

Oracle® Demand Signal Repository

User Guide

Release 12.1

Part No. E12276-07

May 2010

Oracle Demand Signal Repository User Guide, Release 12.1

Part No. E12276-07

Copyright © 2010, Oracle and/or its affiliates. All rights reserved.

Primary Author: Jason Lansdowne

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

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. 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, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Send Us Your Comments

Preface

1 Introduction to Demand Signal Repository

Introduction.....	1-1
Business Intelligence Dashboard.....	1-2
Data.....	1-3
Measures.....	1-4
Exception Measures.....	1-11
ACV Measures.....	1-12
Goals and Thresholds.....	1-12
Reports and Scorecards.....	1-13
Multiple Hierarchies.....	1-14
Clusters.....	1-15
Retail Clusters.....	1-16
Store Clusters.....	1-16
Item Clusters.....	1-17
Calendars.....	1-18
Data Allocation and Aggregation.....	1-20
Currencies	1-20

2 Data Model

Data Model Overview.....	2-1
Organization and Business Units.....	2-1
Customer Organization Hierarchy.....	2-2

Reference Data Load Process	2-2
Fact Data Load Process.....	2-5

3 Users and Security

User and Security Overview.....	3-1
User Roles.....	3-1

4 Dashboards and Scorecards

Category Sales Analysis Dashboard.....	4-1
Manufacturer Promotions Dashboard.....	4-2
Sales Performance by Account Dashboard.....	4-5
Scorecards.....	4-5
Sales Scorecard.....	4-8
Supply Chain Scorecard.....	4-8
Operations Scorecard.....	4-9
Reports.....	4-9

5 Exceptions

Exceptions Overview.....	5-1
Exception Criteria.....	5-1
Exceptions Dashboard.....	5-5
Exceptions Setup.....	5-7
Processing Exceptions.....	5-9

6 Measures

Measures Overview.....	6-1
Loading Syndicated Consumption Measures.....	6-22
Setting Goals.....	6-27
Setting Thresholds.....	6-28

7 Working with Data

Data Overview.....	7-1
Hierarchy Loading.....	7-2
Alternate Organization Hierarchies.....	7-3
Time and Organization Allocations.....	7-16
Data Loading Interface Tables.....	7-18
Replenishment Rules.....	7-22
Materialized Views.....	7-24
User-Defined Fields.....	7-24

Forecast Types and Purpose Codes.....	7-26
Manufacturer Promotions	7-29
Enabling Third Party Distributors.....	7-35
Manufacturer Shipments.....	7-42

8 Implementing Demand Signal Repository

Implementation Overview.....	8-1
Setting Up Codes and Calendars.....	8-2
Loading Organizations and Hierarchies.....	8-15
Loading Items and Hierarchies.....	8-24
Importing Fact Data.....	8-34
Loading Syndicated Consumption Data.....	8-41
Implementing User-Defined Fields.....	8-43
Manufacturer Promotions.....	8-45
Third Party Distributors.....	8-47
Manufacturer Shipments.....	8-48
Alternate Organization Hierarchy.....	8-49

9 Demand Signal Repository Integration

Business Process Task Flow.....	9-1
Web Services.....	9-2
Loading TDLinx Store/Outlet Data.....	9-5
Enterprise Resource Planning to Demand Signal Repository Integration.....	9-6
Demand Signal Repository to Demantra Demand Management Integration.....	9-8
Demand Signal Repository to Retail Merchandising System Integration.....	9-13
RMS Configuration Steps.....	9-15
Attribute Mappings.....	9-17

Index

Send Us Your Comments

Oracle Demand Signal Repository User Guide, Release 12.1

Part No. E12276-07

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Note: Before sending us your comments, you might like to check that you have the latest version of the document and if any concerns are already addressed. To do this, access the new Oracle E-Business Suite Release Online Documentation CD available on My Oracle Support and www.oracle.com. It contains the most current Documentation Library plus all documents revised or released recently.

Send your comments to us using the electronic mail address: appsdoc_us@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 12.1 of the *Oracle Demand Signal Repository User Guide*.

See Related Information Sources on page x for more Oracle E-Business Suite product information.

Deaf/Hard of Hearing Access to Oracle Support Services

To reach Oracle Support Services, use a telecommunications relay service (TRS) to call Oracle Support at 1.800.223.1711. An Oracle Support Services engineer will handle technical issues and provide customer support according to the Oracle service request process. Information about TRS is available at

<http://www.fcc.gov/cgb/consumerfacts/trs.html>, and a list of phone numbers is available at <http://www.fcc.gov/cgb/dro/trsphonebk.html>.

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible to all users, including users that are disabled. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The

conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

Structure

- 1 Introduction to Demand Signal Repository**
- 2 Data Model**
- 3 Users and Security**
- 4 Dashboards and Scorecards**
- 5 Exceptions**
- 6 Measures**
- 7 Working with Data**
- 8 Implementing Demand Signal Repository**
- 9 Demand Signal Repository Integration**

Related Information Sources

This is Release 12.1 of the Oracle Demand Signal Repository User's Guide. This manual describes the features that are available when you access Oracle Demand Signal Repository. If this guide refers you to other Oracle Applications documentation, use only the 12.1 versions of those guides.

Documentation on Oracle*MetaLink* can be found at <http://metalink.oracle.com>.

Integration Repository

The Oracle Integration Repository is a compilation of information about the service endpoints exposed by the Oracle E-Business Suite of applications. It provides a complete catalog of Oracle E-Business Suite's business service interfaces. The tool lets users easily discover and deploy the appropriate business service interface for integration with any system, application, or business partner.

The Oracle Integration Repository is shipped as part of the E-Business Suite. As your instance is patched, the repository is automatically updated with content appropriate for the precise revisions of interfaces in your environment.

Advanced Planning Command Center User's Guide

Oracle Advanced Planning Command Center unifies all the Advanced Planning applications, such as Demand Management, Real-Time Sales and Operations Planning,

Strategic Network Optimization, Advanced Supply Chain Planning, Distribution Requirements Planning, and Inventory Optimization. It provides a unified user interface and a single repository for all data. Its flexibility allows users to access data from external supply chain planning applications and make it available for reporting and analysis within a unified user interface based on Oracle Business Intelligence - Enterprise Edition (OBI-EE).

Oracle Service Parts Planning Implementation and User Guide

Oracle Service Parts Planning is used by repair service operations to ensure that the right parts are available at the right locations and at the right times, in usable condition. It allows planners to forecast and manage the distribution of individual parts in the most efficient manner possible.

Oracle Demantra Demand Management User Guide

This guide provides information on configuring and using Demantra Demand Management.

Oracle Demantra Implementation Guide

This guide provides information for a system administration who sets up the Oracle Demantra application. This guide also includes reference information on key database tables, client and server expressions, and configuring the various Demantra applications.

Do Not Use Database Tools to Modify Oracle E-Business Suite Data

Oracle **STRONGLY RECOMMENDS** that you never use SQL*Plus, Oracle Data Browser, database triggers, or any other tool to modify Oracle E-Business Suite data unless otherwise instructed.

Oracle provides powerful tools you can use to create, store, change, retrieve, and maintain information in an Oracle database. But if you use Oracle tools such as SQL*Plus to modify Oracle E-Business Suite data, you risk destroying the integrity of your data and you lose the ability to audit changes to your data.

Because Oracle E-Business Suite tables are interrelated, any change you make using an Oracle E-Business Suite form can update many tables at once. But when you modify Oracle E-Business Suite data using anything other than Oracle E-Business Suite, you may change a row in one table without making corresponding changes in related tables. If your tables get out of synchronization with each other, you risk retrieving erroneous information and you risk unpredictable results throughout Oracle E-Business Suite.

When you use Oracle E-Business Suite to modify your data, Oracle E-Business Suite automatically checks that your changes are valid. Oracle E-Business Suite also keeps track of who changes information. If you enter information into database tables using

database tools, you may store invalid information. You also lose the ability to track who has changed your information because SQL*Plus and other database tools do not keep a record of changes.

Introduction to Demand Signal Repository

Introduction

Oracle Demand Signal Repository (DSR) helps manufacturers collect detailed retailer Point of Sale and other demand data, analyze it to identify issues and opportunities, and respond using a Services Oriented Architecture (SOA)-enabled services library.

Oracle Demand Signal Repository supports most typical retail data sources at the day and SKU level of detail, including:

- Daily point-of-sale
- On-hand inventory at the store and retail distribution center
- Store orders and receipts
- Replenishment rules
- Retail distribution center withdrawals
- Returns
- Store promotions
- Sales forecasts

Demand Signal Repository transforms retailer data into manufacturer terms, including manufacturer item identification, manufacturer hierarchies, and calendars– so that direct retail data can be analyzed and reported on in either the manufacturer's terms or the terms of each individual retailer. . A high-performance data loading facility cleanses the data and allocates it to a uniform set of dimension levels.

The Demand Signal Repository data model is based on the Oracle Data Warehouse for Retail (ODWR) data model, adapted to a manufacturer point of view. Pre-built reporting and dashboard templates support category management and trading partner

score carding. Users can develop additional reports that combine data in customer-specific ways.

Demand Signal Repository provides the ability to share the external data with complementary applications with both Oracle applications such as Demantra Demand Management and non-Oracle applications. Demand Signal Repository offers direct, pre-built integration into Demantra as well as web services allow for data sharing with non-Demantra applications. These web services provide cleansed retail point of sale, on-hand inventory and other data at daily store level, or aggregated to a higher level as needed.

Business Intelligence Dashboard

When users log in, the Demand Signal Repository creates a seeded dashboard with the appropriate content and dimension levels based on their user roles. These dashboards can be further customized to display the data in which users are most interested.

Role-specific dashboards exist for:

- Account Team (Sales Manager, Customer Service Representative)
- Category Manager (Category Analyst, Marketing Manager, Executive Management)
- Supply Chain (Logistics Personnel, Demand Planner, S&OP Planner)

Each dashboard consists of related pages, which are organized using tabs. Each page corresponds to a single functional area (for example, category management, score carding, and so on). From within the dashboard you can report on any of the measures that apply.



Data

Demand Signal Repository works with demand data from many different sources. For example, the Demand Signal Repository collects direct retail data from multiple retailers as well as syndicated providers (where feasible). In addition, if a retailer provides competitor data to help support advanced category management processes, the Demand Signal Repository can be used by a manufacturer to capture and analyze competitor data as well as the manufacturer's own data.

Because data comes from a variety of sources (collectively called fact data), Demand Signal Repository must be validated and aggregated to an appropriate level. Market item data may apply to an entire market, rather than a specific retailer. It may also be expressed in terms of a brand, category or other higher-level item dimension nodes, at a different periodicity and an aggregated time period.

Oracle's Demand Signal Repository captures and manages the following retail fact data at the following levels of detail:

Data Type	Organization Level	Item Level	Time
Retail sales (POS)	Business Unit (store or DC)	SKU	Day (total)
Ordered quantity/amount	Business Unit (store or DC)	SKU	Day (total)

Backordered quantity/amount	Business Unit (store or DC)	SKU	Day (ending balance)
On-hand inventory	Inventory Location (e.g. back of store, shelf, display) within a Business Unit (store or DC)	SKU	Day (ending balance)
In-transit (inbound) inventory by expected arrival date	Business Unit (store or DC)	SKU	Day (ending balance)
Shipped quantity/amount	Business Unit (DC)	SKU	Day (total)
Received quantity/amount	Business Unit (store or DC)	SKU	Day (total)
Quality hold quantity/amount	Inventory Location (e.g. back of store, shelf, display) within a Business Unit (store or DC)	SKU	Day (ending balance)
Returned quantity/amount	Business Unit (store or DC)	SKU	Day (total)
Retail promotions	Business Unit (store or DC)	SKU	Day
Sales forecast	Business Unit (store or DC)	SKU	Day (total)

Measures

Measures are objective performance criteria that provide detailed insight into the demand situation of an organization. You can use the Demand Signal Repository's base measure values can be used to calculate common performance measures such as sales growth, gross margin, and inventory cover (days of supply), at the appropriate hierarchy levels. DSR Measures are organized into different categories for reporting and ease-of-use. You can use DSR's reporting capabilities to filter measures by category.

Some common measures cannot be calculated from the base data in the Demand Signal Repository data model, and are provided directly by the retailer or from some other external party in precalculated form. For these "external measures," Demand Signal Repository provides a data loading facility and database tables to store the values at the hierarchy levels at which they were received.

The following table outlines the measures that are available in Demand Signal Repository:

Measure	Category	Description	Unit of Measure
Monetary Sales Growth	Sales Growth	Percentage growth of monetary sales compared to prior period.	%
Unit Sales Growth	Sales Growth	Percentage growth of unit sales compared to prior period.	%
Alt UOM Sales Growth	Sales Growth	Percentage growth of alternative UOM sales compared to prior period.	%
% Product to Category Coverage by Value	Market Share	The sales of a single product as a percentage of total category sales - by \$ value.	%
% Product to Category Coverage by Volume	Market Share	The sales of a single product as a percentage of total category sales - by unit volume.	%
Market ACV by \$ value	Market Share	Total sales of all products across all categories in a specific market area. Provided by a syndicated data provider like AC Nielsen or IRI.	Currency
Market ACV by unit volume	Market Share	Total sales of all products across all categories in a specific market area. Provided by a syndicated data provider like AC Nielsen or IRI.	EA
% Category to Market Coverage by Value	Market Share	The sales of a single category as a percentage of total market sales - by \$ sales. ACV provided by syndicated provider like Nielsen or IRI.	%

% Category to Market Coverage by Volume	Market Share	The sales of a single category as a percentage of total market sales - by unit volume. ACV provided by syndicated provider like Nielsen or IRI.	%
Market Share (percentage of subcategory, category, department)	Market Share	One supplier's products sales as a percentage of total category sales.	%
Share of Market (percentage of geographical market)	Market Share	The retailer's sales as a percentage of total market sales.	%
Shopper Penetration	Market Share	The percentage of shoppers who purchased items in the scope of the measure.	%
Retailer Gross Margin	Gross Margin \$	Difference between the retailer's cost and the total sales price of items.	Currency
Retailer Gross Margin Increase	Gross Margin \$	Difference between profit in current period and profit in prior period.	Currency
Retailer Gross Margin %	Gross Margin %	Percentage difference between the retailer's cost and the average selling price of items.	%
Retailer Gross Margin Point Increase	Gross Margin %	The difference between the current gross margin percentage and the prior period gross margin percentage.	
GMROII (or GMROI)	Gross Margin %	Gross Margin Return on Inventory Investment: The retailer's return on inventory expenditures.	
% Contribution Sales Value to Category	Category Contribution	The sales of a single product as a percentage of total category sales - by \$ value.	%
% Contribution Profit to Category	Category Contribution	The retail profit of a selected item as a percentage of total category retail profit.	%

Average Price	Per Unit Performance	Point of sale receipts divided by unit volume sold.	Currency
Unit Profit	Per Unit Performance	Monetary difference between the retailer's cost and the average selling price of items.	Currency
Total Category Average Sales Value	Category Average Performance	Average per product sales amount for a specific product category - by \$ value.	Currency
Total Category Average Sales Volume	Category Average Performance	Average per product sales amount for a specific product category - by unit volume.	EA
Invoice Accuracy %	Invoice Accuracy	Percentage deviation of invoice amount from agreed price and received quantity, or percentage of invoices that fell within tolerance established.	%
Payment days	Timely Payment	Average number of days from invoice to payment.	Days
Late payment	Timely Payment	Average number of days payments were late.	Days
Number of Deductions	Deductions	Count of deductions taken.	
Deduction Amount	Deductions	Monetary amount of deductions taken.	Currency
Deduction Balance	Deductions	Monetary balance of outstanding deductions.	Currency
Item Data Accuracy %	Data Accuracy	Accuracy of the product data provided by supplier through the GDSN.	%
Item Data Synchronization %	Data Synch	The percentage of items that are synchronized via the Global Data Synchronization Network (GDSN).	%
Sales Forecast Error	Forecasting	The percentage error of the POS forecast vs actual sales for the period.	%

Sales Forecast Accuracy	Forecasting	The percentage accuracy of the POS forecast vs actual sales for the period.	%
Order Change %	Order Changes & Accuracy	% of orders or line items that had to be revised prior to shipment.	%
Perfect Order %	Inventory	The number of orders with no error, filled correctly for every line, on time and invoiced correctly divided by the total number of orders.	%
Inventory turns (annual)	Inventory	The value of annual shipments at cost (for the most recent period) divided by the total average daily inventory value at cost.	
Max Inventory cover	Inventory	Number of days of historical or forecast inventory consumption represented by quantity on-hand.	Days
Min Inventory cover	Inventory	Number of days of historical or forecast inventory consumption represented by quantity on-hand.	Days
Promotional Sell-Thru	Promotions	Percentage of promotional order that sold through during the promotional period.	%
Order Cycle Time	Order Cycle Time	The average number of days from when a PO is written until it is received.	Days
On-time Delivery (appointment date)	On-Time Delivery	Percentage of shipments delivered by appointment date.	%
On-time Delivery (appointment time)	On-Time Delivery	Percentage of shipments that were delivered within the appointment window on the appointment date.	%
Service Level / Fill Rate	Service Level	Amount ordered vs amount shipped.	%

% of Stores Scanning	Stocking	Percentage of store locations that sold a particular item.	%
% Not Scanning	Stocking	Percentage of store locations that did not sell a particular item.	%
In-stock %	Stocking	Percentage of stocking locations that had stock available when sampled.	%
Understock %	Stocking	Percentage of stocking locations that were below their minimum expected stock level when sampled.	%
Lost Sales	Stocking	Estimated value of sales lost due to out-of-stock conditions.	Currency
% Returns	Stocking	The percentage of product that the retailer deems unsaleable, as a percentage of cost or units sold .	%
Monetary Sales	Sales	Point of sale receipts.	Currency
Monetary sales transaction	Sales		
Unit Sales	Sales	Unit volume sold to consumers.	EA
UOM Sales	Sales	Unit of Measure	
Unit POS Forecast	Forecasting	Forecasted demand at point of sale in unit terms.	EA
Unit POS Forecast(Alt)	Forecasting		
Monetary POS Forecast	Forecasting	Forecasted demand at point of sale in monetary terms.	Currency
Monetary POS Forecast transaction	Forecasting		
On-hand Inventory	On-Hand	Ending balance of inventory at the storage location (units).	EA

On-hand Inventory(Alt)	On-Hand		
On-hand Inventory (retail basis)	On-Hand	Ending balance of inventory at the storage location (on a retail value basis).	Currency
On-hand Inventory (cost basis)	On-Hand	Ending balance of inventory at the storage location (on a cost basis).	Currency
Ordered Quantity	Orders	Quantity of items ordered (by expected receipt date).	EA
Ordered Quantity(Alt)	Orders		
Ordered Value	Orders	Value of items ordered (by expected receipt date).	Currency
Ordered Value (Txn)	Orders		
Shipped Quantity	Shipments	Quantity of items shipped (units).	EA
Shipped Quantity(Alt)	Shipments		
Shipped Value	Shipments	Value of items shipped (on a cost basis).	Currency
Shipped Value (Txn)	Shipments		
In-Transit Quantity	In-Transit	Quantity of items currently in-transit.	EA
In_Transit Quantity(Alt)	In-Transit		
Received Quantity	Receipts	Quantity of items received (units).	EA

Received Quantity(Alt)	Receipts		
Received Value	Receipts	Value of items received (on a cost basis).	Currency
Received Value (Txn)	Receipts		
Returned Quantity	Returns	Quantity of items returned (units).	EA
Returned Quantity(Alt)	Returns		
Returned Value	Returns	Value of items returned (on a cost basis).	Currency
Returned Value (Txn)	Returns		

Exception Measures

Exception measures count the number and value of exceptions from any of DSR's stored measures, which helps users to quickly identify problems from large amounts of data, and track the quantity and value of these exceptions. Each measure has a corresponding OBIEE iBot that updates the measure at user-defined intervals. Each seeded exception type in DSR has its own exception iBot. For example, an OBIEE iBot tracks exceptions for all Overstock measures.

The following table outlines the exception measures that are available in Demand Signal Repository:

Measure	Source	Loaded or Calculated Levels	Definition
Exception Count	Loaded	Type: Exception Time: Day Business Unit: Store Manufacturer Item: SKU	The total number of current exceptions. This measure increases by 1 for each daily occurrence of the exception.

Exception Quantity	Loaded	Type: Exception Time: Day Business Unit: Store Manufacturer Item: SKU	The number of DSR measures for which there is an exception. An exception is logged for each daily occurrence.
Exception Monetary Value	Loaded	Type: Exception Time: Day Business Unit: Store Manufacturer Item: SKU	The total monetary value (in \$ dollars) of all exceptions. This value is calculated daily.

Note: Exceptions are only calculated for previous time periods (either day or week). When the exception iBot runs, it does not calculate exceptions based on transactions with the current date. Similarly, iBots calculate weekly exceptions for the prior week, but do not calculate weekly exceptions for the current week. Weekly exceptions are dated as of the day at the end of the week for which the exception is calculated.

Forecast Accuracy is a weekly exception, and its exception iBot only needs to run once a week.

ACV Measures

Many Consumer Goods Manufacturers purchase consumption data from syndicated data sources such as AC Nielsen to analyze market penetration by retailer, geography, product and time. The set of syndicated consumption data includes pre-calculated ACV Measures as well as other measures such as sales by promotion type, baseline sales, and coupon sales. Syndicated consumption data can be purchased at various levels of the product hierarchy and customer organization hierarchy, as well as geographic region or sub-region.

These ACV measures can be loaded into Demand Signal Repository and associated to the appropriate dimensions. For more information, see Loading Syndicated Consumption Measures.

Goals and Thresholds

Demand Signal Repository provides the ability to load goals and thresholds at various levels of the organization and item hierarchy for a given time period (i.e., year, quarter, week, etc.) The main purpose of goals and thresholds are to determine whether

performance targets are being met.

Reports and Scorecards

Each dashboard includes pre-made reports for measures contained in that dashboard. These reports display data in multiple formats, including pivot tables and graphs. Users can drill down within reports to the level of detail in which they are interested.

In addition to the pre-made reports, users are also able to create new custom reports about the measures in which they are interested.

Category Management Reports

Welcome, ddr_manu! Dashboards - Answers - More Products - Settings - Log Out

Product To Category Coverage By Quarter

Product To Category Coverage By Period

Category Growth Analysis By Volume

Category Growth Analysis By Sales

Category Sales Volume By Type

Category Sales Value By Type

Retailer Profit By Category

Average Price Analysis

Top Performers By Profit

Bottom Performers By Profit

Top Performers By Sales

Bottom Performers By Sales

Page Options ▾

Year

Quarter

BY 2008 ▾

▾

Go

Product Category

Item

Item Type

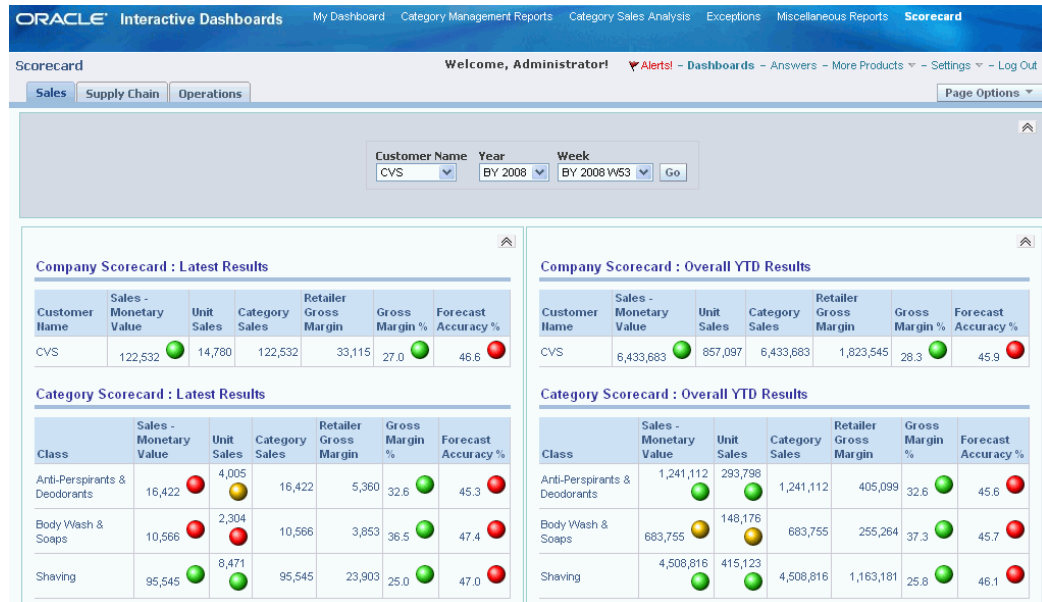
Channel

Go

Product To Category By Quarter

Quarter	Channel	Customer	Product Category	Manufacturer Item	%Product to Category By Unit Sales	%Change Same Period Last Year	%Product to Category By Monetary Sales	%Change Same Period Last Year
BY 2008 Q1	DC	DOLLAR MART	Carbonated Soft Drink	7-Up	1.70%		2.54%	
					2.01%		2.05%	
				A&W Diet Root Beer	3.58%		5.08%	
					2.20%		2.09%	
				A&W Root Beer	2.33%	-1.15%	2.10%	1.96%
					1.82%		2.56%	
				Barq's Root Beer	3.83%		5.12%	
				Caffeine Free Coke	0.85%		1.68%	
				Caffeine Free Diet Pepsi	0.94%		1.69%	
					2.26%	-4.69%	1.01%	-8.74%
				Canada Dry Diet Ginger Ale	3.65%		5.90%	
					2.20%		2.42%	
				Canada Dry Ginger Ale	3.77%		5.94%	
					2.26%		2.43%	

Scorecards use colors to indicate whether or not measures are in bounds or require attention. Scorecards allow users to monitor results over time and to manage by exception. You can customize the green, yellow, and red thresholds and drill down to view the supporting data.



Multiple Hierarchies

Demand Signal Repository represents data using both each individual retailer's and manufacturers' item and organizational hierarchies. For example, data received from a retailer is rolled up in terms of that retailer's hierarchies, and data from all retailers is rolled up in terms of the manufacturer's hierarchies.

Demand Signal Repository provides other dimension constructs that are available in both the retailer and manufacturer item hierarchies. Retailer and manufacturer SKUs can be organized by style, color, and size. You can also define customized item classification dimensions.

By default, retailer data is delivered at the SKU level. If data is provided at a higher retailer organizational level, it is allocated to the lowest (business unit) level.

Manufacturer Product Hierarchy

At the lowest level, the manufacturer's item hierarchy consists of manufacturer SKUs. All fact data pertaining to items is stored at the SKU level. Units that share a SKU can be considered to be identical. Above the SKU is the item. Items collect all of the variations of a single logical product together into one entity. For example, apple juice is an item that may have many packaging variations—six-pack, twelve-pack and sizes—500-ml, 12-oz, and so on. All SKUs within an item share the same brand.

Manufacturer items are grouped into subclasses and within a group in a company division.

Retailer Product Hierarchy

Every retailer in Demand Signal Repository has its own item hierarchy. The retailer's hierarchy begins with the retailer SKU, which has a many-to-one association with a manufacturer SKU, which is linked via the Global Item ID. Like the manufacturer hierarchy, the retailer item hierarchy collects SKUs into items; this level is important to fashion retailers but is often not used by grocery retailers (resulting in one item per SKU). Retailer items are rolled into subclasses (such as organic fruit spreads) and then into categories (such as jam and jelly) and then into departments (such as grocery).

Retailer Organization Hierarchy

All data except on-hand inventory data is stored at the business unit level, which corresponds to an individual store or retailer distribution center. Inventory may be captured at multiple inventory locations within a store or distribution center, which can be aggregated to the business unit level.

Business units in Demand Signal Repository have a single address, so each business unit can be viewed in terms of the logical business organization of a particular retailer (such as Atlantic sales region) or in geographical terms (such as the city, state or country in which that store appears). These dimensions can be used in combination to further refine the selection of business units for reporting or analytical purposes.

Retailer stores and distribution centers are organized into districts, regions, and areas. At the top level of the organization is the chain, which could be a banner. Chains can be amalgamated into retail groups that own multiple banners. The Demand Signal Repository also allows customers to organize each retailer's business units by channel, market area, trade area, or all three, as well as by specified demographic features.

Geography Hierarchy

Geography can be used to classify retail locations across multiple retailers. In addition to the political geography of city, state, and region, manufacturers can organize locations by their own sales regions and subregions, or by the regional scheme provided by a third party.

Clusters

Demand Signal Repository allows you to group like items and stores into organizational units called Clusters. By grouping customers, stores and items, users are able to execute strategies related to specific demographics, store formats, consumer demographics, weather zones, or other characteristics. Clusters are only logical groupings, and items/stores may belong to more than one cluster at a time. For example, For all stores located along the coast may be assigned to a store cluster called "coastal". Sales of sunscreen are analyzed by cluster, which compare sales at coastal stores to overall sunscreen sales. Some of stores that belong to the "coastal" store cluster

may also belong other store clusters. For example, some of the stores may also be associated with a store cluster called "upscale urban" which describes the consumer demographics of the surrounding area.

Retail, Item and Store cluster definitions are loaded into DSR using the existing reference data load process. For more information, see "Data Loading Interface Tables" for more information.

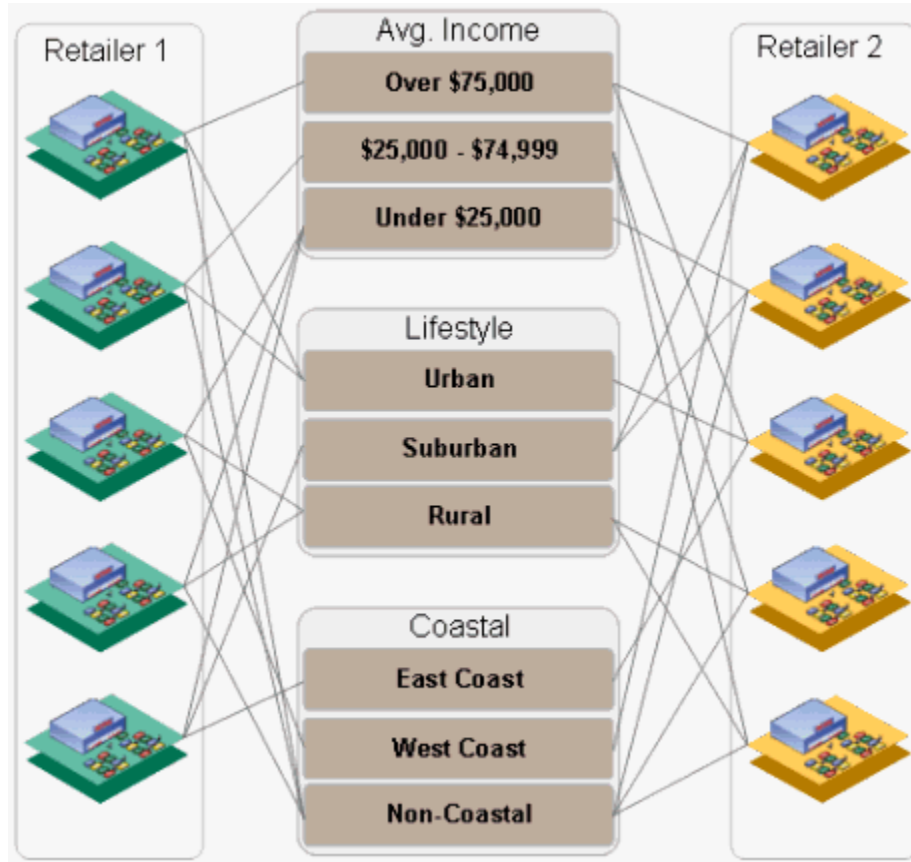
Retail Clusters

Retail clusters are the grouping of similar retailers at the organization level. For example, you may create retail clusters named Department Store, Drug Store, and so on.

