Objects' section. Red arrows point from text labels to these specific areas.

Execution Logs ←

ODI Security Profiles ←

ODI Users ←

ODI Objects ←

Scenarios ←

# Inside an Interface (Mapping)

Source Datastore (SQL Database) →

Field Filter →

Target Datastore (Hyp. Planning) →

Data Mapping (SQL) →

Data Transformation

The screenshot displays the 'FinPlan: build Products' interface. On the left, the 'Source Datastore (SQL Database)' is '1 - ProductCategoryHierarchy (PCH)', with fields like Parent, NAME, SKU, Title, ItemNum, \*Gen, \*CompanyCode, and \*CountryCode. A 'Field Filter' icon is shown. The 'Target Datastore (Hyp. Planning)' is 'Products', with a table listing 17 fields and their mappings. The 'Mapping' column contains SQL snippets like 'L'+cast(PCH.Gen as char(1))+ '-' +Name and 'case when PCH.Gen=1 then Parent else'. The 'Data Mapping (SQL)' section shows the 'Parent - Property Inspector' with 'Mapping Properties' and a 'Technical Description' tab. The 'Implementation' tab shows the SQL snippet: 'case when PCH.Gen=1 then Parent else 'L'+cast(Gen-1 as char(1))+ '-' +Parent end'. The 'Execute on' section has radio buttons for 'Source', 'Staging Area', and 'Target', with 'Source' selected. The 'Source Datastore' is 'ProductCategoryHierarchy (PCH)'. The 'Insert' and 'Update' checkboxes are checked.

Position	Indicators	Name	Mapping
1	S	Products	'L'+cast(PCH.Gen as char(1))+ '-' +Name
2	S	Parent	case when PCH.Gen=1 then Parent else
3	S	Alias: Default	rtrim(Title)+' (L'+cast(PCH.Gen as char(1))+ '-' +Name
4	S	Valid For Consolidations	
5	S	Data Storage	case when PCH.Gen=4 then 'Never Shar
6	S	Two Pass Calculation	
7	S	Description	
8	S	Formula	
9	S	UDA	
10	S	Smart List	
11	S	Data Type	
12	S	Operation	
13	S	Hierarchy Type	
14	S	Plan Type (Stores)	1
15	S	Aggregation (Stores)	
16	S	Platform	case when PCH.Gen=5 then Title end
17	S	Sales Driver	