Store Clusters

Consumer goods manufacturers commonly cluster stores with similar properties together to execute strategies related to specific demographics, store formats, consumer demographics, weather zones, or other characteristics. Demand Signal Repository allows users to group business units (either stores or distribution centers) into smaller, more manageable clusters.

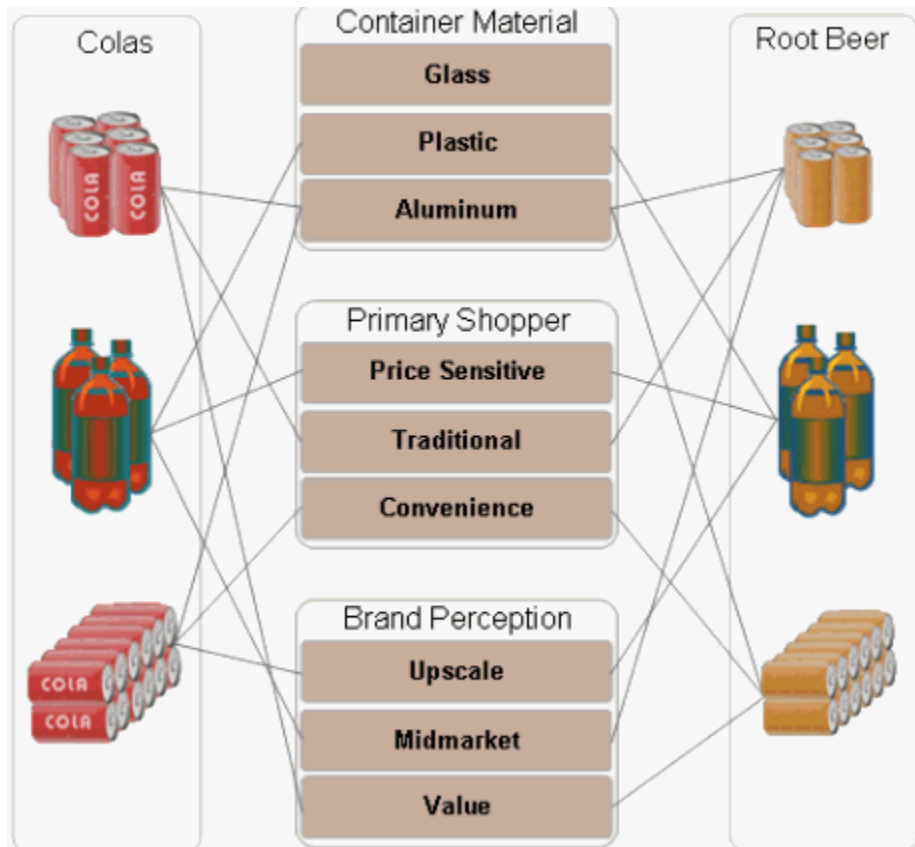
The example below illustrates how stores from different retailers can be assigned to store clusters based on average income, lifestyle and geographic location. A store cluster report is available to analyze sales by store cluster type (for example, Average income, Lifestyle, Coastal) to see breakdown of sales by each unique cluster.



Item Clusters

Consumer goods manufacturers commonly cluster similar items together to evaluate sales strategies and promotional activities. Demand Signal Repository has the ability to group items into item clusters at the item subclass (or higher) level in the item dimension. These item clusters are available at the SKU level.

In the example below, SKUs from different product categories are assigned to item clusters based on average container material, primary shopper and brand perception:



Calendars

Demand Signal Repository stores all data is stored in Demand Signal Repository at the day level. All calendar dimensions use days as the lowest level of detail, so Demand Signal Repository data can be represented in terms of any calendar view. Data that is received in weekly buckets must be allocated to the day level in order to be represented in terms of the alternative calendars.

Demand Signal Repository users must be able to review each retailer's data in terms of the associated retailer's business calendar. They also must be able to roll it up according to the Gregorian calendar and the manufacturer's business, fiscal, advertising, and planning calendars. Demand Signal Repository supports the calendars typically used in an organization: Gregorian, business, fiscal, planning, and advertising.

Gregorian Calendar

This calendar is common to all organizations and is pre-seeded in the system. Calendar days are divided into standard calendar months (January, February, and so on) according to the Gregorian system. Calendar months are then rolled up into calendar years. Separately, calendar days are associated with calendar weeks, which do not fall precisely into calendar month boundaries. During configuration, the customer selects

which day, typically Sunday or Monday, will begin their calendar week.

Business Calendar

The business calendar rolls up daily activity into business weeks, which are the most common way to review retail data.

Each retailer in Demand Signal Repository—along with the manufacturer—has its own business calendar that reflects differences among organizational practices. Some retailers align their business week to a calendar week (for example, Sunday to Saturday); others start the business week at other times (for example, Thursday) so that peak shopping activity through the weekend falls in the same week.

Business weeks are rolled up into business months. Because business months must align with business week boundaries, they have different numbers of days than calendar months: either 28 (four weeks) or 35 (five weeks). In turn, business months roll up to business quarters, which are usually closely aligned with the company's fiscal quarters. Each quarter contains two months, each with four weeks, and one month with five weeks, to yield a uniform 13-week ($4 + 4 + 5$ weeks) quarters. Finally, business quarters roll up into business years, which may begin at a different time of year than the Gregorian year.

The typical business year of four 13-week quarters is 364 days. Every five years or so, a business year must be extended to 53 weeks so that business months continue to approximate Gregorian months. The week is added to one of the four-week months, resulting in a 14-week quarter.

Fiscal Calendar

The fiscal calendar, which is intended for financial reporting, is exclusive to the manufacturer. Usually, calendar days roll into calendar months much as in the Gregorian calendar, but fiscal quarters and years may not align with the Gregorian year.

Planning and Advertising Calendars

Planning and advertising calendars, which provide additional review flexibility, are also exclusive to the manufacturer.

The planning calendar reflects the need to roll up data into demand planning time buckets, which may align with production schedules (for example, a Monday–Sunday production week versus a Sunday–Saturday calendar week). The advertising (ad) calendar, meanwhile, aligns with ad (advertising) weeks.

Both the planning and ad calendars have the same structure. Weeks are collected into periods, which may or may not approximate months. For example, some manufacturers prefer a regular pattern of 13 four-week planning calendar periods. Periods can then be rolled into quarters and years.

Data Allocation and Aggregation

When point-of-sale and sales forecast data is imported to Demand Signal Repository, it may be received in an aggregate time or organizational dimension (for example, weekly, chain-level data). Before the Demand Signal Repository can use this data, it must be allocated to a daily, business-unit level (either store or distribution center values). Allocation to a uniform level allows Demand Signal Repository to aggregate the data according to alternate hierarchies and calendars. Customers can provide a profile that gives the proportions of the aggregate values to be allocated to each component time bucket or location. In the absence of a profile, each value will be the total divided by the number of component data values.

While allocation is a common technique in planning applications, some manufacturers want to prevent the allocation of historical data because it can provide the illusion of detailed data that is an estimate, not an actual value. If a manufacturer collects data from a particular retailer at the chain level, users have the option of mapping the data of the entire chain to a single business unit node, as if it were a store; thereby avoiding allocation in the geographic dimension. Similarly, if all data is weekly, the customer could define a time-based allocation profile that maps all weekly data to a single day of the week.

Alternate Units of Measure

Demand Signal Repository provides an alternate unit of measure that customers can use for aggregation, analysis, and reporting. Data values are stored in terms of both the primary unit and the alternate unit.

The primary unit for measure for all items is assumed to be uniform (typically, each). If primary units of measure differ among items, you should not aggregate items together in their different primary units of measure.

The alternate unit of measure provides a uniform basis of comparison across diverse pack sizes or product categories. For example, a beverage company might select volume in liters as their alternate unit of measure. A company that sells flour might use weight (pounds); some companies use an expected number of consumer uses per package. Each SKU has a ratio to convert the primary unit of measure quantity to the alternate unit of measure quantity.

Currencies

Demand Signal Repository supports two types of currency: reporting currency and transaction currency.

Reporting currency is user-defined and is used for the conversion, aggregation, and viewing of data. You can also view source data in its original transaction currency. Data is stored in both the transactional and reporting currencies for every measure that is a monetary value. By default, the dashboards will use the reporting currency, and you

can customize the dashboards to replace the reporting currency measures with the transaction currency.

Data Model

Data Model Overview

The Demand Signal Repository data model is based on the Oracle retail data model but is designed to meet manufacturer's requirements. Users can view data from either a manufacturer or retailer perspective. For example, sales can be viewed using the manufacturer's item hierarchy, item numbers and descriptions, or from the retailer's item hierarchy, using the retailer's item classification method and the retailer's item numbers and descriptions. Slowly changing data, like organizations and items, is called reference data. Data that contains quantities, amounts, or other types of "facts" is called fact data.

Fact Data is stored by store, SKU and Day. This low level of granularity allows users to "slice and dice" information by time or organizational dimensions. In cases where a retailer submits data at an aggregate level, for example, by week or by a group of stores, DSR will allocate data based on a pre-defined organization allocation or time-allocation pattern.

The DSR Data Model retains historical changes to dimensional (aka hierarchical) data, such as retailer organization hierarchy, or item classification hierarchy. This feature, called Slowly Changing Dimensions (SCDs), allows users to analyze data according to the hierarchy in effect at the time.

Organization and Business Units

Within DSR, an organization generally equates to a company. Organization types supported by DSR include:

- The manufacturer (which is company that owns DSR)
- Customers (also called retailers or distributors)
- Competitors (which are used to analyze all sales within a category including

competitor products)

DSR maintains a customer organizational hierarchy to allow customer data to be summarized based on the customer's view of their organization. Customers with common characteristics can be grouped into Retail Clusters for further analysis.

A Business Unit is a single customer location. Each business unit has one physical address. This enables the Business Unit to be analyzed by a geographic dimension, as well as the organization dimension. While business unit types are user-defined, the two most common types of business units are:

- Stores
- Customer distribution centers

Each business unit in DSR must have a unique ID. Typically this is a GLN (Global Location Number) or a D&B Duns Plus 4 number.

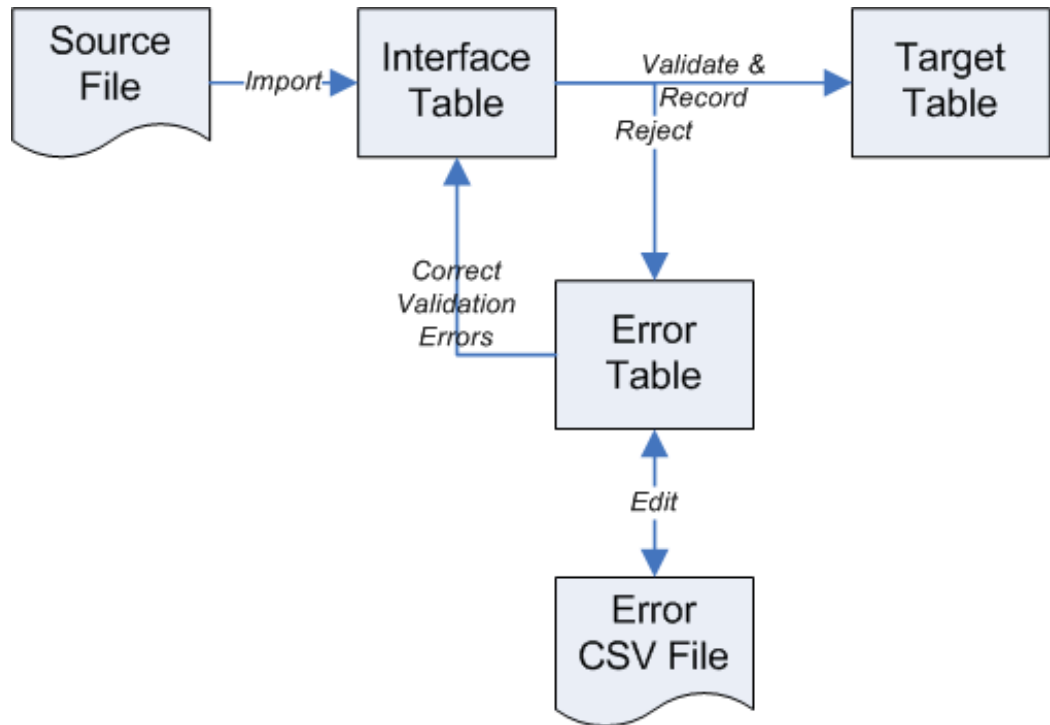
Customer Organization Hierarchy

Retailer stores and distribution centers are organized into a customer organization hierarchy that includes districts, regions and areas. A retail group is an organization that owns one or more banners, or chain stores. For example, Wal-Mart's "Neighborhood Market" chain versus its "Supercenter" chain.

In addition to organization hierarchy, DSR also allows customers to organize each retailer's business units by channel, as well as by demographic features. Geography can be used to classify retail locations across multiple retailers. In addition to the geography of city, state and region, manufacturers can organize locations by their own sales regions and sub-regions, or by the regional scheme provided by a third party.

Reference Data Load Process

Illustrated here is the method by which reference data is loaded into DSR:



The source file can either come from the customer (for example, retailer items), or generated by the manufacturer (for example, manufacturer items). The source file is loaded into a standard interface table, where an Oracle Warehouse Builder process flow validates the data in the interface table.

If the data passes validation, it is moved from the interface table to the associated target table (or tables) in DSR. If the data does not pass validation, the associated record (or records) in the interface table(s) are moved to an error table. The error table contains the original source data, plus a column that lists the error (or errors) encountered. In addition, a related CSV file is created so that the errors can be imported into a spreadsheet (or other) application and corrected, if desired.

Some of the common validations that are commonly performed against reference data include:

- **Basic formatting** - Validating completeness, to ensure required fields have been loaded and that dates are valid.
- **Code validation** - Unit of Measure, Currency Codes, Types of Business Units (which are user defined).
- **Duplicate records** - The system can be configured to error if duplicate records exist in the interface table. For example, if the same manufacturer item exists twice (or more) in the interface table. If the system is configured to ignore duplicate records, then the last record processed overlays any previous records.

- **Simulation or Partial Load** - A system parameter that controls how reference data is loaded. In simulation mode, the data in the interface tables are validated and any errors encountered are written to the error table (and CSV file); however, the valid records are not written to the target table (because this is just a simulation). This feature is useful when first implementing DSR. In partial mode, the system works like we described earlier. Valid records are written to the target table, while records in error are written to the error table.

Reference Data Process Flows

For each data set there is a main process flow which processes data from the interface table to the target table, as well as two additional process flows to move data from the error table (or .CSV error file) back to the interface table once errors have been corrected. The main process flows can be scheduled to run automatically using Oracle Warehouse Builder's scheduling feature, or a script can be created which executes the process flow as part of multi-step process.

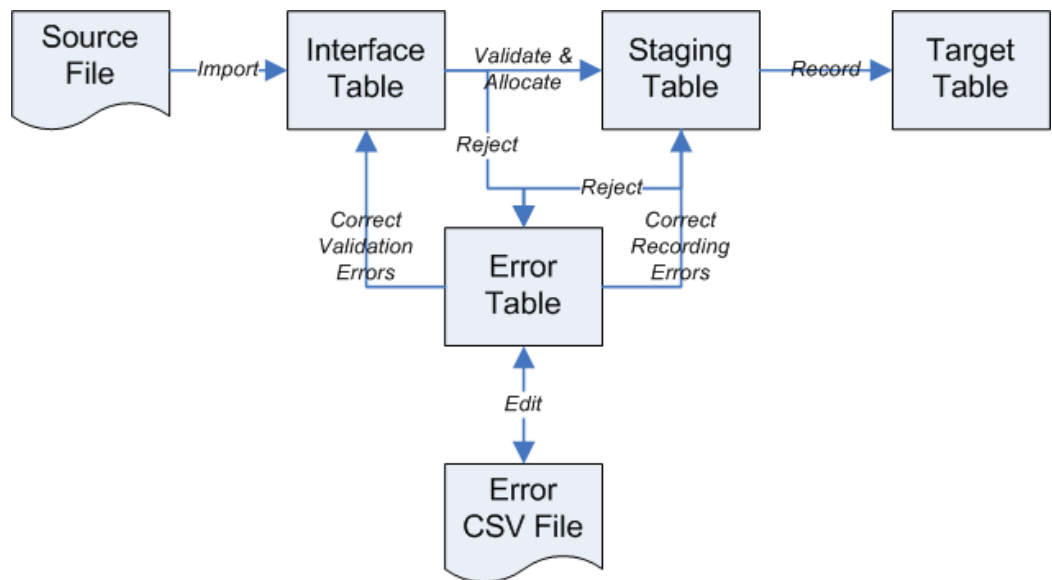
The available DSR reference data process flows are:

- BUNIT_HDAY
- CAUSAL_CNDTN
- CRNCY_EXC_RATE
- GEOGRPC_DEMOGRPY
- GEOGRPY_RGN_HCHY
- GOALS_THD
- ITM_BSNS_UNT_ASG
- ITM_CLUSTER
- ITM_UOM
- LOOKUP_DATA
- MARKET_AREA
- MARKET_AREA_BSNS_UNIT
- MFG_ITM_HCHY
- ORG_HCHY
- RTL_CLUSTER

- RTL_ITM_HCHY
- SKU_BSNS_INV_RULE
- STR_CLSTR
- SYND_GEO
- SYND_PROD
- TDLINX_ORG_HCHY

Fact Data Load Process

Illustrated here is the method by which fact data is loaded into DSR:



The process works similar to loading reference data, but once the fact data passes validation it is moved to a staging table. Only after all fact records associated with a process flow are loaded into the staging tables are they copied to the target tables.

If Fact Load Method (a system parameter) is set to Partial, then all records that passed validation are moved from the staging table to the target table. If Fact Load Method is set to All or None, then all records must pass validation before a single record is written to the target tables.

Some of the common validations that are commonly performed against fact data include:

- **Basic Formatting** - Validating completeness, to ensure required fields have been loaded and that dates are valid.

- **Code Validation** - Unit of Measure, Currency Codes, Types of Business Units (which are user defined).
- **Unrecognized Items or Locations** - This includes validating retailer item number and the existence of a global item number that maps to an item in the manufacturer's item hierarchy, and validating locations (for example, stores, distribution center, or other level in the organization hierarchy).
- **Ambiguous Items** - This occurs when the retailer provided data solely in terms of its local item number, and there is more than one Global Item Number (GTIN, or other item ID type) that maps to that local item.
- **Unauthorized or New Items** - This occurs when the item/business unit combination is not set up in DSR, and item discovery is turned off.
- **Missing Allocation Pattern** - This occurs when data was loaded at an aggregate level, but no allocation pattern was defined for the associated organization and/or week.

Fact Data Process Flows

For each data set (for example, sales and returns data) there is a main process flow which processes data from the interface table to the target table, as well as two additional process flows to move data from the error table (or .CSV error file) back to the interface table once errors have been corrected.

Like reference data process flows, the main process flows can be scheduled to run automatically using Oracle Warehouse Builder's scheduling feature, or a script can be created which executes the process flow as part of multi-step process.

The available DSR fact data process flows are:

- FORECAST_EF_TO_INF_STG
- FORECAST_ERR_TO_INF_STG
- FORECAST_INF_STG_TO_TGT
- INVNTORY_EF_TO_INF_STG
- INVNTORY_ERR_TO_INF_STG
- INVNTORY_INF_STG_TO_TGT
- ORDERS_EF_TO_INF_STG
- ORDERS_ERR_TO_INF_STG

- ORDERS_INF_STG_TO_TGT
- PROMOPLN_EF_TO_INF_STG
- PROMOPLN_ERR_TO_INF_STG
- PROMOPLN_INF_STG_TO_TGT
- SHIPMENT_EF_TO_INF_STG
- SHIPMENT_ERR_TO_INF_STG
- SHIPMENT_INF_STG_TO_TGT
- SLS_RTRN_EF_TO_INF_STG
- SLS_RTRN_ERR_TO_INF_STG
- SLS_RTRN_INF_STG_TO_TGT

Users and Security

User and Security Overview

In Demand Signal Repository, administrators can limit user access to any of the following:

- Organizational and Item hierarchies: users may be restricted to one or more subhierarchies; or may be limited to only viewing aggregate (cross-retailer) data.
- Measures: users may be restricted to see only a subset of the available base or calculated measures.
- Dashboards and Reports: users may be restricted to only a subset of dashboards and reports.

Access control to hierarchies, measures and dashboards can be controlled on a per-user level, or by user role or group. Depending upon their role, some users may be restricted from accessing the scorecard dashboard. Users may also be restricted from accessing specific reports within the category management dashboard.

User Roles

User access to data in Demand Signal Repository is dependent upon the legal agreements that the manufacturer has in place with its customers, and internal policies intended to prevent leakage of information to competitors. The access profile differs for each of the major user roles in the system.

Each user role can be tied to a specific dashboard and associated reports with appropriate content and appropriate dimension levels. Demand Signal Repository supports the following user roles:

- Customer Account Team
- Category Management Team

- Supply Chain Management Team
- Administrator

Customer Account Team

The Demand Signal Repository Customer Account team is responsible for reviewing the detailed financial data for its assigned customer (or customers). This includes data of all types – sales, inventory, shipments and receipts. The Customer Account team is also able to review competitor data provided by the customer for the purpose of managing the overall retail category ("category captain" role). However, they are not allowed to see the data from any other customer. Generally, each account team's access is limited to nodes at or below their assigned customers' organization level in the organization hierarchy.

It is possible that access may be further restricted in the case of large customers. Users may only be able to see data for a single chain or geographic region; they may also be limited to one or more of the retailer's departments or item categories.

Category Management Team

The Demand Signal Repository Category Management team is responsible for identifying product trends across accounts to determine what products to offer, how to price them and how to promote them. Category managers work in terms of the manufacturer's item hierarchy, rather than the retailer item hierarchy that the account teams primarily use.

Based upon the data access restrictions in customer legal agreements, category managers may be restricted from seeing detailed data within a particular account. They often may only see aggregate data at chain level. SKU-level accesses to a manufacturer's own items, rolled up to the chain level, typically would not be restricted. However, category managers usually do not have access to the competitor data provided directly by customers. They typically source competitor data from market data sources (3rd party syndicators) instead.

Depending upon the manufacturer's internal company policies, category managers may also be restricted from seeing data for classes of product that they are not directly responsible for. This would require their access to be restricted to a subhierarchy (or subhierarchies) in the manufacturer's item hierarchy.

Supply Chain Management Team

The Demand Signal Repository Supply Chain Management team is responsible for demand planning and monitoring the supply chain for any inventory issues (for example, overstocks and out-of-stocks). However, supply chain personnel do not typically need access to financial data.

Administrator

The Demand Signal Repository Administrator is responsible for administering users, controlling access, and other security-related tasks. Administrators are able to freely access data across all hierarchy dimensions.

Dashboards and Scorecards

Category Sales Analysis Dashboard

The Category Sales Analysis dashboard measures:

- Category profitability, across time, channel, customer and at the brand and item levels.
- Average price, which compares the price of the product at retail against the competitive products at an item level by channel, time and market.

Supported Category Sales Analysis measures include:

Measure	Explanation
Retailer Net Cost and Gross Margin	Displays the net cost and gross margin for each product category, by Retailer.
Category Coverage	Displays the sales of a single product as a percentage of total category sales - by \$ value.
Product Mix	This pie chart displays the breakdown of categories, brands or items. Click on a category within the pie chart to drill-down to its members. For example, clicking on Tea drills-down to a pie chart displaying all products belonging to the category.
Product Contribution to Gross Margin	Display the percentage that each product category contributes towards gross margin.

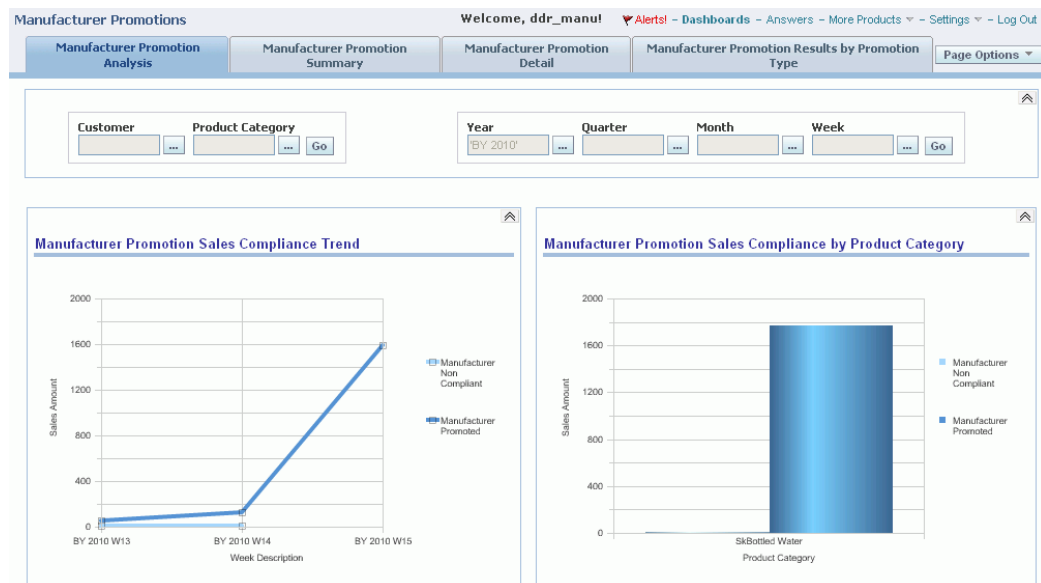
Category Sales By Type

Category sales value by type of promotion – either promotion or regular pricing. This measure helps category managers see how much his sales organization depends upon promotion to sell, and how it affects the overall sales and volume at retail.

Sales Growth % Change to Previous Period

Percentage change in sales growth for the category over the previous period. For example, selecting the Month filter displays the percent change from the previous month to the current month.

Manufacturer Promotions Dashboard



Use the Manufacturer Promotions dashboard to display pre-seeded reports and graphs to facilitate reporting and analysis. This dashboard displays:

- Weekly sales compliance trend by promoted and non-compliant sales
- Manufacturer promotion compliance by promotion type
- Manufacturer promotion compliance by product category for promoted and non-promoted sales by product category
- Expected vs. Actual promotion sales
- Expected vs. Actual ACV % by manufacturer promotion

- Expected vs. Actual sales by promotion type (for example, display, feature, and so on.)

As well, context sensitive drill-downs are available from the manufacturer promotion dashboard to display:

- Manufacturer Promotion Summary Results at the promotion/customer level
- Manufacturer Promotion Detail Results at the promotion/customer/SKU level

The Manufacturer Promotions Analysis tab displays a variety of charts for compliance analysis including:

- Mfr Promotion Sales Compliance Trend
- Mfr Promotion Sales Compliance by Product Category
- Mfr Promotion Sales (Expected vs. Actual)
- Mfr Promotion ACV % (Expected vs. Actual)
- Mfr Promotion Sales by Promotion Type (Expected vs. Actual)
- Mfr Promotion Sales Compliance by Promotion Type

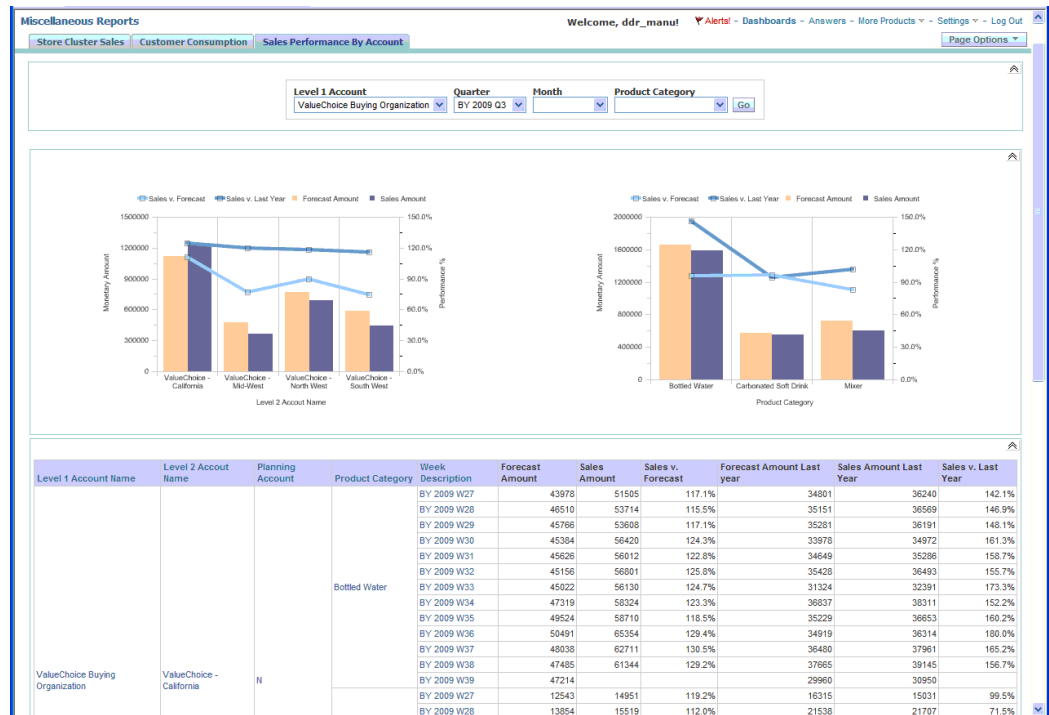
The Manufacturer Promotions Summary tab displays a high level overview of promotions:

- Mfr Promoted Sales Quantity
- Mfr Non-Compliant Sales Quantity
- Total Sales Quantity
- Expected Sales Quantity
- % of Expected Sales Qty
- Expected Customer ACV%
- Actual Customer ACV%
- Number of Stores Running Promotion
- Number of Retail Stores
- Store Participation

The Manufacturer Promotions Detail tab displays detailed promotions data by promotion name and type:

- Promoted Price
- Actual Promoted Price
- Actual Mfr Non-Compliant Price
- Mfr Promoted Sales Quantity
- Mfr Non-Compliant Sales Quantity
- Total Sales Quantity
- Expected Sales Quantity
- % of Expected Sales Qty
- Expected Customer ACV%
- Actual Customer ACV%
- Number of Stores Running Promotion
- Number of Retail Stores
- Store Participation %

Sales Performance by Account Dashboard



Demand Signal Repository has provided a sample out of the box dashboard which utilizes the Alternate Organization Hierarchy. Existing dashboards and reports can be easily modified to use the alternate organization hierarchy instead of the Customer Organization Hierarchy.

The Sales Performance by Account dashboard page can be found on the Miscellaneous Reports dashboard. This dashboard displays sales and forecast facts associated with the ValueChoice Buying Organization (a level one account). The chart on the left shows a breakdown by level two account and the chart on the right shows a breakdown by product category.

The Sales Performance by Account dashboard page is made up of:

- The Sales Performance by Account Prompt
- The Sales Performance by Account query which has several views that are displayed on the dashboard page.

Scorecards

Demand Signal Repository uses color-coded scorecards which provide a graphical interface for monitoring results versus goals over time. The scorecard indicates how well you are meeting your goals and thresholds, and display both latest (weekly) results

as well as overall Year To Date (YTD) results. The following example illustrates how color-coding indicates results:

Company Scorecard: Overall						
Customer Name	Sales - Monetary Value	Unit Sales	Category Sales	Retailer Gross Margin	Gross Margin %	Forecast Accuracy %
Bigmart	900,769 	451,856 	814,720 	147,828 	16.41 	
Dollar Mart	675,572 	338,954 	611,053 	110,884 	16.41 	
Goodway	2,054,798 	1,063,038 	2,172,134 	401,998 	19.56 	
ValueChoice	1,801,484 	903,851 	1,629,411 	295,686 	16.41 	

Demand planning involves establishing performance targets that measure many aspects of a single customer relationship. Then, on a weekly, monthly, or quarterly basis, the scorecard evaluates whether the company has met the goal or has fallen below its objectives.

When drilling down on a measure (either latest results or overall results) from the scorecard, the underlying reports only display weekly results for the specified year. All available weeks are displayed, regardless of the week selected in the scorecard. The results are displayed starting with the first week in which data is loaded through the latest week.

Upper and Lower Bound Measures

How DSR color-codes Scorecard measures depends on whether or not a measure is upper bound (in which lower values are better), or lower bound (in which higher values are better). The following table lists all upper and lower bound measures:

Lower Bound Measures	Upper Bound Measures
Monetary Sales	On-Hand Inventory
Unit Sales	% Returns
Category Sales	Inventory Cover
Retailer Gross Margin	Order Cycle Time (OCT)
Gross Margin %	Order Change %
Forecast Accuracy %	Deduction Balance
On-Time Delivery %	Payment Days

Perfect Order %

Invoice Accuracy %

Item Data Synchronization %

Item Data Accuracy %

Formatting Rules for Lower Bound Measures

Each measure has an associated icon, that is colored to represent a different condition:

Measure (Icon) Appearance	Explanation
No Icon	No goal has been defined for the measure.
Green Circle	Sales are equal to (or above) the goal.
Yellow Circle	The measure is below the specified goal, but above the lower-bound threshold.
Red Circle	Either: <ul style="list-style-type: none">• The measure is below the specified goal and above the lower-bound threshold.• The measure is below the specified goal, and no threshold has been set.

Formatting Rules for Upper Bound Measures

Each measure has an associated icon, that is colored to represent a different condition:

Measure (Icon) Appearance	Explanation
No Icon	No goal has been defined for the measure.
Green Circle	The measure is below or equal to the goal.

Yellow Circle	The measure is above the goal, but below
Red Circle	Either: <ul style="list-style-type: none"> • The measure is above the specified goal, and below the upper-bound limit. • The measure is above the specified goal, and no threshold has been set.

Sales Scorecard

The Sales scorecard displays the following sales-related measures.

- Monetary sales: Sales amount for all of the manufacturer's products in terms of the manufacturer's reporting currency rolled up to chain level versus the goal set for the most recent week or month.
- Unit sales: Sales quantity for all of the manufacturer's products rolled up to chain level versus the goal set for the most recent week or month.
- Category sales: Sales amount for all products (including competitor products) in terms of the manufacturer's reporting currency rolled up to chain level versus the goal set for the most recent week or month.
- Category share: Ratio of monetary sales to category sales as a percentage compared to the goal set for the most recent week or month.
- Retailer gross margin: Retailer gross margin for all of the manufacturer's products in terms of the manufacturer's reporting currency rolled up to chain level versus the goal set for the most recent week or month.
- Gross margin %: Ratio of retailer gross margin to monetary sales as a percentage compared to the goal set for the most recent week or month.
- Forecast accuracy %: Mean absolute percentage deviation of the forecast from actual monetary sales for all manufacturer products across the retail chain.

Supply Chain Scorecard

Supply Chain scorecard displays the results of measures versus goals comparisons for seven supply-chain-oriented measures:

- Service Level: The percentage of product that a buyer received, as compared to the original ordered quantity.

- **Order Cycle Time:** The average time between placing an order and when the order was received at the customer location.
- **Inventory Cover:** How many days of demand that can be filled by the current inventory.
- **On-Time Delivery:** The percentage of customer orders that are received on-time.
- **On-Hand Inventory:** The quantity of current inventory.
- **In-Stock %:** The percentage of items that are currently on-hand (in stock).
- **% Returns:** The percentage of customer orders that are returned.

Operations Scorecard

The Operations Scorecard displays the results of measures versus goals comparisons for the following operations-related measures:

- **Perfect order %:** The percentage of customer orders that do not require changes in items or item quantities.
- **Order change %:** The percentage of customer orders that require changes in items or item quantities.
- **Invoice accuracy %:** The percentage of seller invoices that are accurate.
- **Payment days:** The average number of days that pass between a customer receiving an invoice and paying it.
- **Deduction Balance:** The monetary value of products removed from distribution because of damage, seasonality, discontinuation, and so on.
- **Item data synchronization %:** The percentage of items sold that are synchronized by the buyer using the Global Data Synchronization Network (GDSN).
- **Item data accuracy %:** The percentage of items with accurate (that is, measured) dimensions and weight attributes out of all items.

Reports

Use Demand Signal Repository's prebuilt Category Management reports to view filterable demand data. Each report is built to display a different aspect of your organization's performance.

Category Management Reports

Welcome, ddr_manul

Dashboards - Answers - More Products - Settings - Log Out

Product To Category Coverage By Quarter

Product To Category Coverage By Period

Category Growth Analysis By Volume

Category Growth Analysis By Sales

Category Sales Volume By Type

Category Sales Value By Type

Retailer Profit By Category

Average Price Analysis

Top Performers By Profit

Bottom Performers By Profit

Top Performers By Sales

Bottom Performers By Sales

Page Options

Year

Quarter

BY 2008

Go

Product Category

Item

Item Type

Channel

Go

Product To Category By Quarter

Quarter	Channel	Customer	Product Category	Manufacturer Item	%Product to Category By Unit Sales	%Change Same Period Last Year	%Product to Category By Monetary Sales	%Change Same Period Last Year
BY 2008 Q1	DC	DOLLAR MART	Carbonated Soft Drink	7-Up	1.70%		2.54%	
					2.01%		2.05%	
				A&W Diet Root Beer	3.58%		5.08%	
					2.20%		2.09%	
				A&W Root Beer	2.33%	-1.15%	2.10%	1.96%
					1.82%		2.56%	
				Barq's Root Beer	3.83%		5.12%	
				Caffeine Free Coke	0.85%		1.68%	
				Caffeine Free Diet Pepsi	0.94%		1.69%	
					2.26%	-4.69%	1.01%	-8.74%
				Canada Dry Diet Ginger Ale	3.65%		5.90%	
					2.20%		2.42%	
				Canada Dry Ginger Ale	3.77%		5.94%	
					2.26%		2.43%	
				Cherry 7-Up	4.16%		4.77%	

You can filter reports by time (Year, Quarter, and Month) or product (Product Category, Channel, Item Type). Click a report element (for example, the Bottled Water category or Dollar Mart customer) to drill down to a detailed display of that particular element. Where appropriate, the user can also choose to see demand data related to the manufacturers items only, competitors' items only or all items.

The following table describes the prebuilt reports that are available in Demand Signal Repository:

Report Name	Description
Product to Category Coverage by Quarter	<p>This report displays, by customer, the individual items and their subtotals by customer within category.</p> <p>Data is subtotaled by item within customer, customer within channel type, and grand total for all channels.</p> <p>You can filter this report by:</p> <ul style="list-style-type: none"> Time (Year, Quarter) Item (Product Category, Item, Item Type) Geography (Market Area, Channel)

Product to Category Coverage by Period

This report displays, by customer, category coverage by quarter, month and week.

You can filter this report by:

- Time (Quarter, Month, Week, and Day)
- Item (Product Category, Item, Item Type, Channel)
- Geography (Market Area, Channel)

Category to Market Coverage by Quarter

This report displays the category activity on a market basis, sorted first by the category within the market, and the customers as a subset of the category. This report shows both value (money) and volume (units sold).

You can filter this report by:

- Time (Year, Quarter)
- Item (Product Category, Item, Item Type)
- Geography (Market Area, Channel)

Category Growth Analysis by Volume

This report displays sales growth based on overall volume by product category. This report includes the percentage change in sales versus the same period last year, and the total year-to-date percentage change.

The color-coded threshold is user-defined and enables you to quickly identify under performing categories. Click the color bar to drill down to the report details. For example, clicking the color threshold KPI indicator drills down to the brand report:

- Green indicates positive growth.
 - Yellow indicates zero growth.
 - Red indicates negative growth.
-

Brand Growth Analysis by Volume	<p>This report displays sales growth based on overall volume by brand. This report includes the percentage change in sales versus the same period last year, and the total year-to-date percentage change.</p> <p>Access this report by clicking the threshold indicator in the Category Growth Analysis by Volume report.</p>
Item Growth Analysis by Volume	<p>This report displays sales growth based on overall volume by item. This report includes the percentage change in sales versus the same period last year, and the total year-to-date percentage change.</p> <p>Access this report by clicking the threshold indicator in the Category Growth Analysis by Volume report.</p>
Category Growth Analysis by Sales	<p>This report displays sales growth based on overall sales by product category.</p> <p>The report also enables a view by channel within time (Quarter and Month) and Item/Brand. Click the KPI threshold button to drill down to lower levels of detail: brand and item.</p> <ul style="list-style-type: none"> • Green indicates positive growth. • Yellow indicates zero growth. • Red indicates negative growth.
Brand Growth Analysis by Sales	<p>This report displays sales growth based on overall sales by brand.</p> <p>The report also enables a view by channel within time (Quarter and Month) and Item/Brand. Click the KPI threshold button to drill down to view growth analysis by item.</p> <ul style="list-style-type: none"> • Green indicates positive growth. • Yellow indicates zero growth. • Red indicates negative growth.

Item Growth Analysis by Sales	<p>This report displays sales growth based on overall sales by brand.</p> <p>The report also enables a view by channel within time (Quarter and Month) and Item/Brand.</p> <ul style="list-style-type: none"> • Green indicates positive growth. • Yellow indicates zero growth. • Red indicates negative growth.
Category Sales Volume by Type	<p>This report displays category sales and volume data by type. Type is defined as two different formats of sales value and volume:</p> <ul style="list-style-type: none"> • Promotion: Sales volume (unit sales) and Value (monetary amount) for those sales at retail point-of-sale that are priced and/or based on a promotion. • Regular: Sales volume and value for those sales at retail POS that are priced at standard non-Promotion periods or pricing.
Brand Sales Volume by Type	<p>This report displays brand sales and volume data by type. Type is defined as two different formats of sales value and volume:</p> <ul style="list-style-type: none"> • Promotion: Sales volume (unit sales) and Value (monetary amount) for those sales at retail POS that are priced, based, or both on a promotion. • Regular: Sales volume and value for those sales at retail POS that are priced at standard non-Promotion periods or pricing.
Item Volume by Type	<p>This report displays item sales and volume data by type:</p> <ul style="list-style-type: none"> • Promotion: Sales volume (unit sales) and Value (monetary amount) for those sales at retail POS that are priced, based, or both on a promotion. • Regular: Sales volume and value for those sales at retail POS that are priced at standard non-Promotion periods or pricing.

Category Sales Value by Type	<p>This report displays product category sales value by type:</p> <ul style="list-style-type: none"> • Promotion: Sales value (monetary amount) for those sales at retail point of sale that are priced, based, or both on a promotion. • Regular: Sales value for those sales at retail point of sale that are priced at standard non-Promotion periods or pricing.
Brand Sales Value by Type	<p>This report displays brand sales value, by type:</p> <ul style="list-style-type: none"> • Promotion: Sales value (monetary amount) for those sales at retail point of sale that are priced, based, or both on a promotion. • Regular: Sales value for those sales at retail POS that are priced at standard non-Promotion periods or pricing.
Item Sales Value by Type	<p>This report displays item sales value by type:</p> <ul style="list-style-type: none"> • Promotion: Sales value (monetary amount) for those sales at retail point of sale that are priced, based, or both on a promotion. • Regular: Sales value for those sales at retail POS that are priced at standard non-Promotion periods or pricing.
Retailer Profit by Category	<p>This report displays the specific profit estimated for the retailer, based on the data received, or through direct data received from the retailer.</p> <p>Gross Margin is calculated as the difference between the POS price multiplied by the quantity sold and the actual item cost.</p>
Retailer Profit by Brand	<p>This report displays the specific profit estimated for the retailer by brand. This report is based on the data received, or through direct POS data received from the retailer.</p> <p>Gross Margin is calculated as the difference between the point of sale price multiplied by the quantity sold and the actual item cost.</p>

Retailer Profit by Item	<p>This report displays the specific profit estimated for the retailer, by item. This report is based on the data received either externally or through direct POS data received from the retailer.</p> <p>Gross Margin is calculated as the difference between the point of sale price multiplied by the quantity sold and the actual item cost.</p>
Average Price Analysis	<p>This report displays average prices. You can filter the report by Market or Product Category.</p>
Top Performers by Profit	<p>This report displays the top performers based on retailer profit.</p>
Bottom Performers by Profit	<p>This report displays the bottom performers based on retailer profit.</p>
Top Performers by Sales	<p>This report displays the top performers based on overall sales volume at retail.</p>
Bottom Performers by Sales	<p>This report displays the bottom performers based on overall sales volume at retail.</p>
Syndicated Data	<p>This report displays the results of imported syndicated data.</p>

Exceptions

Exceptions Overview

Demand Signal Repository contains large amounts of detailed data from retailer Point of Sale systems. This data is voluminous, and finding specific problems can be time consuming. Exceptions give users a way to automatically sift through the detailed data to get to the most important issues that require further attention. These can either be positive cases, such as sales exceeding forecast by specified levels, or negative cases, such as inventory stockouts over extended periods of time.

Exceptions are generated using OBIEE iBots, which monitor the DSR repository for exceptions and uploads them periodically (for example, daily) to DSR.

Demand Signal Repository includes an Exception Management dashboard to provide exception-based alerting to monitor key store and regional distribution level supply chain KPIs related to promotions and new product introductions (NPIs). As well, exception measures can be used in Exceptions reports to report on potential issues where measures do not meet expectations.

Daily Versus Weekly Exceptions

With the exception of Forecast Accuracy (which is calculated weekly), all DSR exceptions are calculated daily. For daily exceptions, DSR creates one exception for each day that the exception occurs. The iBot only calculates an exception for prior periods, either day or week and ignores exceptions based on transactions with the current date. Similarly, the iBot calculates weekly exceptions for the prior week but does not calculate weekly exceptions for the current week. Note that weekly exceptions are dated as of the day at the end of the week for which the exception is calculated.

Exception Criteria

The following table explains each of the DSR exception types, along with how they are calculated:

Exception Type	Criteria for Exception Count	Exception Quantity	Exception Monetary Value
Out of Stock	<p>On Hand < or = 0 for the Business Unit and Authorized Business Unit = Yes</p> <p>Note: On Hand = null is not included in the exception count</p>	<p>Sales Forecast(Q) – Sales(Q)</p> <p>Note: If Sales Forecast(Q) is <null>, then result = <null>. If Sales(Q) is <null>, then Sales(Q)=0.</p>	<p>Sales Forecast(\$) – Sales(\$)</p> <p>Note: If Sales Forecast(\$ is <null> then result = <null>. If Sales(\$ is <null>, then Sales(\$)=0.</p>
Out of Stock (Imputed)	<p>Actual Sales < Minimum Threshold Sales</p> <p>Note: This calculation compares the actual sales with the minimum threshold sales. If the actual sales is far below the minimum threshold, then DSR infers that there is an out of stock. Expected sales is either Forecast Sales (Q) if not null, or else calculated based on prior sales.</p>	<p>Expected Sales(Q) - Sales(Q)</p>	<p>Expected Sales(Q) * (Average Selling Price) - Sales(\$)</p>

Sales Forecast Accuracy	<p>Weekly Sales Forecast Accuracy Threshold was loaded and Sales Forecast Accuracy % is less than Sales Forecast Accuracy Threshold</p> <p>or</p> <p>Weekly Sales Forecast Accuracy Threshold was not loaded, but a Weekly Sales Forecast Accuracy Goal was loaded and Sales Forecast Accuracy % is less than Sales Forecast Accuracy Goal.</p> <p>Note: NOTE: If no Sales Forecast Accuracy Goal or Threshold was loaded then no exception is generated. If Sales Forecast(Q) or Sales (Q) is null then no exception is generated.</p>	<p>ABSOLUTE ((Sales Forecast(Q) for the week - Sales (Q) for the week))</p> <p>ABSOLUTE ((Sales Forecast(\$) for the week - Sales (\$) for the week))</p>
-------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------

Reorder Point	<p>On Hand + BackOrder(Q) < Reorder Threshold</p> <p>Note: If both On Hand and BackOrder(Q) are null, then do not create the exception. If Reorder Threshold is <null> then do not create an exception.</p>	<p>Reorder Threshold - On Hand - BackOrder(Q)</p>	<p>(Reorder Threshold - On Hand - BackOrder(Q)) * Average Selling Price</p>
In-flight Promotional Lift	<p>Sales(Q) < 110 % * (Daily Average of the Sales(Q) for the last four full Manufacturing Business Calendar weeks)</p> <p>For example, if the average daily is being calculated for a Wednesday and the business calendar week runs from Sunday to Saturday, The average would include the four full business weeks concluding on the Saturday prior to the Wednesday's date.</p>	<p>- (Sales(Q)- 110 %* (Daily Average of the Sales(Q) for the last four full Manufacturing Business Calendar weeks))</p>	<p>- (Sales(Q)- 110 %* (Daily Average of the Sales(Q) for the last four full Manufacturing Business Calendar weeks)) * Average Selling Price</p>
New Item Not Selling	<p>If Today's Date - Available for Sale Date< 30 and [Authorized Location] = Yes and no sales before or during yesterday</p>	<p>Sales Forecast(Q)</p> <p>Note: If <null> then result = <null></p>	<p>Sales Forecast(\$)</p> <p>Note: If <null> then result = <null></p>

Promotional Price Deviation	ABSOLUTE {Average Selling Price – Promotional Price} > Promotional Price * 10 % and Promote = True	Sales(Q)	[Average Selling Price – Promotional Price] * Sales Quantity
Overstock	On Hand > Maximum Quantity On Hand	On Hand - Maximum Quantity On Hand	{Quantity On Hand - Maximum Quantity On Hand } * Average Selling Price

Exceptions Dashboard

Use the Exceptions dashboard displays summary information about any exceptions that exist in your organization. You can drill-down from the Exceptions dashboard to access the specific exception type report for that row.

- Exceptions Summary by Product Category
- Exceptions Summary by Customer
- Exception Type by Product Category
- Exception Type by Customer



Top Exceptions Dashboard

The Top Exceptions dashboard displays a summary of:

- Top exceptions for the current week
- Total out of stock exception count by week
- Top Exceptions by Product Category
- Top Exceptions by Customer

You can filter this dashboard by Date, Product Category, Customer, or Channel. Exceptions are displayed by count or monetary value.

Top Exceptions by Customer Dashboard

The Top Exceptions by Customer dashboard displays the largest exceptions by customer, calculated using either count (total exceptions) or dollar value. You can filter this dashboard by Date, Exception Type, Product Category, Customer, or Channel.

Top Exceptions by Product Category

The Top Exceptions by Product Category dashboard displays the largest exceptions by product category, calculated using either count (total exceptions) or dollar value. You can filter this dashboard by Date, Exception Type, Product Category, Customer, or

Channel.

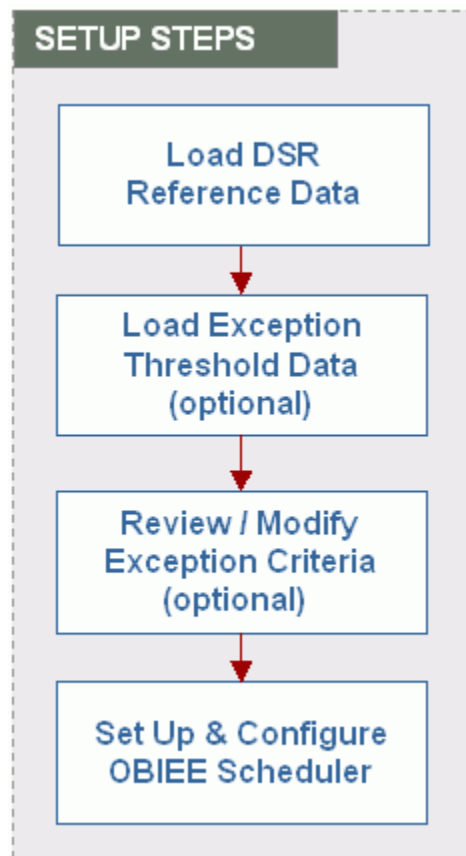
Top Exceptions by Customer

The Top Exceptions by Customer dashboard displays the largest exceptions by customer, calculated using either count (total exceptions) or dollar value. You can filter this dashboard by Date, Exception Type, Product Category, Customer, or Channel.

Exceptions Setup

OBIEE iBots are used to write exceptions. Each of the 8 exception types in DSR has a pre-built iBot. You run the iBots after loading new fact data to analyze the new data and generate any new exceptions. Each iBot executes a query which locates exceptions based on an established set of criteria. If you reload fact data because the original data had errors or was incomplete, you can also re-run the iBots to re-generate exceptions. When you run an Exception Management iBot, it will first remove any existing exceptions for the period being analyzed, and then write new exceptions for the period being analyzed using the latest fact data.

The following diagram shows the setup steps required to configure exceptions in DSR:



Step 1: Load DSR Reference Data

The first step to setting up exception management is to load the prerequisite reference data. This includes:

- Loading manufacturer reference data such as the manufacturer organization, business calendar, and the manufacturer's item hierarchy.
- Loading customer reference data such as the customer organization, item hierarchy and replenishment rules.

Step 2: Load Exception Thresholds

Exception thresholds define a minimum threshold where probability is high if actual sales is below threshold based on expected sales. For Expected Sales, by default DSR uses the sales forecast if it exists. If the forecast is not available, then DSR uses one of two calculations to estimate Expected Sales:

- Non-promotional items: a median of five previous same-day periods.
- For promotions: a median of promoted daily sales in past 180 days.

Exceptions are loaded to the table `DDR_R_EXCPTN_SLS_THRSHLD`.

Step 3: Review Exception Criteria

DSR uses Exception Analyzer Queries to classify queries that are used to locate exceptions. There is one exception analyzer query for each type of exception. When an exception management iBot runs an exception analyzer query, a record is written for each row in the query. The filter criteria within the exception analyzer query determines two things. First, it determines the number of days of related fact data that will be analyzed for exceptions. Exceptions are analyzed for a selected number of days ending with the current day. For example, the last seven or fourteen days. Second, the exception analyzer filter criteria is used to determine whether an exception exists or not. For example, in the case of the out-of-stock on-hand exception, the filter criteria selects records where the on-hand qty is equal to zero.

Because the data in each exception analyzer query is written to the same exception table, the first eight columns of the query must always contain the columns listed here.

- Mfg Org Code
- Rtl Org Code
- Business Unit ID
- Transaction Date

- Mfg SKU Item ID
- Rtl SKU Item ID
- Exception Qty
- Exception Amt

Note: Any columns in the query after the first eight are not written to the exception table and are available to help the end user review the exception.

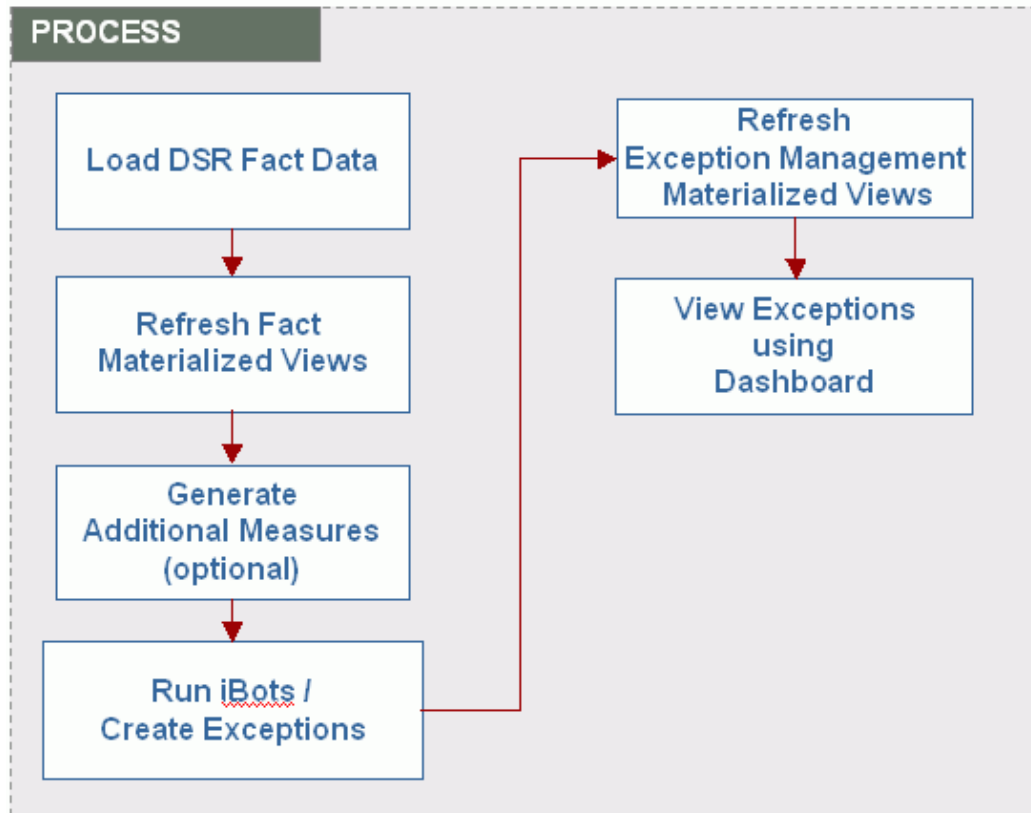
Step 4: Set up Scheduler

The last set up step is to set up and configure the OBIEE scheduler. The scheduler is used to run the iBots which will write the exceptions. You will need to create a file named `dbConnections.properties` containing the connection information specific to your environment and copy that file to the directory `$HOME/OracleBI/web/javahost/lib`.

To set up the scheduler, refer to Oracle® Business Intelligence Infrastructure Installation and Configuration Guide, chapter 10 for instructions.

Processing Exceptions

The following diagram illustrates the steps involved in processing exceptions in DSR:



Step 1: Load DSR Fact Data

Once set up is complete, you can start to process exceptions. The first step is to load the related fact data that will be analyzed to see if any exceptions exist. Each exception type uses data from one or more fact tables. For example, the Out of Stock (OH=0) exception analyzes the inventory facts to locate records where on-hand is zero. It also uses the sales forecast (if available) to calculate lost sales quantity and amount, which is then used as the exception quantity and amount. The facts that are currently used by exceptions include point of sale, inventory and sales forecast facts.

For more information on loading fact data in DSR, see Fact Data Load Process in the Working with Data chapter.

Step 2: Refresh Materialized Views

Once you have loaded the fact data, you will need to refresh the fact data materialized views. The first time you load fact data you will need to run a complete refresh. Once a complete refresh has been performed, you can run a fast refresh to update the materialized view with changes since the last refresh.

For more information on refreshing materialized views, see Materialized Views in the Working with Data chapter.

Step 3: Generate Additional Measures

Prior to creating exceptions, you'll want to generate some pre-calculated measures which are used by the exception analyzer queries. This includes:

- Expected Sales (by BU/SKU/Day).
- The associated minimum threshold related to the expected sales is fetched from the exception threshold table and stored in the additional measures table.
- The new items not selling (DDR_B_NEW_ITEM_NO_SLS) table is populated with the new items introduced to the store in the last 30 days. The new item start and end dates are captured along with the date of first sale.

The additional measures are generated by running the Additional Sales Measures process flow. This process flow can be found under the "DSR Exceptions Process Flow" folder in OWB. This process flow calls a PL/SQL procedure which then generates the additional measures. There is no interface table associated to the process flow. The only input required is to ensure the process flow parameters are set appropriately. The process flow parameters allow you to control what measures get created and the amount of fact data that is used when creating additional measures. For example, if you are not implementing the new item not selling exception, then you can bypass the creation of the new item not selling records and improve overall performance.

The following parameters are used in generating the additional measures:

- Calculate In Flight Promotion – If set to True, calculates Expected Sales Measure and copies the associated min threshold into the additional measures table.
- Calculate New item not selling – If set to True, populates the new items not selling (DDR_B_NEW_ITEM_NO_SLS) table .
- Calculate OOS Imputed Measures – If set to true, calculates Expected Sales Measure and copies the associated min threshold into the additional measures table.
- Customer Organization (optional) – If used, only generates additional measures for the selected customer organization.
- Offset Days (required) - The number of days of related fact for which additional measures will be created. Additional measures are created for a selected number of days ending with the current day. For example, the last seven or fourteen days.