Parent - Property Inspector

Mapping Properties

Active Mapping: ☒

Implementation Technical Description Business Rule

case when PCH.Gen=1 then Parent else 'L'+cast(Gen-1 as char(1))+ '-' +Parent end

Execute on: ☒ Source ☐ Staging Area ☐ Target

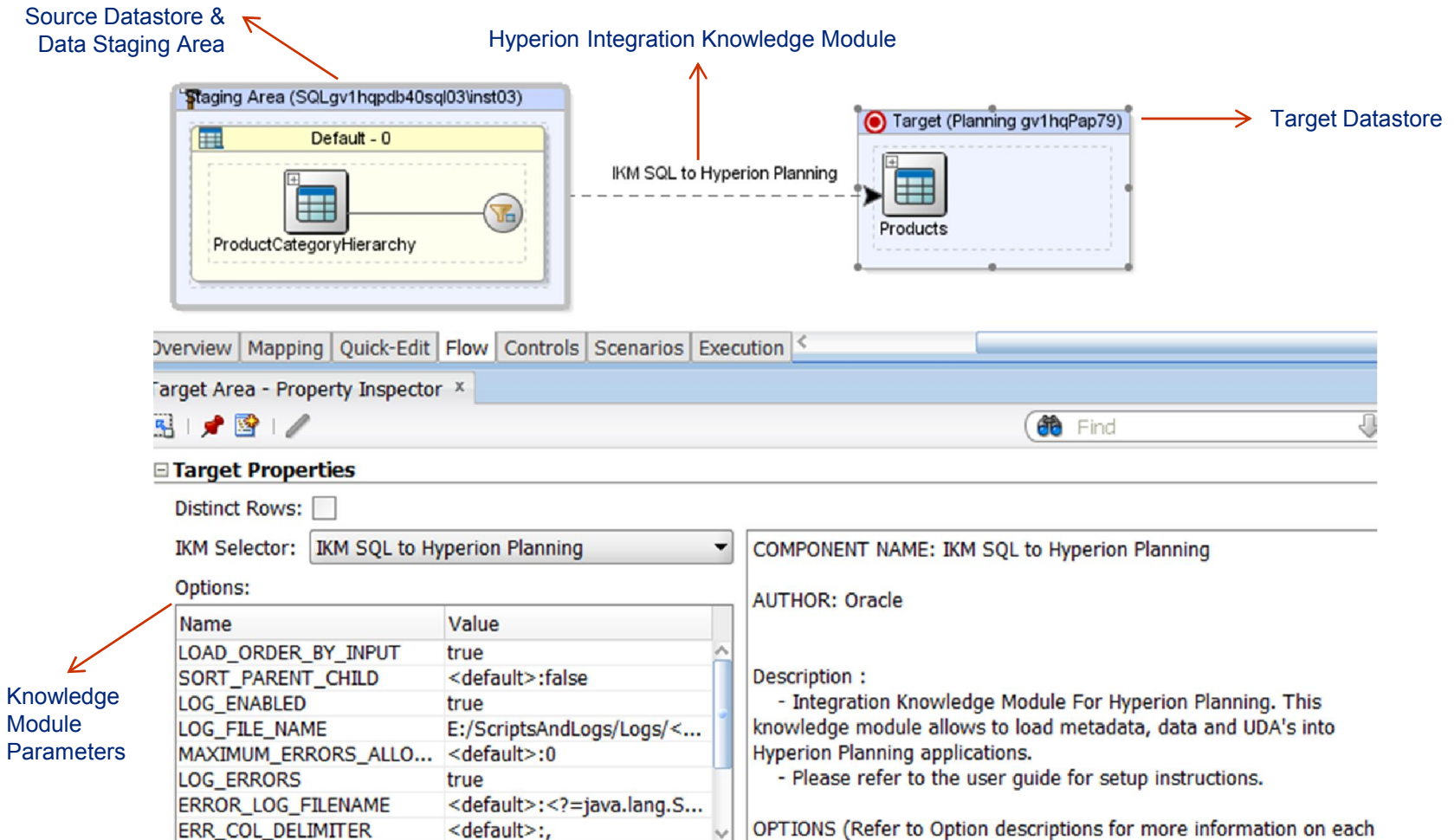
Source Datastore: ProductCategoryHierarchy (PCH)

Insert: ☒

Update: ☒



# Inside an Interface (Flow)



# Configuring an Integration with Hyperion

---

1. Create a Data Server under “Hyperion Planning” or “Hyperion Essbase” Topology
2. Create a Physical and Logical Schema for the cube, assigned to a Context (*Global is default*)
3. Create a Model, using the Hyperion RKM to pull in the dimension and members as Datastores
4. Create an Interface using the Hyperion datastores as a Source or Target Datastore (or both)
5. Perform field and data transformation mappings in the Interface
6. Configure Hyperion IKM and LKM flows to import and export data where needed (*Needs .rul/.rpt*)
7. Execute the Interface to perform the integration
8. Create a Packing to layer in additional Interfaces and other actions
9. Create a Scenario to compile the package, making it available for scheduling & automation programs
10. Schedule nightly/weekly/etc. automated integrations with Windows, Tibco, Tidal, etc.