Step 4: Run iBots

Once you have generated the additional measures, you can then run the iBot to create exceptions. There is one iBot for each exception type:

iBot Name	Exception Type
FCSTACC	Forecast Accuracy
IFPL	In-Flight Promotion Lift
NPISALES	New Product Introduction
OOSIM	Out Of Stock (Imputed)
OOSOH	Out Of Stock On Hand
OVRSTCK	Overstock
PPD	Promotional Price Deviation
REORDR	Reorder Point

Each iBot executes its associated exception analyzer query. For example, the Forecast Accuracy iBot executes the Forecast Accuracy exception analyzer query. You can see the name of the associated exception analyzer query listed in the Conditional Request tab. The mechanism by which the records retrieved from the exception analyzer query get written to the exception table can be found under the Advanced tab. When iBot conditions are satisfied (meaning at least one row was returned by the exception analyzer query), a Java program is called to read the data from the query and write an exception record for each row returned by the query. When iBot conditions are NOT satisfied (meaning no rows were returned by the exception analyzer query), a Java program is called to delete any existing exceptions for the period being analyzed.

Both programs called require the same parameter values. The first parameter is the exception type of the exceptions being written. This value should remain untouched. The second parameter is the exception ID. This value is used to purge exceptions. When processing exception, the program first removes any existing exceptions for the period being analyzed, before writing new exceptions. This allows you to rerun exceptions for the same period in case the fact data was incorrect or incomplete. For example, when processing Forecast Accuracy exceptions for the last 7 days, the program will first purge any forecast accuracy exceptions for the past 7 days that were written by this iBot. The way the program determines what was written by this iBot is that it deletes any records based with the exception ID equal to the second parameter. If you copy and create multiple exception analyzer queries and iBots for any one exception type, you will need to assign a unique exception ID for each iBot.

The last parameter is the offset days. This value must be the same as the offset days used in the associated exception analyzer query. For example, if the exception analyzer query is processing Forecast Accuracy exceptions for the past 7 days, you'll want to

make sure to pass the same number of offset days (7) to these programs. Remember these programs delete any existing exceptions before writing new exceptions, so we want to make sure they are only deleting exceptions for the period being processed.

Step 5: Refresh Materialized Views

Once exceptions have been created, you will need to refresh the exception data materialized views. The first time you load exception data you will need to run a complete refresh. Once a complete refresh has been performed, you can run a fast refresh to update the materialized view with changes since the last refresh. The exception materialized views can be refreshed by running the complete reference process flow or fast refresh process flow found under the DSR Exceptions Process Flows folder in OWB.

Step 6: View Exceptions

Once the exception materialized views have been refreshed, you can use the Exception Management dashboard to view and analyze exceptions and to drilldown into detailed reports.

Measures

Measures Overview

Use Demand Signal Repository's base measure values to calculate common performance measures, such as sales growth, gross margin, and inventory cover (days of supply). Demand Signal Repository calculates these measures at the appropriate hierarchy levels.

As well as base measures, Demand Signal Repository supports external measures, which are measures that cannot be calculated based on the base data model or that are provided directly by the retailer in precalculated form. You can load external measures into Demand Signal Repository.

Note: Demand Signal Repository does not map or aggregate external measures.

DSR Base Measures

Measure Name	Dimensions	Calculation or Loaded Level	Allocation Method	Aggregation Method	Description
Monetary Sales	Time, Customer Org, Customer Item	Time: Day Customer Org: BU (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Point of sale receipts

Monetary sales transaction	Time, Customer Org, Customer Item	Time: Day Customer Org: BU (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Monetary sales transactions.
Unit Sales	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Unit volume sold to consumers.
UOM Sales	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	UOM Sales.
Unit POS Forecast	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Forecasted demand at point of sale in unit terms.
Unit POS Forecast (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Forecasted demand at point of sale in unit terms (Alt).

Monetary POS Forecast	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Forecasted demand at point of sale in monetary terms.
Monetary POS Forecast transaction	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Monetary POS Forecast transaction.
On-hand Inventory	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Average	Ending balance of inventory at the storage location (units).
On-hand Inventory (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Average	Ending balance of inventory at the storage location (units) (Alt).
On-hand Inventory (retail basis)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Average	Ending balance of inventory at the storage location (on a retail value basis).

On-hand Inventory (cost basis)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Average	Ending balance of inventory at the storage location (on a cost basis).
Ordered Quantity	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Quantity of items ordered (by expected receipt date).
Ordered Quantity (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Quantity of items ordered (by expected receipt date) (Alt)
Ordered Value	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Value of items ordered (by expected receipt date).
Ordered Value (Txn)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Value of items ordered (by expected receipt date) (Txn)

Shipped Quantity	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Quantity of items shipped (units)
Shipped Quantity (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Quantity of items shipped (units).
Shipped Value	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Value of items shipped (on a cost basis).
Shipped Value (Txn)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Value of items shipped (on a cost basis) (Txn).
In-Transit Quantity	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Average	Quantity of items currently in-transit.

In-Transit Quantity (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Average	Quantity of items currently in-transit (Alt).
Received Quantity	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Quantity of items received (units).
Received Quantity (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Quantity of items received (units) (Alt).
Received Value	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Value of items received (on a cost basis).
Received Value (Txn)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	None	Sum	Value of items received (on a cost basis) (Txn).

Returned Quantity	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Quantity of items returned (units).
Returned Quantity (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Quantity of items returned (units).
Returned Value	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Quantity of items returned (units) (Alt).
Returned Value (Txn)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Sum	Value of items returned (on a cost basis).

DSR Derived Measures

Measure Name	Dimensions	Calculation or Loaded Level	Allocation Method	Aggregation Method	Description
--------------	------------	-----------------------------	-------------------	--------------------	-------------

Monetary Sales Growth	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Percentage growth of monetary sales compared to prior period.
Unit Sales Growth	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Percentage growth of unit sales compared to prior period.
Alt UOM Sales Growth	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Percentage growth of alternative UOM sales compared to prior period.
% Product to Category Coverage by Value	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Sku or Item	None	N/A	The sales of a single product as a percentage of total category sales - by \$ value.

% Product to Category Coverage by Volume	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Sku or Item	None	N/A	The sales of a single product as a percentage of total category sales - by unit volume.
% Category to Market Coverage by Value	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Levels above Sku & Item	None	N/A	The sales of a single category as a percentage of total market sales - by \$ sales. ACV provided by syndicated provider like Nielsen or IRI.
% Category to Market Coverage by Volume	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Levels above Sku & Item	None	N/A	The sales of a single category as a percentage of total market sales - by unit volume. ACV provided by syndicated provider like Nielsen or IRI.
Market Share (percentage of subcategory, category, department)	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	One supplier's products sales as a percentage of total category sales.

Retailer Gross Margin	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Difference between the retailer's cost and the total sales price of items.
Retailer Gross Margin Increase	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Difference between profit in current period and profit in prior period.
Retailer Gross Margin %	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Percentage difference between the retailer's cost and the average selling price of items.
Retailer Gross Margin Point Increase	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	The difference between the current gross margin percentage and the prior period gross margin percentage.

GMROII (or GMROI)	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Gross Margin Return on Inventory Investment: The retailer's return on inventory expenditures.
% Contribution Sales Value to Category	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	The sales of a single product as a percentage of total category sales - by \$ value.
% Contribution Profit to Category	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	The retail profit of a selected item as a percentage of total category retail profit.
Average Price	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Point of sale receipts divided by unit volume sold.

Unit Profit	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Monetary difference between the retailer's cost and the average selling price of items.
Total Category Average Sales Value	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Average per product sales amount for a specific product category - by \$ value.
Total Category Average Sales Volume	Time, Customer Org, Customer Item	Time: Any Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Average per product sales amount for a specific product category - by unit volume.
Sales Forecast Error	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Any	None	N/A	The percentage error of the POS forecast vs actual sales for the period.
Sales Forecast Accuracy	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Any	None	N/A	The percentage accuracy of the POS forecast vs actual sales for the period.

Inventory turns (annual)	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer Customer Item: Any except Item	None	N/A	The value of annual shipments at cost (for the most recent period) divided by the total average daily inventory value at cost.
Max Inventory cover	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Store Customer Item: Any except Item	None	Ending Balance for Week, Average for time periods above Week	Number of days of historical or forecast inventory consumption represented by quantity on-hand.
Min Inventory cover	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Store Customer Item: Any except Item	None	N/A	Number of days of historical or forecast inventory consumption represented by quantity on-hand.
Promotional Sell-Thru	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, RDC, Store Customer Item: Above Category	None	N/A	Percentage of promotional orders that sold through during the promotional period.

% of Stores Scanning	Time, Customer Org, Customer Item	Time: Day, Week Customer Org: Market Area, Retailer Group, Retailer Customer Item: SKU	None	Average	Percentage of store locations that sold a particular item.
% Not Scanning	Time, Customer Org, Customer Item	Time: Day, Week Customer Org: Market Area, Retailer Group, Retailer Customer Item: SKU	None	Average	Percentage of store locations that did not sell a particular item.
In-stock %	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Market Area, Retailer Group, Retailer Customer Item: Supplier, Group, Category, SubCat, Brand, SKU	None	N/A	Percentage of stocking locations that had stock available when sampled.

Understock %	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Market Area, Retailer Group, Retailer Customer Item: Supplier, Group, Category, SubCat, Brand, SKU	None	N/A	Percentage of stocking locations that were below their minimum expected stock level when sampled.
Lost Sales	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Market Area, Retailer Group, Retailer, Store Customer Item: Supplier, Category, SubCat, Brand, SKU	None	Sum	Estimated value of sales lost due to out-of-stock conditions.
% Returns	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Supplier, Category, SubCat, Brand, SKU	None	N/A	The percentage of product that the retailer deems unsalable, as a percentage of cost or units sold.

% Change Average Price	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	The percentage change for the average price.
% Change Profit	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	The percentage change in profit.
Monetary Sales (Category)	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Monetary Sales (Category)
Total Category Average Profit	Time, Customer Org, Customer Item	Time: Above Day Geography: Any Customer Org: Any Customer Item: Any	None	N/A	Total Category Average Profit

% Change Same Period Last Year By Volume (Unit Sales)	Time, Customer Org, Customer Item	Time: Year Geography: Any Customer Org: Any Customer Item: Any	None	N/A	% Change Same Period Last Year By Volume (Unit Sales)
% Change Same Period Last Year By Sales Value (Monetary Sales)	Time, Customer Org, Customer Item	Time: Year Geography: Any Customer Org: Any Customer Item: Any	None	N/A	% Change Same Period Last Year By Sales Value (Monetary Sales)
% Product to Category Last Year By Volume	Time, Customer Org, Customer Item	Time: Year Geography: Any Customer Org: Any Customer Item: Below Category	None	N/A	% Product to Category Last Year By Volume
% Product to Category Last Year By Sales Value	Time, Customer Org, Customer Item	Time: Year Geography: Any Customer Org: Any Customer Item: Below Category	None	N/A	% Product to Category Last Year By Sales Value

% Change Current Year To Date By Unit Volume	Time, Customer Org, Customer Item	Time: Year Geography: Any Customer Org: Any Customer Item: Below Category	None	N/A	% Change Current Year To Date By Unit Volume
% Change Current Year To Date By Monetary Value	Time, Customer Org, Customer Item	Time: Year Geography: Any Customer Org: Any Customer Item: Any	None	N/A	% Change Current Year To Date By Monetary Value
Back-Ordered Quantity	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Average	Back-Ordered Quantity
Back-Ordered Quantity (Alt)	Time, Customer Org, Customer Item	Time: Day Customer Org: BU, (Store or DC) Customer Item: SKU	Based on Allocation table	Average	Back-Ordered Quantity (Alternate)

DSR External Measures

The following external measures can be loaded into DSR:

Measure Name	Dimensions	Calculation or Loaded Level	Allocation Method	Aggregation Method	Description
Invoice Accuracy %	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	Percent deviation of invoice amount from agreed price and received quantity, or percentage of invoices that fell within tolerance established.
Payment Days	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	Average number of days from invoice to payment.
Late Payment	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	Average number of days payments were late.

Number of Deductions	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	Count of deductions taken.
Deduction Balance	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	Monetary amount of deductions taken.
Deduction Balance	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	Monetary balance of outstanding deductions.
Item Data Accuracy %	Time, Customer Org, Customer Item	Time: Week or Month Customer Org: Retailer Group or Retailer Customer Item: Supplier	None	N/A	The percentage of items that are synchronized via the Global Data Synchronization Network (GDSN).

Item Data Synchronization %	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Any	None	N/A	The percentage error of the POS forecast vs actual sales for the period.
Order Change %	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer Customer Item: Above Category	None	N/A	Percentage of orders or line items that had to be revised prior to shipment.
Perfect Order %	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, RDC Customer Item: Above Category	None	N/A	The number of orders with no errors, filled correctly for every line, on time and invoiced correctly divided by the total number of orders.
Order Cycle Time	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Above Category	None	N/A	The average number of days from when a PO is written until it is received.

On-time Delivery (appointment date)	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Above Category	None	N/A	Percentage of shipments delivered by appointment date.
On-time Delivery (appointment time)	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Above Category	None	N/A	Percentage of shipments that were delivered within the appointment window on the appointment date.
Service Level / Fill Rate	Time, Customer Org, Customer Item	Time: Above Day Customer Org: Retailer Group, Retailer, Store Customer Item: Above Category	None	N/A	Amount ordered vs amount shipped.

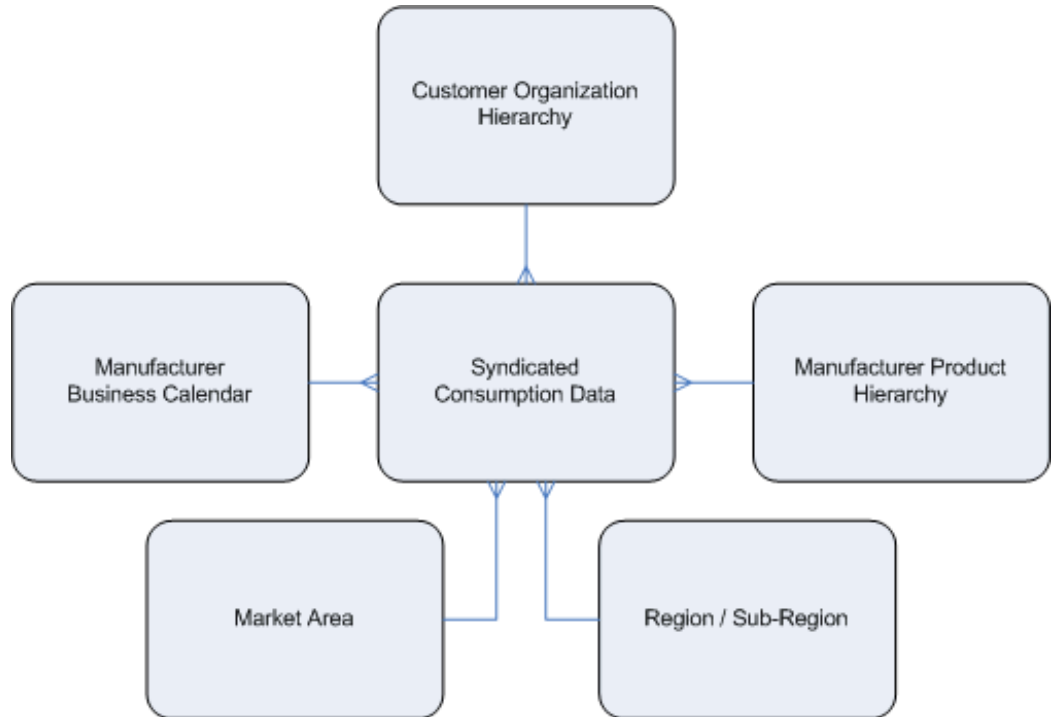
Loading Syndicated Consumption Measures

Syndicated consumption data can be purchased at various levels of the product hierarchy and customer organization hierarchy, as well as geographic region or sub-region and then loaded into Demand Signal Repository. In order to calculate ACV %, the consumer goods manufacturer must obtain store volumes of all stores in the desired geographical area. Obtaining store volume information from syndicated data sources is expensive and can be cost prohibitive. The more common practice is to purchase pre-calculated ACV measures from the syndicated data source for the desired Product, Retailer and/or Geography, and Time Period combinations.

Since Consumer Goods Manufacturers may receive ACV consumption data from

multiple sources, DSR allows the same measure to come from different sources, so long as the data is for different dimensions (for example, geographies). The system will issue an error when a measure is received from a second source for the same dimension. For example, if Kroger measures were received for the Atlanta market area from IRI, an error would be issued if Kroger data for the Atlanta market area was received from AC Nielsen (or other source).

DSR Syndicated Consumption Data references the following logical entities:



All Commodity Volume (ACV) Measures

One of the types of syndicated consumption measures are All Commodity Volume (ACV) measures. These measures are commonly used by consumer goods manufacturers to analyze market share. ACV represents the total annual sales volume of retailers, which is aggregated from the individual store level up to larger geographical sets. When dividing total sales for a store into ACV million dollar blocks, it allows for weighting across different sized markets. For example, Vision Consumer Goods (a beverage manufacturer) has its Homestyle Root Beer distributed in one supermarket in a community with five supermarkets total. A straight percentage of stores displaying Homestyle Rootbeer would read 20% display; however, the one store that handles Homestyle Root Beer is responsible for 90% of all supermarket sales in the community. The numbers that would be reflected for Homestyle root beer would be 90% ACV display activity. In this example, ACV % is the percentage of exposure to consumer spending in the class of trade being analyzed for Homestyle Rootbeer during the specified time period.

Sample Syndicated Consumption Data

The layout of syndicated consumption data varies by Consumer Good manufacturer. The order of columns, measures to be included, and date format may be different; however, the measures will always have an associated product key, geography key and time code. Below is a sample of syndicated consumption data. The product key and geography description fields (bolded) are not actually part of the file; they have been included in the file layout to make it easier for the reader to understand the data. The number of measures received will vary based on what data the Consumer Goods manufacturer has purchased.

Pro duc t Key	Product Key Description	Geog raph y Key	Geography Key Description	Time Code (Week Endin g Date)	Meas ure 1 (Base Dollar s, No Prom o)	Measu re 2 (Base Volum e, No Promo)	Measu re 3 (% ACV, No Prom o)	Meas ure 4 ACV \$ (MM)
100 1	Fit Water 1 Lt	52431 3	KROGER ATLANTA-RMA - FOOD	81130	4508.5 7	49416	6.13	3,500
100 1	Fit Water 1 Lt	52431 7	KROGER DALLAS-RMA - FOOD	81130	6102.1 8	59160	7.95	4,500
100 1	Fit Water 1 Lt	20099 6	SOUTHEAST - FOOD - STANDARD REGION	81130	7713.0 6	62033	9.77	3,600, 000
100 1	Fit Water 1 Lt	20099 7	NORTHEAST - FOOD - STANDARD REGION	81130	1309.7 1	8160	11.59	4,600, 000
200 1	Filtered Water	52431 3	KROGER ATLANTA-RMA - FOOD	81130	1506.7 6	8220	15.23	3,500
200 1	Filtered Water	52431 7	KROGER DALLAS-RMA - FOOD	81130	2109.0 7	9001	18.87	4,500
200 2	Flavored Water	52431 3	KROGER ATLANTA-RMA - FOOD	81130	1004.0 4	4213	22.51	4,500

200 2	Flavored Water	52431 7	KROGER DALLAS-RMA - FOOD	81130	5504.1 2	23391	26.15	3,500
300 1	Bottled Water	52431 3	KROGER ATLANTA-RMA - FOOD	81130	18518. 57	63426	31.60	3,500
300 1	Soft Drinks	52431 7	KROGER DALLAS-RMA - FOOD	81130	21519. 57	66427	37.06	4,500

Product Key

The product key is a numeric code that represents one of the levels in a product hierarchy. For example, the data above illustrates data received at the SKU level (the first four rows), product subclass level (rows 5 – 8), and class (rows 9 & 10). To translate product key into DSR terms, you would need a table that looks like the following:

Source	Product Key	Product Name	Product Hierarchy Level	Product Code
IRI	1001	Fit Water 1 Lt	SKU	Fit Water 1 Lt
IRI	1001	Fit Water 1 Lt	SKU	Fit Water 1 Lt
IRI	1001	Fit Water 1 Lt	SKU	Fit Water 1 Lt
IRI	1001	Fit Water 1 Lt	SKU	Fit Water 1 Lt
IRI	2001	Filtered Water	Subclass	Filtered Water
IRI	2001	Filtered Water	Subclass	Filtered Water
IRI	2002	Flavored Water	Subclass	Flavored Water
IRI	2002	Flavored Water	Subclass	Flavored Water
IRI	3001	Bottled Water	Class	Bottled Water
IRI	3001	Soft Drinks	Class	Soft Drinks

Note: Since product and geography keys are unique by data source, it's necessary to identify the source in order to translate the product key.

Geography Key

The geography key is a numeric code that represents a combination market area, retail organization (at various levels of the hierarchy), Region or Sub-region and Channel. To translate geography key into DSR terms, you would need a table that looks like the following:

Source	Geography Key	Geography Name	Market Area	Organization Level	Organization Code	Region	Sub-Region	Channel
IRI	524313	KROGER ATLANTA-RMA - FOOD	ATLANTA	Chain	KROGER	na	na	FOOD
IRI	524317	KROGER DALLAS-RMA - FOOD	DALLAS	Org	KROGER	na	na	FOOD
IRI	524321	KROGER MID-ATLANTIC-RMA - FOOD	ATLANTA	Org	KROGER	na	na	FOOD
IRI	524322	KROGER MID-SOUTH-RMA - FOOD	SOUTH	Org	KROGER	na	na	FOOD
IRI	524302	KROGER QFC SEATTLE-RMA - FOOD	SEATTLE	Chain	KROGER QFC	na	na	FOOD

The data listed above illustrates valid combinations represented by geography key. For example, market area, region and sub-region are mutually exclusive. A measure can only be associated to one. Valid combinations include:

- Channel / Region
- Channel / Sub-region
- Channel / Market Area

- Channel / Market Area / Retail Organization
- Channel / Retail Organization

Setting Goals

Demand Signal Repository provides the ability to load goals and thresholds at various levels of the organization and item hierarchy for a given time period (i.e., year, quarter, week, etc.) The main purpose of goals and thresholds is to determine whether performance targets are being met; however, these goals and thresholds are not currently used in driving the color-coded performance indicators.

You can enter a goal for each measure at any level combination that is available in the scorecard dashboard. By default, a goal applies to all periods from the present going forward. Alternatively, you can specify a beginning and ending period for which the goal being entered should apply. In this way, goals can be set retroactively, and can be set for future periods without changing the goals for current periods.

The goal is either a constant scalar value, such as 9% for sales growth, or a ratio applied to a prior result (for example, units sold last year's sales * 1.09 = goal for sales units this year for each week of the year).

You can also enter a single goal value that is to apply to many categories and/or regions. For example, if a user wants to set the days of supply goal for each of its regions to 14 days, they can select all of the combinations to which this goal is to apply, and enter the value once. The same value will be saved for each of the affected combinations. If new regions are subsequently added at a level within this type of goal specification, they automatically inherit these goals.

Runtime Evaluation of Goals

DSR uses color-coded performance indicators which are based on the loaded goals and thresholds. For each measure, a goal is either the lower or upper bound of performance. For example, a goal for sales growth is a lower bound measure, since exceeding the performance is considered good news. In other cases, such as the goal for days of supply (an upper bound measure) a higher number of days of supply would indicate an overstock (or negative) condition. If a goal is set for a measure and data is available for the period being measured, the display for that product/location/measure combination will be green if:

- The measure is a lower bound and the value meets or exceeds the goal
- The measure is an upper bound and the value is at or below the goal

The display is red if:

- The measure is a lower bound and the value is outside the tolerance for a major issue

- The measure is an upper bound and the value is outside the tolerance for a major issue

Otherwise the display is yellow.

If there is no data for the period being displayed for a measure, the result for the most recently available prior period should be displayed, with some visual indicator (hash marks or otherwise) that the value is not current. If there is no data from any prior period that was measured against a goal, the display for that product/location/measure combination is grayed.

Setting Thresholds

You can set Demand Signal Repository thresholds that determine whether goals are in the upper or lower bounds of performance.

For example, a goal for sales growth would typically be a lower bound, since exceeding the performance would be considered good news. Meanwhile, the goal for days of supply might be an upper bound, since a higher number of days of supply would indicate an overstock condition.

Note, however, that days of supply can also be too low, creating an understock condition. In this case, a separate goal could be set for the measure as a lower bound. It would then have a separate place in the scorecard for understock and overstock monitoring.

The second aspect of threshold setting is determining whether a goal shortfall is minor (yellow status) or major (red status). Users can either set a global tolerance for this (greater than 20 percent shortfall or overshooting of the goal indicates a red status), or they can specify a tolerance (in percentage or absolute terms) for individual measures, much as they set up the goals themselves.

Note: Scorecard thresholds are only applicable to retailer hierarchies.

Working with Data

Data Overview

Demand Signal Repository is able to receive data not only for a manufacturer's own items, but for competitor items. This allows manufacturers to perform comparative sales analysis of its own items versus competitor items. As well, sales personnel are able to view inventory and delivery information.

Product descriptions and retailer item hierarchy associations for competitor items are typically received from the retailer. A retailer may send item hierarchy data for their own items and competitor items together in the same file. The retailer may be able to provide a vendor number to identify the particular supplier of competitor items; if not, the manufacturer must rely upon another source to identify the supplier – either a 3rd party or their own internal knowledge of the GTIN values.

Competitor item data has the same general format and organization as the manufacturer's own item data, but is treated differently for aggregation and reporting purposes. Mapping of competitor items to the manufacturer's item hierarchy is a service provided by 3rd parties such as Nielsen or IRI – otherwise, manufacturer personnel must classify the competitor items themselves using their own knowledge. Competitor items are added to the manufacturer's item hierarchy by the same mechanism used to load and maintain hierarchies for a manufacturer's own items.

At aggregate level within either the retailer or manufacturer hierarchies, Demand Signal Repository maintains the manufacturer's aggregate value and the aggregate value for each of the competitor organizations. You can also view the total value for all suppliers.

As the manufacturer does not generally have the competitor's internal item identification, the GTIN will be used for both the global and local ID by default.

Sharing Data with Manufacturer Applications

Demand Signal Repository shares demand data with manufacturer applications at both the lowest level of detail, and at an aggregate level. A web service for each group of facts in the data model will allow an application to select and retrieve data according to

the dimensions of time, manufacturer item hierarchy, retailer org hierarchy (selected levels) and location.

Hierarchy Loading

All hierarchy and calendar data (except for the Gregorian calendar) must be loaded into Demand Signal Repository from external sources:

- **Organization Hierarchies:** Organization hierarchies consist of organizations, business units, and a fixed taxonomy (district, area, region, etc.) that relates them. The inbound interface tables follow the same structure – one each dedicated to the organizations with their attributes, the business units with their attributes, and a table containing the parent/child relationships in the hierarchy.
- **Item Hierarchies:** Item hierarchies are organization-specific. SKU-level attributes are loaded into one interface table; parent-child relationships go into another. Demand Signal Repository adds, deletes and moves in the same manner as the organization hierarchy.

Item and organization hierarchies may be updated by loading a new complete hierarchy that reflects the changes since the last hierarchy was loaded. Updates are handled as follows:

- **New nodes:** New nodes must be added to the existing hierarchy in the places indicated by the new hierarchy.
- **Removed nodes:** If a node in the existing hierarchy no longer is included in the new hierarchy, the affected node in the existing hierarchy, Demand Signal Repository takes no action in order to maintain the consistency of historical data. A user may explicitly set the ending effectivity date to indicate that the node is no longer valid.
- **Moved nodes:** If a node moves from one position in the hierarchy to another, a successor node is created in the new location, and the existing node's ending effectivity date is set. Any descendants of the moved node move along with its parent, without creating successor nodes.

Other changes to nodes in the updated hierarchy (such as a revised description) simply overwrite the corresponding attributes in the node in the existing hierarchy, without creating a successor node.

Caution: Changes to a hierarchy are made as of the source system date in the incoming file. If the source system date is in the past, and fact data has been loaded in the interim, aggregate quantities may not match those generated in the retailer's source system for the same data. It is a best practice to load any new hierarchy data prior to any fact data load.

Note: Both SKU- and item-level attributes are loaded via SKU-level interface table rows. If different SKUs that belong to the same item specify different values for item-level parameters, the values specified by the last SKU loaded will overwrite the previous values, so that all SKUs for that item will inherit the item-level attributes from the last SKU.

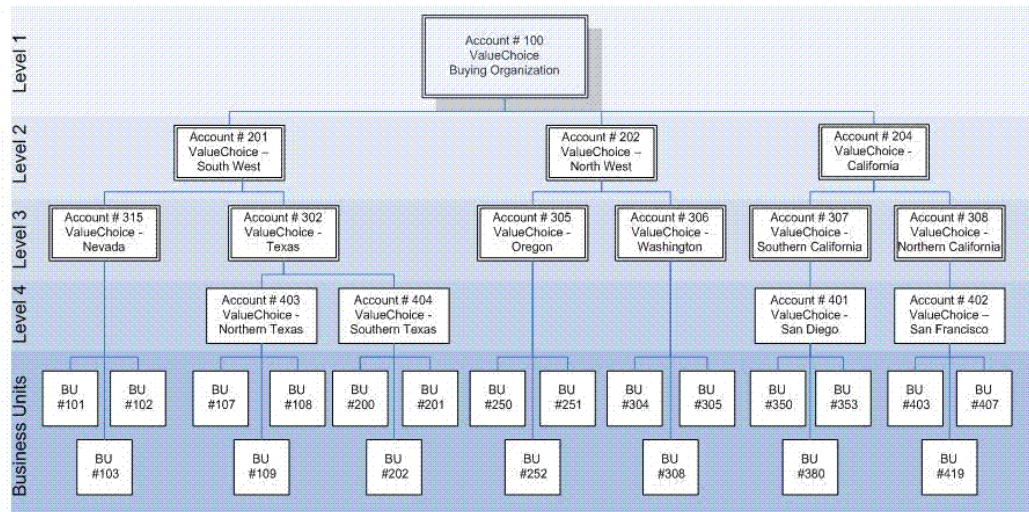
Alternate Organization Hierarchies

Demand Signal Repository captures, analyzes, and displays facts based on a customer's view of their organizational structure. However, some manufacturers (and their supply chain partners) may also have their own view of how the customer's organization is structured for purposes of account management, promotion planning, collecting and aggregating consumption data, and so on. This view is oftentimes different from how the customer views its organization structure.

You can view a customer's organization using an alternate organization hierarchy while keeping the original customer organization hierarchy intact. Facts can be inquired upon using the original customer organization hierarchy, or one or more alternate organization hierarchies. This allows users to view customer data at aggregate levels that represent the various ways in which the manufacturer conducts business with the customer. Users can also analyze fact data at a level that is equivalent to other systems in use by the Manufacturer. For example, to analyze and report on DSR retail demand data by planning account (that is, a level in an alternate organization hierarchy), which would be equivalent to the planning account used in Oracle Demantra Predictive Trade Planning or Oracle Siebel Consumer Goods.

The Alternate Organization Hierarchy feature allows other views of the customer's organization to co-exist within DSR along with the original customer organization hierarchy.

Here's an example of an alternate organization hierarchy. The customer, ValueChoice, has five levels within its buying organization, including business units (which are distribution center or store locations). The Business Unit must be the last level in the alternate organization hierarchy in order for the associated facts, such as sales or inventory facts, to be aggregated to higher levels in the alternate organization hierarchy.



A hierarchy level can contain accounts that have different levels of granularity. For example, at Level 2 in the hierarchy, we see accounts which represent a large geographical area (such as South West and North West) as well as an account that encompasses just one state (California). This occurs because some geographical areas, such as California, may have more stores, than several other states combined. Consequently, California requires the same level of account management as larger territories.

In addition, the hierarchy may be "ragged", meaning it does not contain the same number of levels across all branches of the hierarchy. For example, account # 315, ValueChoice – Nevada, does not have a level 4 account. Neither does Washington or Oregon. In this case, the business unit is associated directly to the level 3 account.

In this example, the account numbers that have been assigned directly correlate to their respective levels, in the real world the account numbers would vary across hierarchy levels, meaning Account # 3456 may be a level 1 account, while Account # 10, may be a level 3 account.

This example illustrates a hierarchy that has 4 levels plus business unit. DSR supports having an Alternate Organization Hierarchy of up to 10 levels (plus business unit).

The Alternate Organization Hierarchy provides a flexible organization structure, which can vary depending on the source of the data, or the type of analysis performed.

Example uses for the Alternate Organization Hierarchy include:

- Representing the organization hierarchy as defined by the Syndicated Data Provider: Manufacturers who subscribe to a syndicated data provider service for account information can load this information into the alternate organization hierarchy and leave the original customer organization hierarchy untouched, so that account team's can still talk to customers in terms of the customer's view of their organization.
- Representing the Customer Buying Organization or Account Planning Hierarchy:

The manufacturer's own view of the customer's organization may be based on how the customer's buying organization is. When performing activities, such as promotion planning and execution, a promotion may apply to a level within the retailer's organization, which represents the area for which a buyer is responsible. For example, a retail department or buying region. One or more of these views of the customer organization can be loaded to the alternate organization hierarchy so sales can be analyzed by planning account.

Interface Tables

The following OWB interface tables are used with Alternate Organization Hierarchies:

- **DDR_I_AOH_ATTR**: Interface table for loading Alternate Organization Hierarchy code and descriptions. The target table is **DDR_R_ORG_HCHY**.
- **DDR_I_AOH_ACCT_ATTR**: Interface table for loading AOH account attributes and their associated parent relationship. Target tables are **DDR_R_ORG_BSNS_ENT** and **DDR_R_AOH_LVL_DTL**.
- **DDR_I_BU_AOH_ASSOC**: Interface table for loading AOH / BU associations. Target table is **DDR_R_BU_AOH_ASSOC**.

Error Tables

For each of the interface tables listed on the previous slide, there is a corresponding error table. The error table contains all the columns found in the interface table, plus an error message column which contains the associated error message (or messages) and a load ID which assigns a unique number each time data is loaded and errors are encountered.

- **DDR_E_AOH_ATTR**: Holds error records for **DDR_I_AOH_ATTR** interface table.
- **DDR_E_AOH_ACCT_ATTR**: Holds error records for **DDR_I_AOH_ACCT_ATTR** interface table.
- **DDR_E_BU_AOH_ASSOC**: Holds error records for **DDR_I_BU_AOH_ASSOC** interface table.

DDR_R_ORG_HCHY Target Table

The Organization Hierarchy (**DDR_R_ORG_HCHY**) table stores the Alternate Organization Hierarchy Code and associated description. This table existed in previous releases of DSR, but was not in use until now. Listed here are some of the key columns: Organization Hierarchy ID is the unique index to the table. It is a number assigned by the system. Each Hierarchy Code (**HRCHY_CD**) is assigned a unique Organization Hierarchy ID.

The effective From and Through date are currently informational only. When loading the table, the source system date is used as the effective from date. Effective through date is left blank.

The Source System Date is the date in which the data was created on the source system. The Source System Identifier is used to identify where the data came from.

DDR_R_ORG_BSNS_ENT Target Table

The Organization Business Entity (DDR_R_ORG_BSNS_ENT) table stores AOH Account Attributes and the associated parent relationship. Listed here are some of the key columns:

- Organization Business Entity ID is the unique index to the table. It is a number assigned by the system. Each Business Entity Code (BSNS_ENT_CD) is assigned a unique Organization Business Entity ID.
- The effective from and through date are informational only. When loading the table, the source system date is used as the effective from date. Effective through date is left blank.
- Attributes are captured for each business entity such as name, description, and address. In addition, there are 20 user-defined fields (10 character and 10 numeric) which can be used to capture additional information. These user-defined attributes can be made visible in Answers by modifying DSR's OBIEE repository.

DDR_R_AOH_LVL_DTL Target Table

The AOH Level Detail (DDR_R_AOH_LVL_DTL) table stores a flattened version of the AOH parent/child relationships. This table is populated at the same as when the Business Entity (from the previous slide) are loaded. Listed here are some of the key columns:

- Business Entity ID (BSNS_ENT_ID) is the unique index to the table. There is a direct association between this table and the Organization Business Entity (DDR_R_ORG_BSNS_ENT) table. A record exists in this table for each child record in Organization Business Entity (DDR_R_ORG_BSNS_ENT) table.
- Parent Business Entity Codes (and associated Ids) 1 through 9 contain a flattened version of the child's lineage up to the topmost level. DSR supports an Alternate Organization Hierarchy of 10 levels (not including business unit) without having to modify the existing AOH tables.

DDR_R_BU_AOH_ASSOC

The BU AOH Association (DDR_R_BU_AOH_ASSOC) table stores the relationship between AOH Account and Business Unit.

The combination of Org Hierarchy ID (ORG_HCHY_ID) and Business Unit ID (BSNS_UNIT_ID) make up the unique index to the table. A business unit can only be associated to one AOH account within each Alternate Organization Hierarchy (also known as Organization Hierarchy ID).

Alternate Organization Hierarchies Materialized Views

Materialized views were created to provide quicker access when querying over the alternate organization hierarchy table in OBIEE. The AOH materialized views are used to aggregate facts with accounts at distinct levels. So each materialized view contains only the accounts at that level and their complete lineage. These materialized views are refreshed as part of the REFERENCE_MV_COMPLETE_REFRESH process flow:

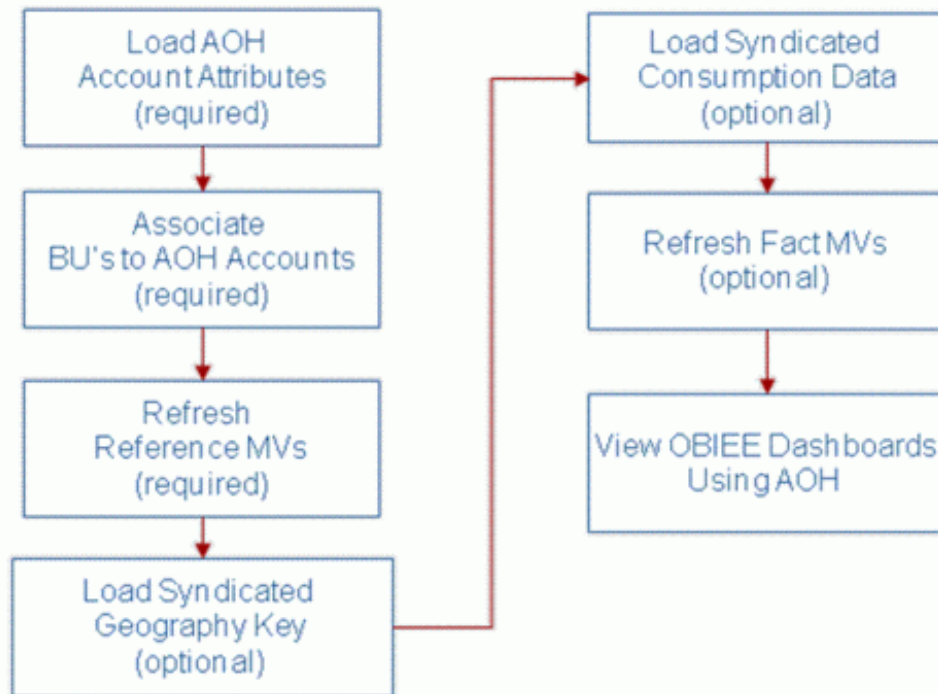
- DDR_R_AOH_LVL_1_ATTR_MV: Contains a "flattened version" of the AOH (level 1 accounts only).
- DDR_R_AOH_LVL_2_ATTR_MV: Contains a "flattened version" of the AOH (level 2 accounts and their lineage).
- DDR_R_AOH_LVL_3_ATTR_MV: Contains a "flattened version" of the AOH (level 3 accounts and their lineage).
- DDR_R_AOH_LVL_4_ATTR_MV: Contains a "flattened version" of the AOH (level 4 accounts and their lineage).

Alternate Organization Hierarchies Process Flows

The following OWB process flows have been created to support Alternate Organization Hierarchies:

- AOH_ATTR: Loads the AOH Codes which uniquely identifies each Alternate Organization Hierarchy in use.
- AOH_ACCT_LVL: Loads account information, including how the accounts are related to each other through the use of parent account.
- BU_ACCT_ASSOC: Loads the relationship between Business Units and Alternate Organization Hierarchy Accounts. This allows facts associated with a Business Unit to be viewed using the Alternate Organization Hierarchy.
- SYND_GEO: Loads data which is used to translate the syndicated provider's geography key to dimension values (market area, region, AOH Account, and so on) that are meaningful to DSR.
- SYND_CNSMPTN_INF_STG_TO_TGT: Loads external aggregated consumption (for example, sales) measures received from syndicated data providers.

PROCESS



To view DSR facts using an Alternate Organization Hierarchy, you need to complete the following steps:

1. Load the Alternate Organization Hierarchy Account Attributes. This defines the valid accounts (or nodes) within the hierarchy and how they related to each other.
2. Load information which associates the customer's business units (meaning stores or distribution centers) to an account within the Alternate Organization Hierarchy.
3. Refresh the reference data materialized views which summarizes data to improve performance when running queries.

After completing these steps, you can use the Alternate Organization Hierarchy and display facts associated with the Business Units. This includes Sales, Sales Forecast, Inventory, Order and Shipment facts. If you also need to associate Syndicated Consumption Data to an account within the Alternate Organization Hierarchy, you will also need to perform the following tasks:

1. Load the Syndicated Geography Key information. This information provides a cross-reference from the Geography Key provided by the syndicated data provider to the appropriate Alternate Organization Hierarchy Account (or other dimension within DSR).

- 2. Load the Syndicated Consumption Data.
- 3. Refresh the Fact Materialized Views which summarizes data to improve performance when running queries.

You can now view dashboards or create queries in OBIEE that use the Alternate Organization Hierarchy.

Step 1 Load Account Attributes

Maps and Process Flows have been created to load Alternate Organization Hierarchy information. The first step is to load Alternate Organization Hierarchy Account information. The Alternate Organization Hierarchy Account Attributes Process Flow loads account information, such as account name, address, and contact information, including how the accounts are related to each other through the use of parent account.

When you execute the AOH Account Level process flow, the system reads data from the source table, performs the necessary validations and populates the target tables listed below. Prior to loading Alternate Organization Account information, you must load:

- Organization Hierarchy (DDR_R_ORG_HCHY) with at least one Alternate Organization Hierarchy code
- Country codes (DDR_R_CNTRY)
- State codes (DDR_R_STATE).

Process Flow Name:	AOH_ACCT_LVL
Source Table(s):	DDR_I_AOH_ACCT_ATTR
Target Table(s):	DDR_R_ORG_BSNS_ENT DDR_R_AOH_LVL_DTL
Dependent Table(s):	DDR_R_ORG_HCHY DDR_R_STATE DDR_R_CNTRY

Required Fields:	AOH Code
	Account Code
	Account Name
	Parent Account
	Ultimate Parent Account
Key Validations:	AOH Code be a valid value in the Organization Hierarchy (DDR_R_ORG_HCHY) table
	Country Code and City Codes (if populated) must be valid. Address information is optional, so these fields are not required.
	Parent Account must be valid account within the same Ultimate Parent Account. So, if the account record that's being validated belongs to Ultimate Parent Account 100, the parent of that account must also have an Ultimate Parent Account of 100.
	Ultimate Parent Account must be a level 1 account, meaning it cannot be a child of another account.

Step 2: Associate Business Units to Alternate Organization Hierarchy Accounts

The second step is to connect the existing Business Units to the Alternate Organization Hierarchy. The Business Unit Account Association Attributes Process Flow loads data which associates Business Units to an Alternate Organization Hierarchy Account. This allows facts associated with a Business Unit to be viewed using the Alternate Organization Hierarchy.

When you execute the BU Account Association process flow, the system reads data from the source table, performs the necessary validations and populates the target table listed here.

Prior to loading the Business Unit / Account Association information, you will need to have loaded the Alternate Organization Hierarchy Account tables (DDR_R_ORG_BSNS_ENT and DDR_R_AOH_LVL_DTL) as well as the Customer's Organization Business Unit table (DDR_R_ORG_BSNS_UNIT).

Process Flow Name:	BU_ACCT_ASSOC
Source Table(s):	DDR_I_BU_AOH_ASSOC

Target Table(s):	DDR_R_BU_AOH_ASSOC
Dependent Table(s):	DDR_R_ORG_BSNS_ENT DDR_R_ORG_BSNS_UNIT DDR_R_AOH_LVL_DTL
Required Fields:	Org Code Business Unit Code AOH Code Business Unit Parent Account Code
Key Validations:	Org Code / Business Unit Code must be a valid combination. Meaning the store must reside under the specified organization AOH Code / Parent Account Code must be a valid combination. The Account being associated to the Business Unit cannot have any child accounts. Meaning the account must be at lowest level. This can be at level 2, 3, 4 or lower, so long as there are no additional children parented to that account

Note: Partial and Simulation modes are supported for this process flow.

Step 3: Refresh Reference Materialized Views

Once all Alternate Organization Hierarchy reference data is loaded, you will need to refresh reference materialized views. Materialized views are database objects that are used to aggregate data. A view might be created by joining several tables and aggregating measures by summarization, averaging, etc. With materialized views, the performance-intensive operations (for example joining or grouping) are done when the view is refreshed. The materialized view process flows are located under the "Other Process Flows" module. They are broken down into two groups:

- The MV_COM (Materialized View Complete) folder contains process flows which populate the materialized views from scratch.
- The MV_FAST (Materialized View Fast) folder contains process flows which will update existing views and apply changes made since the materialized view was last updated.

When using the Alternate Organization Hierarchy feature you must perform a complete refresh. Fast refreshes are not supported. You can either refresh the reference data at this time, or you can postpone the refresh of the reference materialized views until all facts are loaded and then refresh both the reference and fact materialized views at the same time.

Note: If you have syndicated data that you want to associate to an account in the Alternate Organization, and if that syndicated data provider uses a geography key, you will need to load the Syndicated Geography Key cross-reference information. The Syndicated Geography Key Process Flow loads data which is used to translate the syndicated provider's geography key to the appropriate dimension values (market area, region, AOH Account, etc.) that are meaningful to DSR.

When you execute the Syndicated Geography Key process flow, the system reads data from the source table, performs the necessary validations and populates the target table listed here.

Process Flow Name:	SYND_GEO
Source Table(s):	DDR_I_SYND_GEO_KEY
Target Table(s):	DDR_R_SYND_GEO_KEY
Dependencies:	customer organization hierarchy market Area geography region and/or sub-region channel source AOH account

Step 4: Load Syndicated Geography Key

Before running the syndicated geography key process flow, you must load any dimension data to which the syndicated geography is related. This includes:

- Customer Organization Hierarchy
- Market Areas

- Geography regions and/or sub-regions
- Channel Types
- Source codes.
- AOH Account Information

Required Fields:	Source
	Geography Key
	Geography Name
Key Validations:	Source must be valid
	Organization Hierarchy and AOH Account are mutually exclusive
	Org Hierarchy Level must be valid
	ORGANIZATION, CHAIN, AREA, REGION, DISTRICT, BU
	AOH Code / Account code combination must be valid
	Channel Type, Market Area, Geography Region, Geography Sub-region (if populated) must be valid
	Org Hierarchy Level / Code, Market Area, Geography Region, Sub-region cannot all be null
	Market Area, Geography Region, Geography Sub-region are mutually exclusive

Note: Partial and Simulation modes are supported. Changes to geography key definition may require manual clean up or re-loading of consumption data. Syndicated Geography Key deleted when Process Action = 'D'

Step 5: Load Syndicated Consumption Data

The Syndicated Consumption Data Process Flow loads external aggregated consumption (for example, sales) measures received from syndicated data providers. By default, the measures do not have any aggregation rule and can only be viewed at the level they were loaded.

Before running the syndicated consumption process flow, you must load any dimension data to which the syndicated consumption data is related. This includes:

- Manufacturer item hierarchy
- Customer organization hierarchy
- Market area
- Geographic region or sub-region
- Manufacturer business week
- Channel Type, Measure Set, and Source code (in look up master)
- AOH Account information

Process Flow Name:	SYND_CNSMPTN_INF_STG_TO_TGT
Source Table(s):	DDR_I_SYND_CNSMPTN_DATA
Target Table(s):	DDR_B_SYND_CNSMPTN_DATA
Dependencies:	Manufacturer item hierarchy Customer organization hierarchy Market Area Geographic region or sub-region Manufacturer Business Week Channel Type Measure Set Source AOH Account
Required Fields:	Record ID Source Code Measure Set Week End Date

Key Validations:	<p>Source Code and Measure Set must be valid</p> <p>Syndicated product and geography keys must be valid (if populated)</p> <p>Either Syndicated Product Key or Item Hierarchy Level / Item Hierarchy Codes must be populated</p> <p>Either Syndicated Geography Key or one of the following combinations must be populated:</p> <ul style="list-style-type: none"> • Org Hchy Level / Org Hchy Code / Channel • Org Hchy Level / Org Hchy Code / Market Area / Channel • Geography Region or Sub-Region / Channel • Market Area / Channel • AOH Code / AOH Account
Additional Information:	<p>All (or None), Partial, and Simulation modes are supported.</p> <p>All consumption data is assumed to be at week level.</p> <p>No currency conversion.</p> <p>No allocation to lower dimensional levels.</p> <p>Default aggregation rule is none.</p>

Step 6: Refresh Fact Materialized Views

Once all fact data is loaded, you will need to refresh reference materialized views. This step must be completed for the data to appear on the distributor dashboard. You can choose to perform either a complete refresh of the fact data or a fast refresh. If you postponed refreshing the reference data materialized views, you must refresh both reference and fact materialized views at this time.

To refresh fact materialized views, use one of the following process flows:

- FACT_MV_COMPLETE_REFRESH
- FACT_MV_FAST_REFRESH

Note: There is no interface table to load, running the process flows are

simply a matter of starting the process flow.

Step 7: View Dashboard

The last step is to view the Alternate Organization Hierarchy and associated facts in OBIEE dashboards and reports. DSR has provided a sample out of the box dashboard which utilizes the Alternate Organization Hierarchy. Existing dashboards and reports can be easily modified to use the alternate organization hierarchy instead of the Customer Organization Hierarchy.

The Sales Performance by Account dashboard page can be found on the miscellaneous reports dashboard.

By drilling down on the level two account, you can see sales and forecast facts broken down by level three account. You can continue to drill down through the levels until you reach the individual business units (stores or distribution centers).

Time and Organization Allocations

Point of Sale and sales forecast fact data is imported to Oracle Demand Signal Repository at an aggregate level. This data is allocated within DSR to the appropriate business unit and day level. With time allocation, weekly Point of Sale and Sales Forecast data is allocated as a percentage to each day of the week. Similarly, organization allocation assigns, at any level of the retail organization, percentages to each associated business unit.

Allocation percentages are loaded into DSR using the existing method for loading interface tables. These percentages are visible from the BI Answers catalog, so that end users can view the percentages that were used when allocating sales facts to the business unit or day level. For more information on loading allocations, see Data Loading Interface Tables.

Time Allocation

Time allocation assigns percentages to each day of the week to allocate weekly Point of Sale and Sales Forecast data to a specific day. The following table is an example of time allocation data:

BSNS_UNIT_CD	WK_NBR	PRCNT_ALLOC	STATUS_IND
Goodway #001	1	20	A
Goodway #001	2	20	A

Goodway #001	3	15	A
Goodway #001	4	10	A
Goodway #001	5	10	A
Goodway #001	6	10	A
Goodway #001	7	15	A
Goodway #001	1	15	
Goodway #001	2	15	
Goodway #001	3	15	
Goodway #001	4	15	
Goodway #001	5	15	
Goodway #001	6	15	
Goodway #001	7	10	

The WK_NBR column represents a day of the week relative to the Point of Sale or Sales Forecast data's transaction date. For example, If weekly point of sales data has a transaction of November 23rd (a Sunday), based on the rules specified above, the system will allocate 20% of sales to Sunday, November 23, 20% of sales to Saturday, November 22nd, 15% of sales to Friday, November 21st, and so on.

DSR expects that weekly point of sales and sales forecast from a single business unit will always have a transaction date which refers to the same day of the week (for example, always a Sunday's date, or always a Friday's date, etc.) Weekly data should normally follow the retailer's business week, so that the transaction date would always reference the week ending date.

Note: Only records with a status indicator of A (Active) are used to allocate point of sale or sales forecast data.

Organization Allocation

Organization allocation assigns percentages to specified business units, in order to allocate Point of Sale and Sales Forecast data at any level of the retail organization

hierarchy. The following table is an example of organization allocation data:

ORG_CD	BSNS_UNIT_CD	PRCNT_ALL OC	LOC_IDNT_CD	ORG_LVL_CD
GW	Goodway #001	100	CA	DISTRICT
GW	Goodway #002	100	OR	DISTRICT
GW	Goodway #003	40	TX	DISTRICT
GW	Goodway #004	60	TX	DISTRICT
GW	Goodway #005	100	AR	DISTRICT
BM	BigMart #001	40	Pacific	REGION
BM	BigMart #002	60	Pacific	REGION
BM	BigMart #003	25	Plains	REGION
BM	BigMart #004	35	Plains	REGION
BM	BigMart #005	40	Plains	REGION

Notice that the allocation is always from a level in the organization hierarchy directly to the lowest level of the organization hierarchy (Business Unit). If data is received at the region level, the allocation is direct to the business unit, and not to the district (the next level down in the hierarchy).

Data Loading Interface Tables

The data loading capabilities of Demand Signal Repository are divided into multiple layers:

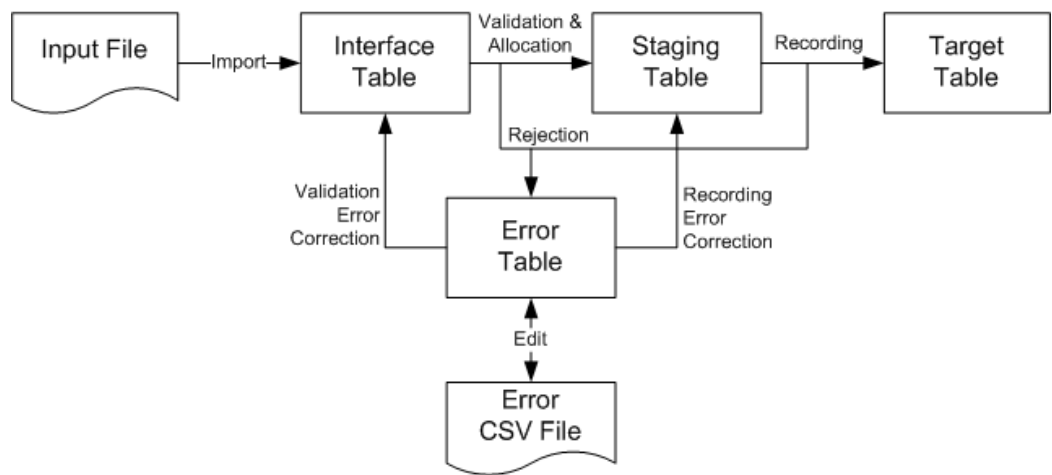
Interface table - If the retailer provides data in a file, the manufacturer imports it into a set of interface tables that correspond to the retailer data sources received. Demand Signal Repository provides a standard interface table layout, and a sample set of input files that match the interface tables field-for-field. A manufacturer has the option creating its own interface table structure that differs from the standard structure that Demand Signal Repository provides.

Staging table – Demand Signal Repository performs validations and allocation on the data in the interface table, and (except in the case of simulation mode) writes values that pass the tests to staging tables that map one-to-one with the target tables in the Demand

Signal Repository database.

Error table – Values in the interface tables that do not pass validation are moved to the error table, accompanied by error messages that identify the cause of the problem. From here, the data can be exported to a CSV file format, where it can be edited to correct any errors, and resubmitted for processing. Corrected values are typically moved back to the interface tables for re-validation, except in the case of errors that were originally encountered in the process of moving from staging to target – these are moved directly to the staging tables for further processing.

Target table – If the mode used to run the data loader is "partial", or the mode was "all or none" and none of the records had errors, data is moved from the staging tables to the target tables in the database, where it becomes visible to Demand Signal Repository users. In the case of "simulation" mode, the records are never moved to the target tables.



Loading Fact Data

Each related group of fact data types that corresponds to the entities in the data model has its own inbound interface table.

Data that accompanies each fact table data load includes:

- **Global Item ID** - A universally unique identifier for the product item, which is used to link each retailer's report of sales results to the corresponding manufacturer's item. Most often, this number will be a 14-digit Global Trade Item Number (GTIN) assigned by the GS1 standards organization. GTINs are a superset of the former UPC and EAN item codes used in the retail industry.
- **Global Item ID Type** – An indicator of the identification scheme used to assign the global item number. By default, the value is "GTIN" if unspecified. The combination of the Global Item Number and the Global Item Number Type forms the key for product item IDs in the system. The manufacturer must take care that UPCs, EAN codes and GTINs that it receives from different retailers have been formatted consistently (UPCs and EAN codes masked with leading zeros, no separator

characters such as "-", presence/absence of check digit). Otherwise, the same item sold at multiple retailers will be identified as different items.

- **Retail Item Number** – The input file may provide a second identifier for the item, which is the retailer's proprietary item number. This number is primarily used for validation, but it can also be used in place of the global item number for retail data sources that lack a global number reference. In cases where only the retail item number is supplied, Demand Signal Repository must already have an item record with an assigned global item number, so the data can be mapped to a unique item on the manufacturer side.

Note: Retailers may sometimes have many-to-one mappings of their item codes (SKUs) to GTINs (due to recycling of SKU codes), as well as one-to-many mappings (because the retailer considers a manufacturer's successor item to be equivalent). If a retail item number is supplied alone in the many-to-one case, it should be accepted; in the one-to-many case, it should be rejected unless the record also includes a global item number that disambiguates the item assignment.

- **Retail Company (Organization) ID** – The global identifier for the legal entity that supplied the data. This is typically a standard 12-digit Global Location Number (GLN) assigned by the GS1 standards organization. However, in environments that have not adopted GLNs, this could also be a 9-digit Dun and Bradstreet company identifier (DUNS number), or a 13-digit DUNS number plus a four-digit local location qualifier (DUNS+4 number). This identifier must be at least 20 characters long, in order to accommodate other proprietary company identification schemes. If a proprietary scheme is used, it is the customer's responsibility to ensure the uniqueness of company IDs.
- **Retail Location (Business Unit) ID** – The global identifier for the location (typically a retail store or distribution center) for which the data is being reported. In the case of data values that apply to the entire retail enterprise, this value could also be the same as the Retail Company ID.

The Retail Location ID is typically a standard 12-digit Global Location Number (GLN) assigned by the GS1 standards organization. However, in environments that have not adopted GLNs, this could also be a 13-digit DUNS number plus a four-digit local location qualifier (DUNS+4 number). This identifier must be at least 20 characters wide, in order to accommodate other proprietary location identification schemes. If a proprietary scheme is used, it is the customer's responsibility to ensure the uniqueness of retail location IDs.

- **Retail Location ID Type** - An indicator of the identification scheme used to assign the Retail Company ID. By default, the value is "GLN" if unspecified. Other permitted values are "DUNS", "DUNS_PLUS_4" and "PRIVATE." The combination

of the Retail Location ID and the Retail Location ID Type forms the key for location IDs in the system.

- Quantities and Amounts – Any quantity or amount provided for each fact is a decimal value with unlimited precision, which may have a positive or negative sign.
- Unit of Measure – The unit of measure to be tied to a quantity is specified using a standard ISO code. The default unit of measure (if unspecified) is "each" (EA).
- Date – For daily data, the date is the day for which the value is being reported. If weekly data is being processed, the date is the ending date for the week.
- Source Date/Time – The date or date/time that the data was created. This is an optional field, which can be used to cross-reference the data with the source system.
- Source System – The source file name or system that provided the data. This is an optional field, which can be used to cross-reference the data with the source system.

Validation

Demand Signal Repository uses two validation methods to reject point of sale data that reports sales for items at stores other than their authorized stores:

- Strict Validation – When strict validation is set, the data loading process will reject any records that contain a global item numbers / business unit (store) combination that has not previously been configured as an authorized item/store. The rejected item and store records will be set aside so they can be reviewed and reloaded after the authorizations are set up for them.
- Item Discovery – When item discovery is set, the data loading process will create a new authorized item/store record in the system the first time that it encounters a global item number at a business unit (store) where it has never previously been reported. A log will record the new item/store combinations that were created as a result of the data load.

Authorized item/store combinations have effective dates. By default, if an authorized item/store combination is created through item discovery, its effective date will be the date that it is created. During the data loading process, Demand Signal Repository provides the following functional validations:

- Data Formatting Errors – Problems with the column count, column order, delimiters or other details of the file format prevents processing.
- Ambiguous Item – The retailer provided data solely in terms of its local item number, and there is more than one GTIN that maps to that local item.

- **Unrecognized Item** – Either one or both of the global item number or retail item numbers do not match any item in the system.
- **Duplicate Item** – A single file contains two or more records with the same Global Item Number or Retail Item Number for the same data type, date and location. In this case, both records should be rejected.