## Knowledge Modules for Hyperion (per ODI 11.1.1.7)

---

**Reverse Engineer (RKM)** – Pulls the Hyperion dimension and member structure into ODI for mapping

- *RKM Hyperion Essbase*
- *RKM Hyperion Planning*

**Load (LKM)** – Loads data/metadata from Hyperion into staging table

- *LKM Hyperion Essbase DATA to SQL*
- *LKM Hyperion Essbase METADATA to SQL*
- *IKM SQL to Hyperion Planning [for data and metadata]*

**Integration (IKM)** – Loads data/metadata from staging area into Hyperion

- *IKM SQL to Hyperion Essbase (DATA)*
- *IKM SQL to Hyperion Essbase (METADATA)*

Also recommend importing “LKM File to SQL” and “IKM SQL to File Append” to test loading flat files to a SQL staging area or exporting SQL staging data to a flat file, respectively, for testing purposes

# Better Practices for Hyperion Integration

---

- Test against a flat file on the server/laptop first to eliminate additional points of failure
- Compare against metadata before data loads for error trapping
- String together multiple Interfaces via Packages
- Automate Packages where possible via Scenarios & pre-packaged ODI Windows .bat files
- Set up email notifications on both Pass and Fail conditions
- Customize where needed via Java and built-your-own Knowledge Modules

## ODI-centric blogs for much more info:

<http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html> (*ODI Homepage*)

<https://blogs.oracle.com/dataintegration> (*Oracle's Official ODI Blog*)

<https://gurcanorhan.wordpress.com> (*Gurcan Orhan*)

<http://devepm.com/> (*DEV EPM*)

<http://john-goodwin.blogspot.com/> (*More To Life Than This*)

## ODI vs FDMEE

---

Discussion Point	ODI	FDMEE
Primary Users	IT	Finance
Out-of-the-box availability	Minimal	Heavy
Customization	High with included tools	Available with Jython
Primarily Supports	Any technology	Hyperion applications
Workflow and Automation	Must be built	Included & configurable
Product Maturity	Mature	Early
HFM Support	Up to 11.1.2.3 (OOTB)	Ongoing
Configuration	Multiple, deep steps	Drop boxes, forms
Reporting Capabilities	None within app	Multiple pre-built
Ongoing EPM Support	Uncertain	Part of EPM product family

## ODI and Hyperion 11.1.2.4

---

- Per Oracle Support Document 1969035.1 (“Oracle Enterprise Performance Management and Oracle Data Integrator Statement of Direction”), released Feb 9, 2015:
  - “Starting with EPM release 11.1.2.4, a new Planning Adapter will be added to the Application Adapters for Data Integration for the ODI release 12c, but the HFM KM will not be available”
  - “Please note that the ODI KMs for stand-alone Essbase will still be available and the licensing has not been changed”
  - “Users can use ODI to load HFM in release 11.1.2.4, but will need to develop custom ODI content and call the HFM java APIs from ODI to load data and metadata”
  - “FDME uses a version of ODI that is installed and configured during the EPM installation process, and is not impacted by the future certification status of the Knowledge Modules.”
  - “The KM’s for EPM release 11.1.2.3 and ODI release 11.1.1.7 are not certified with EPM release 11.1.2.4.”

<https://support.oracle.com/epmos/faces/DocumentDisplay?id=1969035.1>



## Appendix: Key ODI Elements (Definitions)

---

Master Repository: main database for all ODI foundational elements

Work Repository: main database for a specific ODI Project(s) elements

Data Server: Vendor-specific database or file for source or target data

Physical & Logical Schemas: – Vendor-specific settings and addresses for the Data Server

Context: A grouping of Physical and Logical schemas

Model: Collection of datastores using a Reverse Knowledge Module to retrieve file/database layouts

Datastore: A specific file or table associated with a Model

Project: A collection of KMs, Interfaces, Packages, and Scenarios for an integration

Knowledge Module: Pre-written adapters/connectors to almost any data source/target

Interface: Field and data mappings between a Source and Target Datastore

Package: Collection of Interfaces and other actions used to automate the integration

Scenario: A compiled Package able to be called by scheduling programs