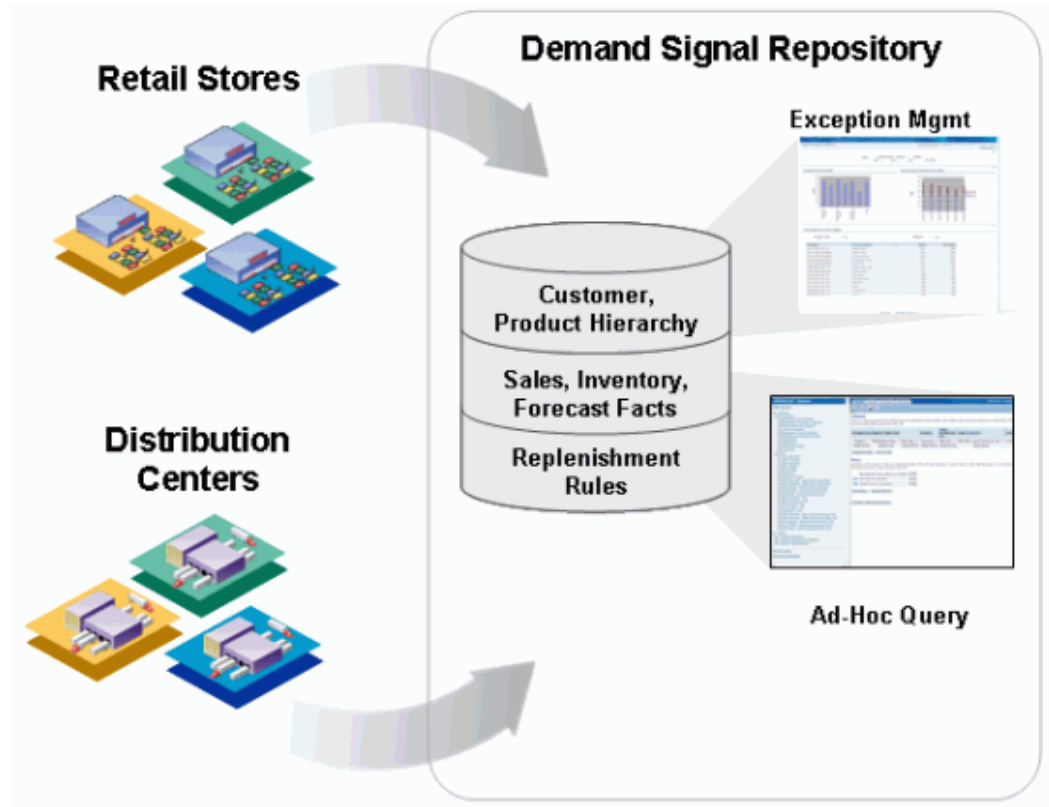
Note: This validation is optional, and switched off by default, due to potential performance concerns.

- **Unrecognized Location** – The retail location ID does not match any location ID currently in the system.
- **ID Type Mismatch** – An Item ID Type, Retail Company ID Type or Retail Location ID type is provided that does not match the type currently listed for an item in Demand Signal Repository, and item discovery is not set.
- **Invalid Selling Location** – Strict validation is set and an item was reported as being sold at a location that is not in its authorized list.
- **No Allocation Profile Found** – Data was loaded at an aggregate level, but no allocation profile was defined for the organization and/or week.

Replenishment Rules

Some retailers provide information to suppliers (for example, Consumer Goods manufacturers) about the rules used to replenish inventory at store locations. Suppliers use these rules to determine whether out-of-stocks or overstocks are being caused by an inappropriate inventory management policy, or to anticipate future order volumes.

These replenishment rules can be loaded into DSR to provide visibility into the criteria used by the store or distribution center when reordering products. Replenishment rules enable DSR to determine if out-of-stocks or overstocks are being caused by an inappropriate inventory management policy, or to anticipate future order volumes. As well, replenishment rules are used in the Exception Management workbench to calculate over and under stock conditions.



Replenishment Measures

The following Replenishment Rule measures are supported in DSR:

- Reorder quantity
- Reorder interval
- Reorder threshold
- Replenishment method
- Lead time days
- Target days of supply
- Target on hand quantity
- Maximum days supply
- Maximum stock on hand quantity
- Minimum days supply

- Minimum stock on hand quantity

Loading Replenishment Rules

DSR stores replenishment rules in the `DDR_R_RTL_SKU_BU_INV_RL` table. Data is loaded into this table using the existing data loading interfaces. For more information, see Data Loading Interface Tables.

Materialized Views

Materialized views are database objects that are used to aggregate data. For example, a materialized view might be created by joining several tables and then aggregating measures by summarization, averaging, and so on. Materialized views improve system performance, since performance-intensive operations (for example, joining or grouping) are performed when the view is refreshed. Queries can access materialized views to retrieve pre-aggregated measures and provide a faster response time.

The Materialized Views process flows are located in OWB's Other Process Flows module. They are organized in two different groups:

- `MV_COM` (Materialized Views Complete): This folder contains process flows that populate the materialized views from scratch.
- `MV_FAST` (Materialized Views Fast): This folder contains process flows that update existing views and apply changes made since the last update.

Each of the above folders contains three process flows:

- `FACT_MV_COMPLETE_REFRESH/FACT_MV_FAST_REFRESH`: refreshes the fact data materialized views.
- `REFERENCE_MV_COMPLETE_REFRESH/REFERENCE_MV_FAST_REFRESH`: refreshes the reference data materialized views.
- `MV_COMPLETE_REFRESH/MV_FAST_REFRESH`: refreshes both fact and reference data materialized views.

User-Defined Fields

User-defined fields provide flexibility in capturing and displaying additional attributes within Oracle Demand Signal Repository. A set of 20 user-defined fields (ten alphanumeric and 10 numeric) have been added to each of the DSR entities below:

- Manufacturer Item Hierarchy (all levels)
- Retailer Item Hierarchy (all levels)

- Organization Hierarchy (all levels)
- Item Cluster Definition
- Retailer Cluster Definition
- Customer Promotion Plans

The column names and data type for the User-Defined Fields are shown in the table below:

Column Name	Data Type	Column	Data Type
Attribute1	VarChar2 (240)	Attribute11	Number
Attribute2	VarChar2 (240)	Attribute12	Number
Attribute3	VarChar2 (240)	Attribute13	Number
Attribute4	VarChar2 (240)	Attribute14	Number
Attribute5	VarChar2 (240)	Attribute15	Number
Attribute6	VarChar2 (240)	Attribute16	Number
Attribute7	VarChar2 (240)	Attribute17	Number
Attribute8	VarChar2 (240)	Attribute18	Number
Attribute9	VarChar2 (240)	Attribute19	Number
Attribute10	VarChar2 (240)	Attribute20	Number

By default, the first three fields are visible within DSR. Administrators can add or remove the visibility for each field. These user-defined fields do not require customization, since these fields use existing interface tables and Oracle Warehouse Builder process flows.

Viewing User-Defined Fields as Attributes

To view user-defined fields as attributes:

1. Login to OBIEE.

2. Choose DSR Reports.
3. Choose the Reference Folder.
4. Create query or report with the dimension level and related attributes.
5. Display the report.

Forecast Types and Purpose Codes

Retailers provide different types of forecasts to Manufacturers, including base demand without promotional volumes, promotional demand only, or total demand (including both base and promotion data). Administrators may want to configure DSR to display attributes in the forecast sales source file that describe the forecast data provided by the Retailer.

The provided forecast can be to represent demand for:

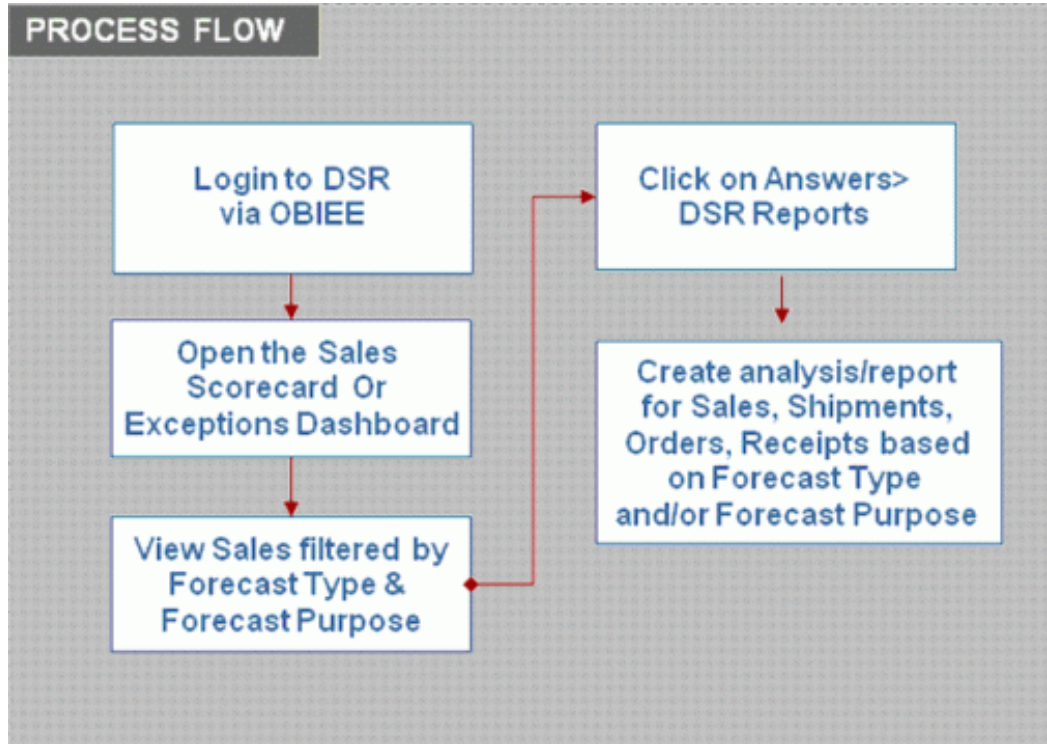
- Projected Orders
- Projected Receipts
- Projected Store Sales
- Projected Shipments

A new Reference folder, named Forecast, contains the new dimensions Forecast Type, Purpose Code. All forecast measures can be displayed and analyzed using the Forecast Type and Forecast Purpose attributes. Reports and analysis can now be done on the components of a forecast in order to analyze the source of the demand. Forecasts can be viewed by specific type and purpose, or by the Forecast's components: Base, Promoted and Total.

Forecast Type and Purpose codes Process Flow

The process to view Manufacturer Promotion data is:

1. Login to OBIEE.
2. Open the Sales Scorecard Or Exceptions Dashboard.
3. View Sales filtered by Forecast Type & Forecast Purpose.
4. Click on Answers>DSR Reports.



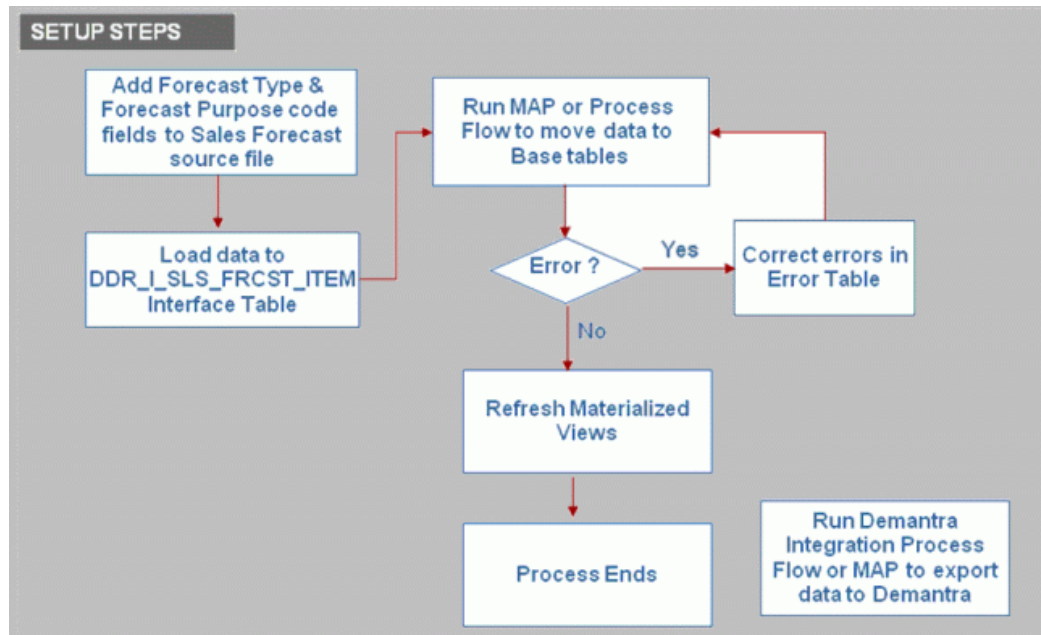
Oracle Warehouse Builder (OWB) contains MAPs and Process Flows to load data from the interface tables into DSR. MAPs move the data from the interface tables into the target or error tables. The Process Flows run the MAPs that load forecast data, including the forecast type and forecast purpose codes.

Forecast Type and Purpose Code Setup Flow

The process to load Forecast Type and Purpose Codes is:

1. Create the source files, or modify existing files, to include the Forecast Type and Forecast Purpose code fields.
2. Load data to the DDR_I_SLS_FRCST_ITEM Interface table.
3. Run MAP or process flow to move data to target tables.
 - Correct any errors.
 - Rerun the MAP or process flow.
4. Refresh the Materialized views.
5. Run the Demantra Integration Process Flow or MAP to export data to Demantra.

Note: This step can be run anytime after the Material Views are refreshed.



Forecast Type Process Flows

DSR uses the following process flows to setup forecast types and purpose codes:

- FORECAST_EF_TO_INF_STG: EXCEL file to interface staging table.
- FORECAST_ERR_TO_INF_STG: Error table to interface staging table.
- FORECAST_INF_STG_TO_TGT: Interface staging table to target.
- DDR_B_SLS_FRCST_ITEM_IS_MAP: Moves data from the interface table to the staging table, we do most of the validations in the Interface to staging maps.
- DDR_B_SLS_FRCST_ITEM_SF_MAP: Moves the data from the staging table to the Final or Base table.

Forecast Type Tables

The tables pertaining to forecast data information are:

- DDR_I_SLS_FRCST_ITEM: Interface staging table.
- DDR_S_SLS_FRCST_ITEM: Staging table

- DDR_B_SLS_FRCST_ITEM: Base table
- DDR_E_SLS_FRCST_ITEM: Error table

Dependencies and Interactions

Point-of-Sale data can be exported from DSR to Demantra. The existing Demantra export process has been modified to consider forecast type and forecast purpose codes. Only data with Forecast Type of 'TOTAL' is exported. The following are the details for the data that is exported:

- Retailer POS Forecast data with a Forecast Type = 'TOTAL' and Forecast Purpose = 'SALES'

Implementation Considerations

The Maps & Process Flows validate the Forecast Type & Purpose Code against the Lookup Master, DDR_R_LKUP_MST. For validation purposes, Frcst_Typ, Frcst_Purpose values have been preseeded in LKUP_TYP_CD and LKUP_CD columns in DDR_R_LKUP_MST. The preseeded values for LKUP_CD are in the chart below:

Type	Purpose
Base	Sales
Promo	Orders
Total	Shipment

Manufacturer Promotions

DSR is able to capture, display, and analyze manufacturer promotions. This gives Manufacturers insight into the reasons for the success or failure of a promotion by providing:

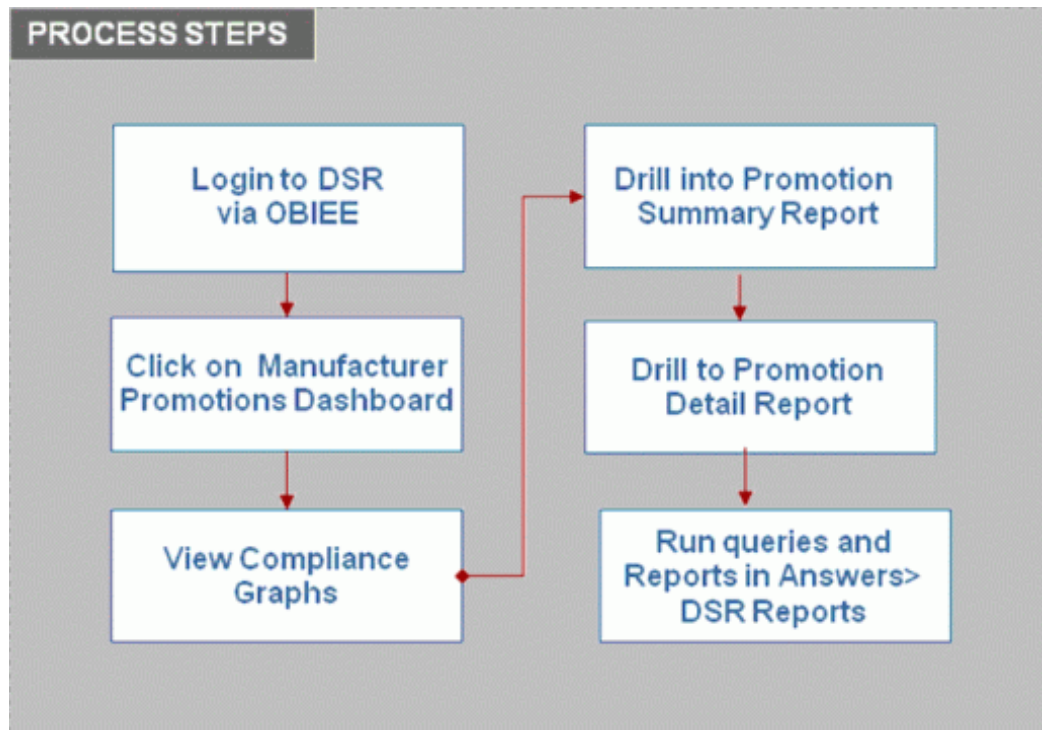
- Promotion compliance (the number or percentage of stores reporting promotional sales).
- Promoted volume (lift) in sales due to promotion activity.
- Items sold on promotion.
- Non-compliant stores (items in a Manufacturer promotion not promoted by the

customer).

- Items sold on promotion other than the Manufacturer offered promotion.

DSR loads manufacturer promotion information using out of the box generic interface tables and Oracle Warehouse Builder process flows. You can associate sales to a manufacturer promotion which provides the capability to report promoted versus non-compliant sales.

Manufacturer Promotions Process Flow



The User process to access Manufacturer Promotions is:

1. Login to DSR via OBIEE..
2. Click on the Manufacturer Promotions Dashboard.
3. View Compliance graphs.
4. Navigate to the Summary reports.
5. Navigate to the Detail report.
6. Run queries and reports in Answers>DSR Reports.

Manufacturer Promotions Setup Flow

The process to load Manufacturer Promotion data is:

1. Ceate the Mfr Promotion reference file and detail file in the required format.
2. Create a record, SLS_TYP, in existing table DDR_R_LKUP_TYP.
3. Add values for the lookup names in table DDR_R_LKUP_MST table (such as Manufacturer Promoted, Manufacturer Non Compliant, Other Promoted, or Regular).
4. Load the files to the relevant interface tables. Note that the Manufacturer must load the data into the interface tables.
5. Run the MAP or the Process Flow to load the data to the Mfr Promotion reference table and check for errors.
 - PRMTN_CD, RTL_ORG_CD and MFG_SKU_ITEM_NBR are checked for duplication and to ensure they are valid.
 - PRMTN_FROM_DT and PRMTN_TO_DT are checked for validity.
 - PRMTN_SHIP_FROM_DT and PRMTN_SHIP_TO_DT are checked for validity.
6. Run the Process Flow to associate the promotions to Customer sales.
7. Run the Process Flow to refresh the Materialized Views.

The values for lookup type, SLS_TYP, are contained in table DDR_R_LKUP_MST. These sales types are used in associating the Manufacturer Promotion data to the Customer's sales data.

LKUP_CD	LKUP_NAME	LKUP_DESC
1	Manufacturer Promoted	POS Sales that were identified by the customer as promotional sales and have been associated to a manufacturer promotion.

2	Manufacturer Non Compliant	POS Sales that were identified by the customer as regular (non-promotional) sales but according to the manufacturer promotion plan criteria should have been promoted.
3	Other Promoted	POS Sales that were identified by the customer as promotional sales but there is no associated manufacturer promotion.
4	Regular	POS Sales that were identified by the customer as regular (non-promotional) sales and there is no associated manufacturer promotion.

Process Flows can be used to load the data in stages to facilitate troubleshooting. These Process Flows allow the Administrator to load the data in stages:

1. Load to the Interface table.
2. Run the Process Flow to move data to the target table.
3. Check error table and correct errors, if any.
4. Run the Process Flow to move corrected data to the interface table.
5. Run the Process Flow to move data to the target table.
 - MFG_PRMTN_ERR_TO_INF moves corrected data from the error table to the interface table.
 - MFG_PRMTN is then used to move corrected data from the interface table to the reference table.

This chart contains the Manufacturer Promotion data available in the DSR tables after the setup and load process:

DDR_R_MFG_PRMTN Table	DDR_R_MFG_PRMTN_DTL Table
------------------------------	----------------------------------

Promotion Code	Promotion Start/End Date
	Ship From/To Date
Promotion Name	Promoted Price
Promotion Description	Expected Base Sales Quantity
	Expected Incremental Sales Quantity
	Expected Total Sales Quantity
Promotion Type	Expected Base Sales Amount
	Expected Incremental Sales Amount
	Expected Total Sales Amount
Promotion Status	Expected ACV %
	Number of Stores Running Promotion
Promotion Dimension Attributes 1 - 20	Promotion Detail Attributes 1 - 20

Manufacturer Promotions are associated to Retailer sales by running the OWB process flow PRM_ASSC>MFG_PRMTN_ASSC. In the INPUT Parameters screen, the PRMTN_ASSC_DAYS parameter determines the number of days prior to the current date to be considered; the default is 14 days.

An item's sales record and promotion flag value are used in determining whether an item should be associated with a loaded Manufacturer's Promotion:

PRMTN_FLAG	PRMTN_ID	SLS_TYP	Description
Y	ID	1	POS Sales that were identified by the customer as promotional sales and have been associated to a manufacturer promotion.

N	ID	2	POS Sales that were identified by the customer as regular (non-promotional) sales but according to the manufacturer promotion plan criteria should have been promoted.
Y	NULL	3	POS Sales that were identified by the customer as promotional sales but there is no associated manufacturer promotion.
N	NULL	4	POS Sales that were identified by the customer as regular (non-promotional) sales and there is no associated manufacturer promotion.

Mapping Criteria

For POS records within a specified date range, a POS record within the Promotion timeframe that matches a SKU and Retailer in the Manufacturer's Promotion Plan will cause the POS record to be mapped to the Promotion. For overlapping promotions, the SKU is mapped to the promotion with the earliest Effective End Date.

Refreshing the Materialized Views

After loading the data and associating the sales to promotions, use one of the following Process Flows to refresh the Materialized Views:

- **ALL_MV_COMPLETE:** performs a complete refresh of all Materialized Views.
- **FACT_MV_COMPLETE:** performs a complete refresh of fact Materialized Views.
- **REFERENCE_MV_COMPLETE:** performs a complete refresh of reference Materialized Views.

- ALL_MV_FAST: performs a net change refresh of all Materialized Views.
- FACT_MV_FAST: performs a net change refresh of fact Materialized Views.
- REFERENCE_MV_FAST: performs a net change refresh of Reference Materialized Views.

Note: The Process Flows used to refresh Materialized Views are in Oracle Warehouse Builder under DSR_OTHER_PROCESS_FLOWS.

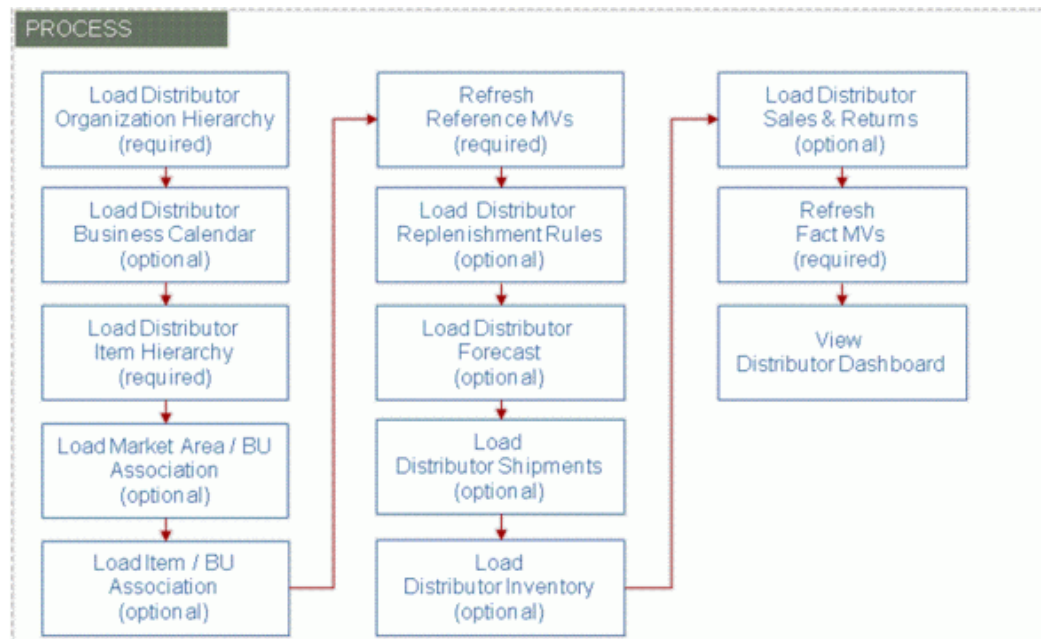
Enabling Third Party Distributors

Manufacturers use Demand Signal Repository to gain visibility into products as they move downstream in the supply chain to the consumer. In previous releases, Demand Signal Repository captured and analyzed data coming direct from retail customers. With this feature, Demand Signal Repository has extended its capabilities to capture data coming from third party distributors.

The business benefits to having third party distributor functionality include:

- Increased visibility into downstream data, regardless of whether the manufacturer ships directly to the customer, or indirectly to the customer using a third party distributor.
- The ability to follow inventory from the third party distributor to retail customers.

Third Party Distributors Process Flow



To use Third Party Distributors, you need to complete the following steps:

1. Load the distributor's organization hierarchy which defines the distributor's distribution centers.
2. Load the distributor's business calendar, if you'll be creating reports which use this calendar.
3. Load the distributor's item hierarchy which defines the items in the terms that the distributor recognizes.
4. To assign market areas to distribution centers (for market area analysis), you will need to load the Business Unit / Market Area associations.
5. If the discovery mode is turned off and sales or forecast data is being loaded, load the Item/BU associations.
6. Refresh the reference data materialized views.
7. Load the distributor's replenishment rules which contain information on how inventory is replenished.
8. Next, Load distributor shipment forecasts, if available.
9. Optionally, you may load distributor shipment facts. Note that while this step is

optional, the most common facts manufacturer's receive from distributors is shipment facts, so it highly likely that you will be loading these facts.

10. Load distributor inventory information.
11. If you receive sales and return information from the distributor, in which case you can load that information into DSR.
12. Once all fact data is loaded, refresh the fact materialized views which summarizes data to improve performance when running queries.

At this point you are ready to view the Distributor dashboard.

Step 1: Load Distributor Organization Hierarchy

The first step is to load the Distributor Organization Hierarchy. Distributors are loaded similar to retailers except that the Organization Type must be "DST" (for distributor). The organization type is used as a filter in the Distributor Dashboard to distinguish distributors from other organization types.

The ORG_HCHY process flow loads the distributor organization, its related organization hierarchy, and the individual business units (also known as distribution centers). For more information, see "Loading Organizations and Hierarchies".

Step 2: Load Distributor Business Calendar

The next step is to load the Distributor Business Calendar. All existing out-of-the-box DSR reports and dashboards use the manufacturer's business calendar, if you are creating queries which need to use the distributor's business calendar, you will need to load the distributor's business calendar as well.

The calendar is loaded by executing the PL/SQL procedure `DDR_POP_CALENDAR_PKG.Populate_BSNS_Calender`. This procedure creates a specified number of years based on the passed parameters. The populate business calendar procedure performs inserts only. If you need to regenerate a particular business year (because it was created incorrectly in the first place), you will need to truncate the distributor's business calendar before re-executing this procedure. Additional years can be created without having to delete existing years. For more information, see "Loading Organizations and Hierarchies".

Step 3: Load Distributor Item Hierarchy

Use the `CST_ITM_HCHY` process flow to load the Distributor Item Hierarchy. The distributor item hierarchy is loaded in the same manner as retailer item hierarchy.

Step 4: Load Market Area / BU Association

Use the MARKET_AREA_BSNS_UNIT process flow to load the Market Area / Business Unit Association. This is an optional step and is only required if you wish to perform analysis on the distributor's distribution centers by market area. For more information, see "Loading Organizations and Hierarchies".

Step 5: Load Item / BU Association

Use the MARKET_AREA_BSNS_UNIT process flow to load the Item/ Business Unit Association. The Item Business Unit Association process flow loads data that identifies the SKUs each business unit is authorized to sell. When discovery mode (a system parameter) is set to an "N", this table is used when loading sales and returns, and sales forecast fact data. This is an optional step and is only required if the discovery mode system parameter is set to an N (No) and you are planning to load distributor POS or forecast data.

For more information, see "Loading Organizations and Hierarchies".

Step 6: Refresh Reference Materialized Views

Once all distributor reference data is loaded, you will need to refresh reference materialized views. Materialized views are database objects that are used to aggregate data. A view might be created by joining several tables and aggregating measures by summarization, averaging, and so on. With materialized views, performance-intensive operations (like joining and grouping) are done when the view is refreshed and not when the user performs the query.

The materialized view process flows are located under the "Other Process Flows" Module). They are broken down into two groups:

- The MV_COM (Materialized View Complete) folder contains process flows which populate the materialized views from scratch.
- The MV_FAST (Materialized View Fast) folder contains process flows which will update existing views and apply changes made since the materialized view was last updated.

There are three process flows under each (MV_COM and MV_FAST) folder:

- The first process flow listed in each folder (FACT_MV_COMPLETE_REFRESH or FACT_MV_FAST_REFRESH) refreshes fact data materialized views.
- The last process flow listed in each folder (REFERENCE_MV_COMPLETE_REFRESH or REFERENCE_MV_FAST_REFRESH) refreshes reference data materialized views.
- The process flow listed in the middle of each folder (MV_COMPLETE_REFRESH

and MV_FAST_REFRESH) refreshes both reference and fact data materialized views.

You can either refresh the reference data at this time, or you can postpone the refresh of the reference materialized views until all facts are loaded and then refresh both the reference and fact materialized views. Since there is no interface table to load, running the process flows are simply a matter of starting the process flow.

Step 7: Load Distributor Replenishment Rules

Use the SKU_BSNS_INV_RULE process flow to load the distributors' inventory replenishment rules. This step is optional as not every distributor will provide this information. Capturing the customer's inventory replenishment rules by BU / SKU combination provides insight into stocking policies and possible out of stock causes. For more information, see "Replenishment Rules".

Step 8: Load Distributor Forecast

Use the FORECAST_INF_STG_TO_TGT process flow to load forecast information, if it is available from the distributor. The Forecast Purpose Code and Forecast Type code columns are used to differentiate the types of forecasts.

The existing forecast process flow was enhanced to capture the additional information.

Step 9: Load Distributor Shipments - Process Flow Description

Use the SHIPMENT_INF_STG_TO_TGT process flow to load distributor shipments. The Shipments process flow loads shipped units and monetary amount for a given ship-from location (org and BU)/ship-to location (org and BU) /SKU/day combination. For example, a shipment records might represent a total of 250 Units of SKU 123 being shipped from AAA Distribution's west coast distribution center to Bigmart store #100.

When you execute the Shipment process flow, the system reads data from the source table, DSR performs the necessary validations (which we will discuss in a subsequent slide) and populates the target table listed here.

Prior to loading the distribution shipment information, you will need to have loaded business units, both retailer and manufacturer SKUs, and calendar information.

Process Flow Name:	SHIPMENT_INF_STG_TO_TGT
Source Table(s):	DDR_I_RTL_SHIP_ITEM
Target Table(s):	DDR_B_RTL_SHIP_ITEM_DAY

Dependent Table(s):	DDR_R_ORG_BSNS_UNIT DDR_R_RTL_SKU_ITEM DDR_R_MFG_SKU_ITEM DDR_R_DAY
Required Validations:	Mfg Org Code Retailer/Distributor Org Code Business Unit Ship To Org Code Ship To Business Unit Retail SKU Item Number Unit of Measure Transaction Date Global Item ID and Global ID Type are required if Retail SKU Item Number is associated to more than one Global ID / Global ID Type.
Key Validations:	Common validations

Note: When upgrading from a previous release, Ship-To Organization Code and Ship To Business Unit must be populated by:

- Truncating shipment data and reloading
- Custom SQL Script

Step 10: Load Distributor Inventory

Use the INVENTORY_INF_STG_TO_TGT process flow to load inventory information, assuming it is available from the distributor. The inventory facts are captured at a lower level than the other facts within DSR. Inventory facts are captured at the store/SKU/Day/Stocking Location level. For example, if a store stocks a SKU in two different places within the store, DSR can capture the quantity at each stocking location. In addition to On-Hand information, the inventory facts also include, Received, In-Transit, On Quality Hold and On Back Order quantities and amounts.

Step 11: Load Distributor Sales and Returns

Use the SLS_RTRN_INF_STG_TO_TGT process flow to capture sales and return

information at the store/SKU/day/level.

Step 12: Refresh Fact Materialized Views

Use the FACT_MV_COMPLETE_REFRESH and FACT_MV_FAST_REFRESH process flows to refresh the fact materialized views. This step must be completed for the data to appear on the distributor dashboard. You can choose to perform either a complete refresh of the fact data or a fast refresh. If you postponed refreshing the reference data materialized views, you must refresh both reference and fact materialized views at this time.

There is no interface table to load, running the process flows are simply a matter of starting the process flow.

Step 13: View Distributor Dashboard

The last step is to view the data using the Distributor dashboard. The Distributor dashboard provides visibility into distributor shipment, forecast and inventory facts. The dashboard can be filtered by one or more distributors and/or product categories, in addition to a date range.

The Manufacturer Shipments by Distributor chart provides a breakdown of manufacturer shipment amount by distributor.

The Distributor Shipments by Customer chart displays the top n customers based on outbound shipment amount from the distributor(s) to the customer.

The Manufacturer Shipment Trend chart displays a weekly trend of shipments from the manufacturer to the selected distributors. This lets you know whether manufacturer shipments to the selected distributor(s) are on the rise or decline.

The Forecast Accuracy by Distributor chart displays the average shipment forecast accuracy percent by distributor to see how well each distributor forecast shipments.

The Distributor Inventory by Product Category chart displays the monetary value of the on-hand inventory at selected distributors by product category.

The Customer Shipments by Supplier chart displays shipments to a customer broken down by the ship from organization (supplier).

The Distributor Shipment Throughput by Product Category chart compares inbound shipments to the distributor and outbound shipments from the distributor by product category, allowing you to see whether inventory at the distributor is increasing or decreasing by product category.

The Distributor Shipment Throughput Trend displays inbound shipments to the distributor and outbound shipments from the distributor by week. This lets you know whether inventory is passing through the distributor at an equal rate at which it arrives.

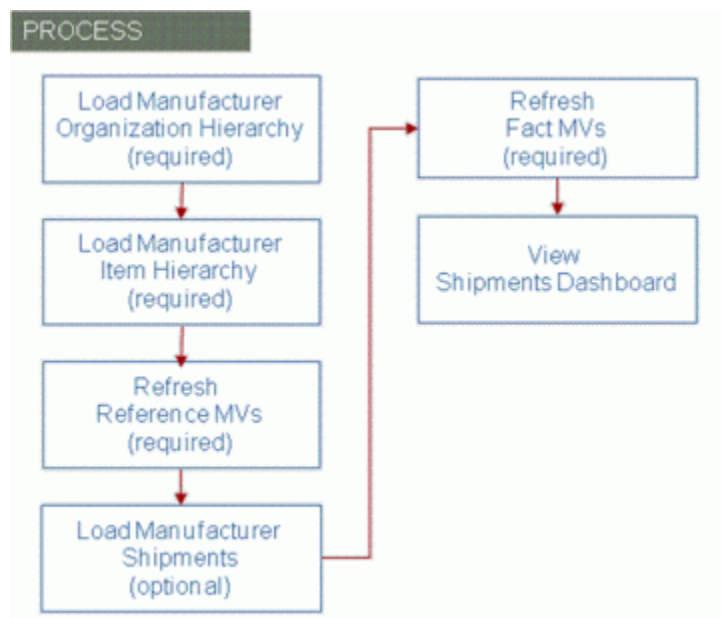
Manufacturer Shipments

Manufacturers use Demand Signal Repository to gain visibility into products as they move downstream in the supply chain to the consumer. In previous releases, Demand Signal Repository captured and analyzed data from retail customers. This feature extends the capture of data to include Manufacturer Shipments.

The business benefits to having Manufacturer Shipments include:

- Ability to analyze manufacturer shipments within Demand Signal Repository.
- Ability to compare supply against demand to determine potential causes of poor performance.

Manufacturer Shipments Process Flow



To use Manufacturer Shipments, you need to complete the following steps:

1. Load the manufacturer's organization hierarchy which defines the manufacturer's shipping locations.
2. Load the manufacturer's item hierarchy.
3. Refresh the reference data materialized views.
4. Load manufacturer shipment facts.
5. Refresh the fact materialized views which summarizes data to improve

performance when running queries.

6. View the Shipments dashboard.

Step 1: Load Manufacturer Organization Hierarchy

Manufacturers are loaded similar to other organizations like retailers and distributors except that the Organization Type must be "MFG" (for manufacturer). It's not necessary to capture all the manufacturer's locations, only the locations from which product is shipped.

The ORG_HCHY process flow loads the manufacturer organization, its related organization hierarchy and the individual business units, or shipping locations.

Step 2: Load Manufacturer Item Hierarchy

Manufacturer item hierarchy is loaded using the MFG_ITEM_HCHY OWB process flow. This is the same method as in previous releases of DSR. No changes were required to support manufacturer shipments.

Step 3: Refresh Reference Materialized Views

Once all reference data is loaded, you will need to refresh reference materialized views. Materialized views are database objects that are used to aggregate data. A view might be created by joining several tables and aggregating measures by summarization, averaging, and so on. With materialized views, performance-intensive operations like joining and grouping are done when the view is refreshed (vs. when the user performs the query).

The materialized view process flows are located under the "Other Process Flows" Module. They are broken down into two groups:

- The MV_COM (Materialized View Complete) folder contains process flows which populate the materialized views from scratch.
- The MV_FAST (Materialized View Fast) folder contains process flows which will update existing views and apply changes made since the materialized view was last updated.

There are three process flows under each (MV_COM and MV_FAST) folder:

- The first process flow listed in each folder (FACT_MV_COMPLETE_REFRESH or FACT_MV_FAST_REFRESH) refreshes fact data materialized views.
- The last process flow listed in each folder (REFERENCE_MV_COMPLETE_REFRESH or REFERENCE_MV_FAST_REFRESH) refreshes reference data materialized views.

- The process flow listed in the middle of each folder (MV_COMPLETE_REFRESH and MV_FAST_REFRESH) refreshes both reference and fact data materialized views.

You can either refresh the reference data at this time, or you can postpone the refresh of the reference materialized views until all facts are loaded and then refresh both the reference and fact materialized views. Since there is no interface table to load, running the process flows are simply a matter of starting the process flow.

Step 4: Load Manufacturer Shipments

The next step is to load manufacturer shipments. Manufacturer shipments reside in a new fact table, separate from customer and distributor shipments. The Manufacturer shipments process flow loads shipped units and monetary amount for a given ship-from location (org and BU)/ship-to location (org and BU) /SKU/day combination. For example, a shipment record might represent a total of 250 units of SKU 123 being shipped from the manufacturer's Houston manufacturing facility to ValueChoice's Texas distribution center.

When you execute the manufacturer's shipment process flow, the system reads data from the source table, performs the necessary validations, and populates the target table. Prior to loading the manufacturer shipment information, you will need to have loaded business units (both ship-from and ship-to locations), manufacturer SKUs, and calendar information.

The Shipments process flow loads shipped units and monetary amount for a given ship-from location/ship-to location /SKU/day combination. For example, 250 units of SKU 123 being shipped from the manufacturer's Houston manufacturing facility to ValueChoice's Texas distribution center.

Process Flow Name:	MFG_SHIP_INF_STG_TO_TGT
Source Table(s):	DDR_I_MFG_SHIP_ITEM
Target Table(s):	DDR_B_MFG_SHIP_ITEM_DAY
Dependent Table(s):	DDR_R_ORG_BSNS_UNIT
	DDR_R_DAY
	DDR_R_MFG_SKU_ITEM

Step 5: Refresh Fact Materialized Views

Once all fact data is loaded (including manufacturer shipments), you will need to refresh fact materialized views. This step must be completed for the data to appear on

the manufacturer dashboard.

You can choose to perform either a complete refresh of the fact data or a fast refresh. If you postponed refreshing the reference data materialized views, you must refresh both reference and fact materialized views at this time. There is no interface table to load, running the process flows are simply a matter of starting the process flow.

Step 6: View manufacturer Dashboard

The last step is to view the data using the shipments dashboard. The shipments dashboard provides visibility into manufacturer and customer shipments. The dashboard can be filtered by one or more customers and/or product categories, in addition to a date range.

The Manufacturer Shipments chart compares year over year shipment history to see if shipments are increasing/decreasing. You can drill down within the chart from month, to week to day.

The Customer Sales chart displays customer sales quantity by month. It compares year over year sales to see if sales are increasing/decreasing. This chart is used to make a side-by side comparison with manufacturer shipments to see any correlation between increase/decrease in manufacturer shipments vs. customer sales. Similar to the Manufacturer Shipments chart, you can drill down from month to week to day.

The Inbound Shipments vs. Receipts by Product Category chart displays inbound customer shipments compared to receipts by product category. Receipt quantities in excess of shipments sent from a distribution center may indicate phantom inventory (products purchased through unauthorized sources). You can drill down within the chart from product category, to sub-category, to item and then individual SKU.

The Inbound Shipments vs. Receipts by Business Unit chart displays inbound customer shipments compared to receipts by product category. Here you can drill to a separate Inbound Shipments vs. Receipts by Business Unit report which breaks down shipments by product category (for the selected business unit).

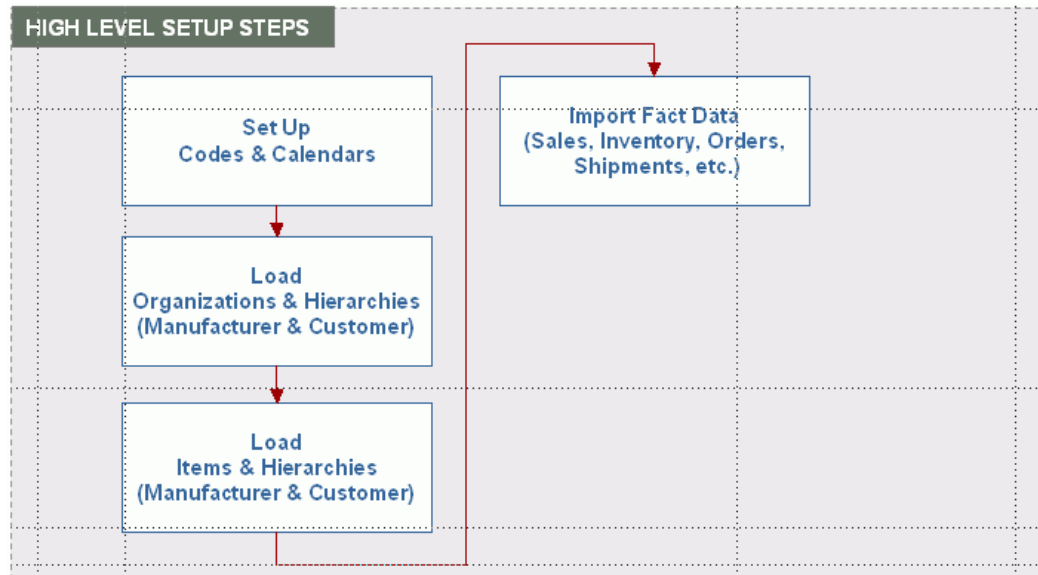
Implementing Demand Signal Repository

Implementation Overview

The DSR implementation can be grouped into these high-level functional areas:

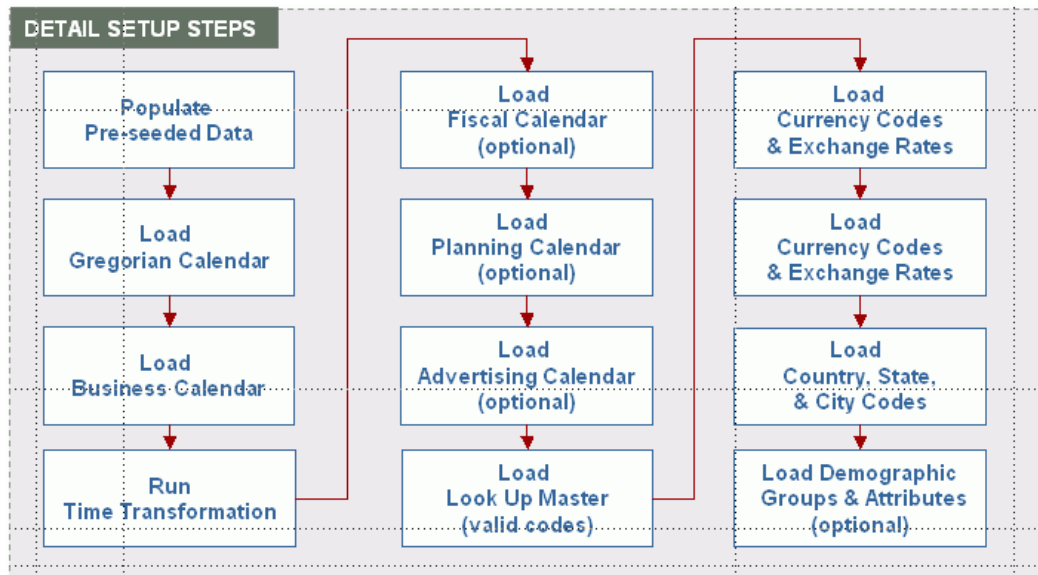
- Initial setup of codes and calendar data.
- Loading of manufacturer and retailer (customer) organizations, hierarchies and related data.
- Loading of manufacturer and retailer (customer) items, hierarchies and related data.
- Importing of fact data, such as sales, inventory, orders, shipments, and so on.

The following diagram illustrates the DSR integration setup steps:



Setting Up Codes and Calendars

The first step required in the DSR implementation is to configure your codes and calendars. The following diagram illustrates the required and optional calendar setup steps:



Step 1: Populate Pre-Seed Data

Run the `DDR_SETUP_UTIL.SETUP` procedure to populate predefined code types and values used by DSR. These types and values include system parameter default values,

error messages, hierarchy level names, and external measure names.

The setup procedure has only one parameter, the Manufacturer's Organization Code, which is used to create a manufacturer's organization record in the organization (DDR_R_ORG) table.

Procedure Name	DDR_SETUP_UTIL.SETUP
Parameters	Mfg Org Code
Source Tables	None
Target Tables	DDR_R_LKUP_TYPE DDR_R_LKUP_MST DDR_R_ORG
Dependent Tables	None
Example	APPS. DDR_SETUP_UTIL.SETUP('VCG');

Step 2: Load Gregorian Calendar

Run the DDR_POP_CALENDAR_PKG.Populate_STND_Calendar procedure to populate the Gregorian (or standard) calendar. The Gregorian calendar is independent of any organization, so it has no dependency to other tables. You must populate the Gregorian calendar before you populate any of the other calendar types (for example, business, fiscal, advertising, and planning) for both manufacturer and retailer.

To populate Gregorian calendar, execute the DDR_POP_CALENDAR_PKG.Populate_STND_Calendar procedure. The first parameter is the number of years you want to create. The second parameter is the starting year.

This procedure only inserts years. It does not update existing data. The procedure checks for the existence of data with respect to the calendar type and year. If the calendar type and year combination already exists, then the procedure reports an error and stops processing.

To add years to the Gregorian calendar, supply only the Number of Years parameter. The second parameter must be null.

Procedure Name	DDR_POP_CALENDAR_PKG.Populate_STND_Calendar
-----------------------	---------------------------------------------

Parameters	Number of Years, Starting Year
Source Tables	None
Target Tables	DDR_R_CLNDR_YR DDR_R_CLNDR_QTR DDR_R_CLNDR_MNTH DDR_R_CLNDR_WK DDR_R_DAYDDR_R_WKDAY
Dependent Tables	None
Example	DDR_POP_CALENDAR_PKG.Populate_STND_Calendar('5,2006');

Step 3: Load Business Calendar

Run the `DDR_POP_CALENDAR_PKG.Populate_BSNS_Calendar` procedure to populate the manufacturer's or customer's business calendar. Before populating any organization-specific calendars, like business, fiscal, advertising, or planning calendars, the corresponding organization record must be available in the organization table (`DDR_R_ORG`).

You must load the manufacturer's business calendar. Loading the customer's (or retailer's) business calendar is optional and is dependent on loading the customer organization first. Once the populate business calendar procedure completes, the tables are populated based on the parameter values that are supplied.

To execute the populate business calendar procedure, you must supply these parameter values:

- The organization code to which the business calendar applies. Initially, this will be the manufacturer's organization code used when loading pre-seeded data (Step 1). After loading customer organizations, you may want to load the customer's business calendar as well.
- Number of years to create.
- Beginning date. This is the first day of the first business year to be created.
- Months that will contain 5 weeks.
- Years that will contain 53 weeks.

- Month that will contain the 53rd week.

This procedure, like the procedure to load the Gregorian calendar, only inserts (or adds) years. If the calendar type and year combination already exists, then the procedure reports an error and stops processing. To add years to the business calendar, supply all but the third parameter, Beginning Date, which must be null.

All of the calendar types (excluding the Gregorian calendar, which we just learned is not associated to any organization) can be associated to the manufacturer; however, only the business calendar can be associated to the customer or retailer.

Procedure Name	DDR_POP_CALENDAR_PKG.Populate_BSNS_Calendar
Parameters	Org Code, Number of Years, Beginning Date, Months with 5 wks, Years with 53 wks, Month with 53rd Week
Source Tables	None
Target Tables	DDR_R_CLNDR DDR_R_BSNS_YR DDR_R_BSNS_QTR DDR_R_BSNS_MNTH DDR_R_BSNS_WK DDR_R_BASE_DAY
Dependent Tables	DDR_R_ORG
Example	DDR_POP_CALENDAR_PKG.Populate_BSNS_Calendar ('VCC', '5', '01-JAN-06', 'M3,M5,M8,M11', '2008', 'M12');

Step 4: Run the Time Transformation Procedure

Run the DDR_TIME_TRANSFORM_PKG.Populate_BSNS_Transformation procedure to group days into business weeks, quarters, and years. This procedure must be run for the date filters in the DSR reports and scorecards to work. This procedure only applies to the business calendar, and it must be re-executed when years are added.

To run the time transformation procedure, you must supply these parameter values:

- Organization code for the business calendar you are transforming
- Beginning year to transform

- Ending year

You can run this procedure multiple times. It will recreate any transformations that were previously created. After adding new years in a business calendar, you should run the transformation for the years that were added.

Procedure Name	DDR_TIME_TRANSFORM_PKG.Populate_BSNS_Transformation
Parameters	Org Code, Beginning Year, Ending Year
Source Tables	DDR_R_BSNS_YR DDR_R_BSNS_QTR DDR_R_BSNS_MNTH DDR_R_BSNS_WK DDR_R_BASE_DAY DDR_R_DAY
Target Tables	DDR_R_DAY_TRANS DDR_R_WK_TRANS DDR_R_MNTH_TRANS DDR_R_QTR_TRANS DDR_R_YR_TRANS DDR_R_DAY_TODATE_TRANS DDR_R_WK_TODATE_TRANS DDR_R_MNTH_TODATE_TRANS DDR_R_QTR_TODATE_TRANS
Dependent Tables	DDR_R_BSNS_YR DDR_R_BSNS_QTR DDR_R_BSNS_MNTH DDR_R_BSNS_WK DDR_R_BASE_DAY DDR_R_DAY

Example	DDR_TIME_TRANSFORM_PKG.Populate_BSNS_Transformation ('VCG', '2006', '2010');
----------------	------------------------------------------------------------------------------

Step 5: (Optional) Load Fiscal Calendar

Run the DDR_POP_CALENDAR_PKG.Populate_FSCL_Calendar procedure to populate the manufacturer's fiscal calendar. This step is optional.

To populate fiscal calendar, you must supply these parameter values:

- Manufacturer's organization code
- Number of years you want to create
- First day of the starting fiscal year

Like the other calendar procedures, this procedure only inserts (or adds) years. If the calendar type and year combination already exists, then the procedure reports an error and stops processing.

To add years to the fiscal calendar, supply only the first two parameters; the Start of Year parameter must be null.

Procedure Name	DDR_POP_CALENDAR_PKG.Populate_FSCL_Calendar
Parameters	Mfg Org Code, Number of Years, Start of Year
Source Tables	None
Target Tables	DDR_R_CLNDR DDR_R_FSCL_YR DDR_R_FSCL_QTR DDR_R_FSCL_MNTH DDR_R_BASE_DAY
Dependent Tables	DDR_R_ORG
Example	DDR_POP_CALENDAR_PKG.Populate_FSCL_Calendar ('VCG', '5', '200601');

Step 6: (Optional) Load Planning Calendar

Run the `DDR_POP_CALENDAR_PKG.Populate_PLNG_Calendar` procedure to populate the planning calendar. This step is optional.

To execute the populate planning procedure, you must supply these parameter values:

- Organization code to which the planning calendar applies
- Number of years to create
- Beginning date (this is the first day of the first planning year to be created)
- Months that will contain 5 weeks
- Years that will contain 53 weeks
- Month that will contain the 53rd week

This procedure, like others that load calendar information, only inserts (or adds) years. If the calendar type or year combination already exists, then the procedure reports an error and stops processing. To add years to the planning calendar, supply all but the third parameter, Beginning Date, which must be null.

Procedure Name	DDR_POP_CALENDAR_PKG.Populate_PLNG_Calendar
Parameters	Org Code, Number of Years, Beginning Date, Months with 5 wks, Years with 53 wks, Month with 53rd Week
Source Tables	None
Target Tables	DDR_R_CLNDR DDR_R_PLNG_YR DDR_R_PLNG_QTR DDR_R_PLNG_PRD DDR_R_PLNG_WK, DDR_R_BASE_DAY
Dependent Tables	DDR_R_ORG
Example	DDR_POP_CALENDAR_PKG.Populate_PLNG_Calendar ('VCG', '5', '01-JAN-06', 'Q1=3, Q2=3, Q3=4, Q4=3', P1=4, P2=4, P3=4, P4=4, P5=4, P6=4, P7=4, P8=4, P9=4, '2008', 'P12');

Step 7: (Optional) Load Advertising Calendar

Run the `DDR_POP_CALENDAR_PKG.Populate_ADVR_Calendar` procedure to populate the Manufacturer's Advertising calendar. This step is optional.

To populate the advertising procedure, supply these parameter values (similar to those for the loading planning calendar):

- Organization code to which the advertising calendar applies
- Number of years to create
- Beginning date (the first day of the first advertising year to be created)
- Months that will contain 5 weeks
- Years that will contain 53 weeks
- Month that will contain the 53rd week

This procedure, like others that load calendar information, only inserts (or adds) years. If the calendar type and year combination already exists, then the procedure reports an error and stops processing. To add years to the advertising calendar, supply all but the third parameter, Beginning Date, which must be null.

Procedure Name	DDR_POP_CALENDAR_PKG.Populate_ADVR_Calendar
Parameter	Mfg Org Code, Number of Years, Start of Year, Qtrs with 5 wks, Periods and Weeks, Starting Year, Period with extra week
Source Tables	None
Target Tables	DDR_R_CLNDR DDR_R_ADVR_YR DDR_R_ADVR_QTR DDR_R_ADVR_PRD DDR_R_ADVR_WK DDR_R_BASE_DAY
Dependent Tables	DDR_R_ORG

Example	DDR_POP_CALENDAR_PKG.Populate_ADVR_Calendar ('VCG', '5', 'Q1=3, Q2=3, Q3=4, Q4=3', P1=4, P2=4, P3=4, P4=4, P5=4, P6=4, P7=4, P8=4, P9=4');
----------------	--------------------------------------------------------------------------------------------------------------------------------------------

Step 8: Load Lookup Masters

Run the DDR_LKUP_MAIN_PF procedure to load any required lookup values. You must load additional lookup values before you can load organizations or items. Lookup values include Global Location ID Types (for example, GLID), Global Item ID Types (for example, GTIN, EAN, UPC), and Address Types (for example, Store Add and Mailing Add).

Loading lookup information is accomplished by executing an Oracle Warehouse Builder (OWB) process flow. The appropriate lookup values must first be loaded into the source table, in this case, the Look Up Interface table (DDR_I_LKUP). When you run (or start) the OWB Look Up process flow, the system reads the source table and populates the lookup master table.

Before running the Look Up process flow, you must load the manufacturer's organization code and Look Up Types (using the setup utility discussed in Step 1).

The following required fields must be populated in the interface table:

- Manufacturer's Organization Code (MFG_ORG_CD)
- Look Up Type Code (LKUP_TYP_CD)
- Look Up Code (LKUP_CD)

If any of these fields is not populated, an error is written to the error table.

The following key validations are required:

- Manufacturer's Organization must be valid.
- Look Up Type Code must be valid.

Procedure Name	DDR_LKUP_MAIN_PF
Parameter	Manufacturer's Organization, Look Up Type Code, Look Up Code
Source Tables	DDR_I_LKUP
Target Tables	DDR_R_LKUP_MST

Dependent Tables	DDR_R_ORG
	DDR_R_LKUP_TYP

Step 9: Load Currency Codes

Run the DDR_ETL_UTIL_PKG.Import Data procedure to load currency codes and exchange rates. Currency code is a required column in some of the subsequent process flows, so even if all data is in a single currency, you must set up at least one currency code. Additional currency codes are required if customer data is received in other currencies.

This procedure has no process flow. Currencies are loaded directly into the Demand Signal Repository target table (DDR_R_CRNCY). DSR provides a generic import data procedure that can be used to load currency codes, as well as other data, into DSR. After the procedure completes, data is loaded from the specified tab-delimited text file into the currency code table (DDR_R_CRNCY).

To populate currency codes using the generic import data procedure, you must specify two parameter values. The first parameter is the target table to be populated. The second parameter is the source table, which must be a tab-delimited text file.

To successfully process the tab-delimited text file, you must ensure that the first row contains column headings that match the database column names of the target table. Also, date fields must be in year, month, day format (yyyy-mm-dd).

This procedure only inserts data; it does not update existing data. You will receive a unique constraint error if you attempt to load a record that already exists.

Note: You also can use this import data procedure to load data into any DSR interface table before running the associated OWB process flow.

Procedure Name	DDR_ETL_UTIL_PKG.Import Data
Parameter	Target Table, Source Table
Source Tables	Text (tab delimited) .txt file
Target Tables	DDR_R_CRNCY
Dependent Tables	DDR_R_ORG

Example	DDR_ETL_UTIL_PKG.Import Data ('DDR_R_CRNCY', 'DDR_R_CRNCY.txt');
----------------	---------------------------------------------------------------------

Step 10: Load Currency Exchange Rate

The DDR_CRNCY_EXC__RATE_MAIN_PF Currency Exchange Rate process flow populates exchange rates between two currencies for a given day so that retailer monetary amounts can be converted to the manufacturer's currency.

These required fields must be populated in the interface table:

- From currency (FROM_CRNCY_CD)
- To currency (TO_CRNCY_CD)
- Calendar Date (CLNDR_DT)
- Exchange Rate (EXCHNG_RATE)

If any of these fields is not populated, an error is written to the error table.

The following key validations are required:

- From and To Currency Codes must be valid (in table DDR_R_CRNCY)
- Date must be valid (based on Gregorian Calendar)

Procedure Name	DDR_CRNCY_EXC__RATE_MAIN_PF
Source Tables	DDR_I_CRNCY_EXCHNG_RATE
Target Tables	DDR_B_CRNCY_EXCHNG_RATE
Dependent Tables	DDR_R_DAY DDR_R_CRNCY

Step 11: Load Country Codes

Load country codes using the DDR_ETL_UTIL_PKG.Import Data procedure. This procedure loads data from a tab-delimited file into the specified target table.

The same rules for formatting the tab-delimited text apply, which means the first row must contain the appropriate column headings and any date fields must be in yyyy-mm-dd format. This procedure only inserts data; it does not update existing data.

You will receive a unique constraint error if you attempt to load a record that already exists.

Procedure Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab-delimited) .txt file
Target Tables	DDR_R_CNTRY
Dependent Tables	DDR_R_ORG

Step 12: Load State Codes

Load state codes using the DDR_ETL_UTIL_PKG.Import Data procedure. This procedure loads data from a tab-delimited file into the specified target table.

The same rules for formatting the tab-delimited text apply, which means the first row must contain the appropriate column headings and any date fields must be in yyyy-mm-dd format. This procedure only inserts data; it does not update existing data. You will receive a unique constraint error if you attempt to load a record that already exists.

Procedure Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab-delimited) .txt file
Target Tables	DDR_R_STATE
Dependent Tables	DDR_R_ORG DDR_R_CNTRY

Step 13: Load City Codes

Load city codes using the DDR_ETL_UTIL_PKG.Import Data procedure. This procedure loads data from a tab-delimited file into the specified target table.

The same rules for formatting the tab-delimited text apply, which means the first row must contain the appropriate column headings and any date fields must be in yyyy-mm-dd format. This procedure only inserts data; it does not update existing data. You will receive a unique constraint error if you attempt to load a record that already exists.

Procedure Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab-delimited) .txt file
Target Tables	DDR_R_CITY
Dependent Tables	DDR_R_ORG

Step 14: Load Organization Demographic Groups

Organization demographic groups (and their associated attribute values) can be associated to customer business units. Load demographic groups using the DDR_ETL_UTIL_PKG.Import Data procedure, which loads data from a tab-delimited file into the specified target table.

Organization demographic groups (and their associated attribute values) are assigned to business units. For example, organization demographic groups can be used to create a group called Retail Floor Space. Attributes, such as Large, Medium, and Small can then be assigned to the Retail Floor Space demographic group. When loading a retail organization, you can then specify that a store have a value of Medium for the Retail Floor Space. In addition, you can capture an associated attribute value, such as 18,000 SQFT (square feet).

Procedure Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab-delimited) .txt file
Target Tables	DDR_R_ORG_DEMOG_GRP
Dependent Tables	DDR_R_ORG

Step 15: Load Demographic Attributes

Organization demographic attributes are associated demographic groups and can be assigned to customer business units. Load demographic attributes using the DDR_ETL_UTIL_PKG.Import Data procedure, which loads data from a tab-delimited file into the specified target table.

Procedure Name	DDR_ETL_UTIL_PKG.Import Data
-----------------------	------------------------------

Source Tables	Text (tab-delimited) .txt file
Target Tables	DRR_R_ORG_DEMOG_ATTR
Dependent Tables	DDR_R_ORG

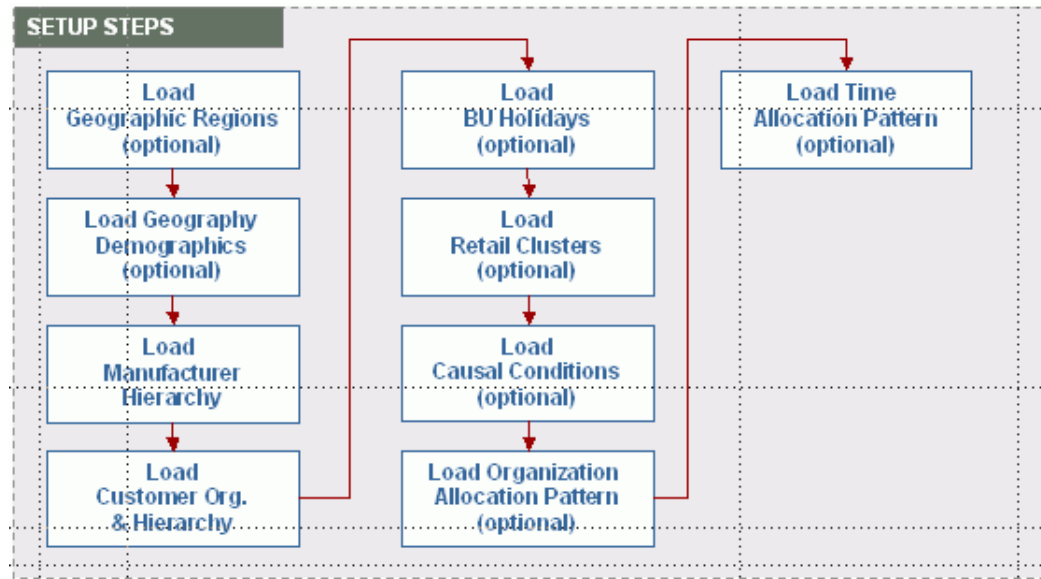
Loading Organizations and Hierarchies

Within DSR, an organization generally equates to a company. Organization types supported by DSR include:

- The manufacturer (company that owns the DSR)
- Customers (also called retailers or distributors)
- Competitors (companies used to analyze all sales within a category, including competitor products)

DSR maintains a customer organizational hierarchy to allow customer data to be summarized based on the customer's view of its organization. Customers with common characteristics can be grouped into Retail Clusters for further analysis.

The following diagram illustrates the required and optional organization and hierarchy setup steps:



Step 1: (Optional) Load Geographic Regions

The Geographic Region Hierarchy process flow captures the manufacturer's hierarchical view of geographic region to sub-regions. It is used to aggregate data by geographical area. This step is optional.

When you execute the Geographic Region Hierarchy process flow, the system reads the source table and populates the geographic region and subregion target tables listed. Before running the Geography Region Process flow, you must load the Manufacturer's Organization Code into the organization table (via the setup utility).

The following fields are required and must be populated in the interface table:

- Manufacturer's Organization Code (MFG_ORG_CD)
- Region Code (RGN_CD)
- Sub-region Code (SUB_RGN_CD)

If any of these fields is not populated, an error is written to the error table.

You must populate both levels of the Geographic Region hierarchy. For example, if a manufacturer uses only one level for Geographic Region (for example, sub-region), then the manufacturer must populate the data for the sub-region into the region level of the Interface table.

The following key validation is required:

- Manufacturer's Organization must be valid.

Process Flow Name	GEOGRPY_RGN_HCHY
Source Tables	DDR_I_GEO_RGN_HCHY
Target Tables	DDR_R_GEO_RGN DDR_R_GEO_SUB_RGN
Dependent Tables	DDR_R_ORG

Step 2: (Optional) Load Geography Regions

The Geography Demographics process flow captures demographic information like population, household, education, and so on and associates it to a geographic code (which is distinct from region or sub-region). When loading retail business units (stores, distribution centers, and so on.) you can associate a geography code (GEO_CD) with the address. This code links the geography demographics to a location. Loading of geography demographic data is optional.

When you execute the Geography Demographics process flow, the system reads the source table and populates the target table listed. Before running the Geography Demographics process flow, you must load the Manufacturer's Organization Code into the organization table.

The following fields are required in the interface table (DDR_I_GEO_DEMOG):

- Manufacturer's Organization Code (MFG_ORG_CD)
- Geography code (GEO_CD)
- Geography Value (GEO_VAL) cannot be null

The following key validation is required:

- Manufacturer's Organization must be set up in the organization table (DDR_R_ORG)

Process Flow Name	GEOGRPC_DEMOGRPY
Source Tables	DDR_I_GEO_DEMOG
Target Tables	DDR_R_GEO_DEMOG
Dependent Tables	DDR_R_ORG

Step 3: Load Manufacturer Hierarchy

The Organization Hierarchy (DDR_ORG_HCHY_MAIN_PF) process flow captures Manufacturer and Retailer Organization information, including valid locations (business units), location types (store, distribution center, and so on), and organization hierarchy information. Competitor organizations are also captured, but not competitor business unit and store information.

Competitor information is loaded only to the organization table. The system does not load competitor hierarchy or competitor business unit information (such as stores and distribution centers).

Step 4: Load Customer Organization and Hierarchy

The Organization Hierarchy process flow (ORG_HCHY) captures Manufacturer and Retailer Organization information, including valid locations (or business units), location types (store, distribution center, and so on), and organization hierarchy information. Competitor organizations are also captured, but not competitor business unit and store information.

Competitor information is loaded only in the organization table. The system does not load competitor hierarchy or competitor business unit information (such as stores and distribution centers).

Three source tables are loaded with organizational data. The main organization source table (DDR_I_ORG) identifies each organization and whether the organization is a manufacturer, retailer, or competitor organization. Based on organization type, the process flow writes a retail or manufacturer item company record.

The organization hierarchy source table (DDR_I_ORG_HCHY) associates business units to districts, regions, areas, and chains. Retail organization hierarchy values can be different from the manufacturer's organization hierarchy.

The organization business unit source table (DDR_I_ORG_BSNS_UNIT) describes the business units (stores or distribution centers) or other user-defined business unit types and their related attributes. This information includes the address of the business unit, valid inventory locations, and attributes such as number of external signs, number of checkout counters, and so on.

These fields are required for the Organization table (DDR_I_ORG):

- Manufacturer's Organization Code (MFG_ORG_CD)
- Organization Code (ORG_CD)
- Organization Type (ORG_TYP)
- Global Location ID (GLBL_LOC_ID)

- Global Location Type (GLBL_LOC_ID_TYP)

If any of these fields is not populated, the system writes an error to the error table.

These fields are required for the Organization Hierarchy table (DDR_I_ORG_HCHY):

- Chain Code, Chain Global Location ID, and Chain Global Location ID Type (CHAIN_CD, CHAIN_GLBL_LOC_ID, CHAIN_GLBL_LOC_ID_TYP)
- Area Code, Area Global Location ID, and Area Global Location ID type (AREA_CD, AREA_GLBL_LOC_ID, AREA_GLBL_LOC_ID_TYP)
- Region Code, Region Global Location ID, and Region Global Location ID type (RGN_CD, RGN_GLBL_LOC_ID, RGN_GLBL_LOC_ID_TYP)
- District Code, District Global Location ID, and District Global Location ID type (DSTRCT_CD, DSTRCT_GLBL_LOC_ID, DSTRCT_GLBL_LOC_ID_TYP)

These fields are required for the Organization Business Unit table (DDR_I_BSNS_UNIT):

- Business Unit Code, Business Unit Global Location ID, and Business Unit Global Location ID type (BSNS_UNIT_CD, BSNS_UNIT_GLBL_LOC_ID and BSNS_UNIT_GLBL_LOC_ID_TYP)

The following key validations are required:

- Manufacturer's Organization must be set up in the organization table (DDR_R_ORG).
- Organization type must be RTL (Retailer), CMP (Competitor), or MFG (Manufacturer).
- All levels of the hierarchy must be populated.
- SCD type 2 loading is used for all the organization hierarchy tables except the upper-level organization table.

Process Flow Name	ORG_HCHY
Source Tables	DDR_I_ORG
	DDR_I_ORG_HCHY
	DDR_I_ORG_BSNS_UNIT

Target Tables	DDR_R_ORG
	DDR_R_MFG_ITEM_CMPNY
	DDR_R_RTL_ITEM_CMPNY
	DDR_R_ORG_CHAIN
	DDR_R_ORG_AREA
	DDR_R_ORG_RGN
	DDR_R_ORG_DSTRCT
	DDR_R_ORG_BSNS_UNIT
	DDR_R_ORG_BSNS_UNIT
	DDR_R_ADDR_LOC
	DDR_R_ORG_DEMOG_VAL
	DDR_R_INV_LOC
Dependent Tables	DDR_R_GEO_SUB_RGN
	DDR_R_GEO_RGN
	DDR_R_GEO_DEMOG
	DDR_R_ORG_DEMOG_GRP
	DDR_R_ORG_DEMOG_ATTR
	DDR_R_GEO_SUB_RGN
	DDR_R_GEO_RGN
	DDR_R_GEO_DEMOG
	DDR_R_ORG_DEMOG_GRP
	DDR_R_ORG_DEMOG_ATTR

Step 5: Load BU Holidays

The Business Unit Holidays ETL captures the days in which a business unit (for example, store, distribution center) is not open for business. When you execute the business unit holidays process flow (BUNIT_HDAY), the system reads the source table and populates the target table listed.

The following fields are required in the interface table:

- Organization Code (MFG_ORG_CD)

- Business Unit Code (BSNS_UNIT_CD)
- Calendar Date (CLNDR_DT)

If any of these fields is not populated, the system writes an error to the error table.

The following key validations are required:

- Organization must be set up in the organization table (DDR_R_ORG).
- Date must be valid (DDR_R_DAY).

Note: The Business Unit Holidays process flow does not support removing a holiday date from the Business Unit Holiday table.

Process Flow Name	BUNIT_HDAY
Source Tables	DDR_I_CAUSAL_CNDTN
Target Tables	DDR_R_CAUSAL_CNDTN
Dependent Tables	DDR_R_ORG DDR_R_ORG_BSNS_UNIT DDR_R_DAY

Step 6: Load Retail Cluster Codes

Retail clusters are used to group customer organizations (for example, grocery chains versus convenience stores). Load Retail Cluster Codes using the DDR_ETL_UTIL_PKG.Import Data procedure, which loads data from a tab-delimited file into the specified target table.

Source Tables	Text (tab-delimited) .txt file
Target Tables	DDR_R_RTL_CLSTR
Dependent Tables	DDR_R_ORG

Step 7: Load Retail Cluster Associations

Use the Retail Cluster process flow (RTL_CLSTR) to associate customer (retail)

organizations to predefined cluster codes. Once you define cluster codes, you can run the Retail Cluster process flow to associate customer organizations with retail cluster codes. In addition to the retail cluster codes, you must also load the retail organization code into the organization table.

If any of the following fields is not populated, an error is written to the error table:

- Retail Organization Code (RTL_ORG_CD)
- Retail Cluster Code (RTL_CLSTR_CD)

The following key validations are required:

- Retail Organization Code must be valid.
- Retail Cluster must be valid.

Process Flow Name	RTL_CLSTR
Source Tables	DDR_I_RTL_CLSTR
Target Tables	DDR_R_RTL_CLSTR_RTL_ASSC
Dependent Tables	DDR_R_ORG (Retailer) DDR_R_RTL_CLSTR

Step 8: Load Causal Conditions

The Causal Condition process flow (CAUSAL_CNDTN) captures the internal, external, and weather condition data from retailers for different business units on a given day. The Causal Condition process flow captures the internal, external, and weather condition data from retailers for different business units on a given day.

Before running the Causal Conditions process flow, you must load the associated retailer's business unit as well as geography demographics and Gregorian calendar.

If any of the following fields is not populated, an error is written to the error table:

- Retail Organization Code (ORG_CD)
- Business Unit Code (BSNS_UNIT_CD)
- Geography Code (GEO_CD)
- Calendar Date (CLNDR_DT)

The following key validations are required:

- Retail Organization must be valid.
- Retail Business Unit must be valid for the Retailer.
- Geography Code must be valid (in table DDR_R_GEO_DEMOG).
- Calendar date must be valid (in table DDR_R_DAY).

Process Flow Name	CAUSAL_CNDTN
Source Tables	DDR_I_CAUSAL_CNDTN
Target Tables	DDR_R_CAUSAL_CNDTN
Dependent Tables	DDR_R_ORG_BSNS_UNIT
	DDR_R_GEO_DEMOG
	DDR_R_DAY

Step 9: Load Organization Allocation Patterns

If sales (or sales forecast) fact data is received at a higher level than business unit, an organization allocation pattern must be defined to allocate the fact data down to the business units (based on percentage). Load Organization Allocation Patterns using the DDR_ETL_UTIL_PKG.Import Data procedure.

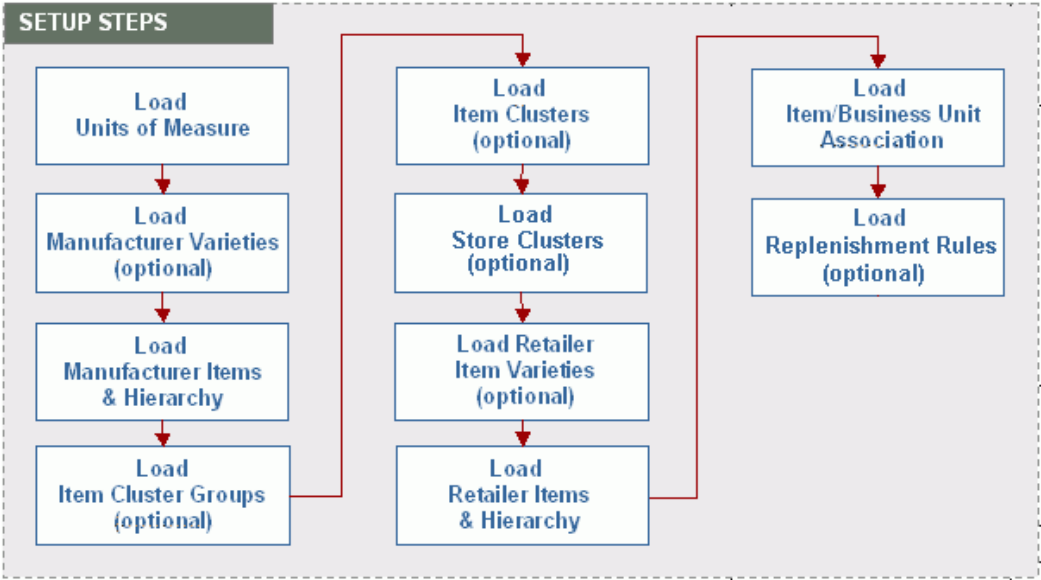
Source Tables	DDR_I_CAUSAL_CNDTN
Target Tables	DDR_R_CAUSAL_CNDTN
Dependent Tables	DDR_R_ORG_BSNS_UNIT
	DDR_R_GEO_DEMOG
	DDR_R_DAY

Step 10: Load Time Allocation Patterns

If sales or sales forecast fact data is received on a weekly (versus daily) basis, a time allocation pattern must be defined to allocate the fact data to the days of the week (based on percentage). Load Time Allocation Patterns using the DDR_ETL_UTIL_PKG.Import Data procedure.

Loading Items and Hierarchies

The following diagram illustrates the required and optional item and hierarchy setup steps:



Step 1: Load Units of Measure

The Unit of Measure process flow (ITM_UOM) defines the valid units of measure (UOM) used by an organization. When you execute the Unit of Measure process flow, the system reads the source table and populates the target table. Before running the Unit of Measure process flow, the Manufacturer's Organization Code must be loaded into the organization table.

If any of the following fields is not populated, an error is written to the error table:

- Manufacturer's Organization Code (MFG_ORG_CD)
- Unit of Measure Code (UOM_CD)

The following key validation is required:

- Manufacturer's Organization must be set up in the organization table (DDR_R_ORG).

Process Flow Name	ITM_UOM
Source Tables	DDR_I_UOM

Target Tables	DDR_R_UOM
Dependent Tables	DDR_R_ORG

Step 2: Load Manufacturer Varieties

Load the manufacturer varieties using the DDR_ETL_UTIL_PKG.Import Data procedure, which loads data from a tab delimited file into the specified target table.

Process Flow Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab delimited) .txt file
Target Tables	DDR_R_MFG_VRTY
Dependent Tables	DDR_R_ORG

Step 3: Load Manufacturer Items and Hierarchy

The Manufacturer Item Hierarchy process flow (MFG_ITM_HCHY) populates all levels of the manufacturer's item hierarchy as well as the manufacturer SKU item information. Some of the SKU item attributes that can be loaded are:

- Price
- Item type
- Color
- Coating
- Fiber
- Weave

Two source tables are loaded with manufacturer item data. The Manufacturer Item Hierarchy source table (DDR_I_MFG_ITEM_HCHY) associates individual SKU items to the various levels of the item hierarchy, including item, subclass, class, group, and division. The Manufacturer SKU Item source table (DDR_I_MFG_SKU_ITEM) identifies the unique SKUs that the manufacturer produces or the unique competitor SKUs for which they are the category captain for a retailer.

Before running the Manufacturer Item Hierarchy process flow, you must load the

manufacturer into the organization table. In addition, Manufacturer Variety table (DDR_R_MFG_VRTY) and UOM codes referenced in the source table must be set up in the UOM tables. Any brand names referenced in the source tables must be set up in the Look Up Master table.

If any of the following fields is not populated, an error is written to the error table. Required fields for Manufacturer Item Hierarchy table (DDR_I_MFG_ITEM_HCHY) are:

- Manufacturer's Organization Code (MFG_ORG_CD)
- Company Code (MFG_CMPNY_CD)
- Division Code (MFG_DIV_CD)
- Group Code (MFG_GRP_CD)
- Class Code (MFG_CLASS_CD)
- Sub-Class Code (MFG_SBC_CD)
- Item Number (MFG_ITEM_NBR)
- SKU Item Number (MFG_SKU_ITEM_NBR)
- Global Item ID (GLBL_ITEM_ID)
- Global Item ID Type (GLBL_ITEM_ID_TYP)

Required fields for Manufacturer SKU Item table (DDR_I_MFG_SKU_ITEM) are:

- SKU Item Number (MFG_SKU_ITEM_NBR)
- Global Item ID (GLBL_ITEM_ID)
- Global Item ID Type (GLBL_ITEM_ID_TYP)
- UOM Code (UOM_CD)
- Alternate UOM Code (ALT_UOM_CD)

The following key validations are required:

- Manufacturer's Organization must be set up in the organization table (DDR_R_ORG).
- Brand, UOM, Alternate UOM codes must be valid.
- All Item Hierarchical Levels must be populated.

- SCD type 2 loading is used for all the item hierarchy tables.
 - SKU Item Type, Color, Coating, Size, Fiber, Fabric, Weave, Style, Stock Item Type and Dye values are automatically loaded into the Look Up Master table.
 - UOM Conversion Factors (from Transaction to Alternate UOM) are automatically loaded into the UOM Conversion table.

Process Flow Name	MFG_ITM_HCHY
Source Tables	DDR_I_MFG_ITEM_HCHY DDR_I_MFG_SKU_ITEM
Target Tables	DDR_R_MFG_ITEM_DIV DDR_R_MFG_ITEM_GRP DDR_R_MFG_ITEM_CLASS DDR_R_MFG_ITEM_SBC DDR_R_MFG_ITEM DDR_R_MFG_SKU_ITEM DDR_R_MFG_SKU_ITEM DDR_R_MFG_SKU_ITEM_VRTY DDR_R_UOM_CNVRSN
Dependent Tables	DDR_R_ORG DDR_R_MFG_ITEM_CMPNY DDR_R_MFG_VRTY DDR_R_UOMDDR_R_LKUP_MST

Step 4: Load Item Cluster Groups

Item clusters are used to group items together. Load Item Cluster Codes using the DDR_ETL_UTIL_PKG.Import Data procedure, which loads data from a tab-delimited file into the specified target table.

Process Flow Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab delimited) .txt file
Target Tables	DDR_R_ITEM_CLSTR
Dependent Tables	DDR_R_ORG

Step 5: Load Item Clusters

Item cluster types and codes are loaded using a generic import data procedure that loads data from a tab delimited file into the specified target table. If any of the following fields is not populated, an error is written to the error table:

- Manufacturer's Organization Code (MFG_ORG_CD)
- Item Cluster Code (ITEM_CLSTR_CD)
- Item Cluster Type (ITEM_CLSTR_TYP)

The following key validations are required:

- Manufacturer's Organization must be set up in the organization table (DDR_R_ORG).
- Cluster Code must be valid (based on DDR_R_ITEM_CLSTR).

Process Flow Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab delimited) .txt file
Target Tables	DDR_R_ITEM_CLSTR
Dependent Tables	DDR_R_ORG

Step 6: Associate SKUs to Cluster

The Item Cluster process flow loads the association between SKUs and item clusters.

Process Flow Name	DDR_ITM_CLUSTER
--------------------------	-----------------

Source Tables	DDR_I_ITEM_CLSTR
Target Tables	DDR_R_ITEM_CLSTR_ASSC
Dependent Tables	DDR_R_ITEM_CLSTR
	DDR_R_MFG_SKU_ITEM

Step 7: Load Store Clusters

The Store Cluster process flow (DDR_STR_CLSTR) loads data which defines the valid store cluster codes for each cluster type and the business units (stores or distribution centers) associated with each store cluster code.

The following key validations are required:

- Store Cluster Type must be set up in the look up master table (DDR_R_LKUP_MST). .
- Store Cluster Code must be valid before being associated to a business unit.
- Organization and Business Unit must be a valid combination.
- Process Action must be a Null, D or R in the Store Cluster interface table, or a Null or D in the Business Unit Association interface table.

Process Flow Name	DDR_STR_CLSTR
Source Tables	DDR_I_STR_CLSTR
	DDR_I_STR_CLSTR_BU_ASSC
Target Tables	DDR_R_STR_CLSTR
	DDR_R_STR_CLSTR_BU_ASSC
Dependent Tables	DDR_R_LKUP_MST
	DDR_ORG_BSNS_UNIT

Step 8: Load Retailer Item Varieties

Load retailer varieties using the DDR_ETL_UTIL_PKG.Import Data procedure, which loads data from a tab-delimited file into the specified target table.

Process Flow Name	DDR_ETL_UTIL_PKG.Import Data
Source Tables	Text (tab delimited) .txt file
Target Tables	DDR_R_RTL_VRTY
Dependent Tables	DDR_R_ORG (Manufacturer & Retailer)

Step 8: Load Retailer Items and Hierarchy

The Retailer Item Hierarchy process flow (RTL_ITM_HCHY) populates all levels of the Retailer's Item Hierarchy as well as the Retailer's SKU item information. The available Retailer SKU Item attributes are similar to the Manufacturer's SKU Item attributes (price, item type, color, coating, fiber, weave, dye, fabric, item style, stock item type, size, and so on).

Two source tables are loaded with retailer item data: Retailer Item Hierarchy source table (DDR_I_RTL_ITEM_HCHY) and Retailer SKU item source table (DDR_I_RTL_SKU_ITEM). The Retailer Item Hierarchy source table associates individual SKU items to the various levels of the retailer's item hierarchy, including item, subclass, class, group, and division.

The Retailer SKU item source table identifies the unique SKUs that the retailer sells, including any competitor SKUs that the manufacturer manages as the category captain. Before running the Retailer Item Hierarchy Process Flow, you must load the Retailer's Organization into the organization table. In addition, varieties must be set up in the Retailer's Variety table (DDR_R_RTL_VRTY) and UOM codes must be set up. Brand names referenced in the source tables must be set up in the Look Up Master table.

If any of the following fields is not populated, the system writes an error to the error table. Required fields for Retailer Item Hierarchy table (DDR_I_RTL_ITEM_HCHY) are:

- Manufacturer's Organization Code (MFG_ORG_CD)
- Company Code (RTL_CMPNY_CD)
- Division Code (RTL_DIV_CD)
- Group Code (RTL_GRP_CD)
- Class Code (RTL_CLASS_CD)
- Sub-Class Code (RTL_SBC_CD)
- Item Number (RTL_ITEM_NBR)

- SKU Item Number (RTL_SKU_ITEM_NBR)
- Global Item ID (GLBL_ITEM_ID)
- Global Item ID Type (GLBL_ITEM_ID_TYP)

Required fields for Retailer SKU Item table (DDR_I_RTL_SKU_ITEM) are:

- Retail Organization (RTL_ORG_CD)
- SKU Item Number (RTL_SKU_ITEM_NBR)
- Global Item ID (GLBL_ITEM_ID)
- Global Item ID Type (GLBL_ITEM_ID_TYP)
- UOM Code (UOM_CD)

The following key validations are required:

- Retailer Organization must be set up in the organization table (DDR_R_ORG).
- Brand and UOM codes must be valid.
- All Item Hierarchical Levels must be populated.
- SCD type 2 loading is used for all the item hierarchy tables.

Note: SKU Item Type, Color, Coating, Size, Fiber, Fabric, Weave, Style, Stock Item Type and Dye values are automatically loaded into the Look Up Master table.

Process Flow Name	RTL_ITM_HCHY
Source Tables	DDR_I_RTL_ITEM_HCHY
	DDR_I_RTL_SKU_ITEM

Target Tables	DDR_R_RTL_ITEM_DIV
	DDR_R_RTL_ITEM_GRP
	DDR_R_RTL_ITEM_CLASS
	DDR_R_RTL_ITEM_SBC
	DDR_R_RTL_ITEM
	DDR_R_RTL_SKU_ITEM
	DDR_R_RTL_SKU_ITEM
	DDR_R_RTL_SKU_ITEM_VRTY
	DDR_R_UOM_CNVRSN
Dependent Tables	DDR_R_ORG
	DDR_R_RTL_ITEM_CMPNY
	DDR_R_RTL_VRTY
	DDR_R_UOMDDR_R_LKUP_MST

Step 9: Load Item/Business Unit Association

The Item Business Unit Association process flow (ITM_BSNS_UNIT_ASC) loads data that identifies the SKUs each business unit is authorized to sell. When the Discovery mode system parameter is cleared, this table is used when validating sales and returns, and sales forecast fact data. When Discovery mode is selected, the fact data process flow inserts a record into this table automatically.

When you execute the item/BU association process flow, the system reads the source table and populates the target table. Before running the Item/Business Unit Association process flow, the Retailer's organization and their related business units, as well as the manufacturer's items must be loaded.

If any of the following fields is not populated, an error is written to the error table:

- Retail Organization Code (RTL_ORG_CD)
- Retail Business Unit Code (RTL_BSNS_UNIT_CD)
- Manufacturer SKU Item Number (MFG_SKU_ITEM_NBR)

The following key validations are required:

- Retail Organization must be set up in the organization table (DDR_R_ORG).

- Retail Business Unit must be valid.
- Manufacturer SKU Item Number must be valid.

Process Flow Name	ITM_BSNS_UNT_ASC
Source Tables	DDR_I_ITEM_BSNS_UNT_ASSC
Target Tables	DDR_R_ITEM_BSNS_UNT_ASSC
Dependent Tables	DDR_R_ORG
	DDR_R_ORG_BSNS_UNIT
	DDR_R_MFG_SKU_ITEM

Step 10: Loading Replenishment Rules

The Replenishment Rules process flow (SKU_BSNS_INV_RULE) loads data which defines the customer's inventory replenishment (stocking) rules by business unit (store or DC) and SKU. Once the process flows runs successfully, the source records are written to the DDR_R_RTL_SKU_BU_INV_RL table. Before you can run the replenishment rules process flow, you will need to load the customer organization (DDR_R_ORG) and business units (DDR_R_ORG_BSNS_UNIT), as well as the Retailer SKU Item Master (DDR_R_RTL_SKU_ITEM).

The following key validations are required:

- Organization / Business Unit Code combination must be valid
- Retailer SKU item must be valid

Process Flow Name	SKU_BSNS_INV_RULE
Source Tables	DDR_I_RTL_SKU_BU_INV_RL
Target Tables	DDR_R_RTL_SKU_BU_INV_RL
Dependent Tables	DDR_R_ORG
	DDR_R_ORG_BSNS_UNIT
	DDR_R_MFG_SKU_ITEM

Importing Fact Data

The Demand Signal Repository (DSR) data model is based on the Oracle retail data model, but it is designed to meet manufacturer's requirements. The user can view data from either a manufacturer perspective or retailer perspective. For example, sales can be viewed using the manufacturer's item hierarchy, item numbers, and descriptions or from the retailer's item hierarchy, using the retailer's item classification method and the retailer's item numbers and descriptions. Slowly changing data, like organizations and items, is called reference data. Data that contains quantities, amounts, or other types of facts is called fact data.

Fact data is stored by store/SKU/day. This low-level granularity gives the user the maximum flexibility to organize information by time or organizational dimensions. For cases in which a retailer submits data at an aggregate level, for example, by week or by a group of stores, DSR allocates data based on a predefined organization allocation or time-allocation pattern.

The DSR data model retains historical changes to dimensional (hierarchical) data, such as retailer organization hierarchy or item classification hierarchy. This feature, called Slowly Changing Dimensions (SCD) enables users to analyze data according to the hierarchy in effect at a specific time.

System Parameters

When processing fact data, the system uses the following system parameters to determine how the data should be processed.

Run Type for Fact Data

The Run Type for Fact Data parameter uses values similar to the Run Type for Reference Data parameter. However, in addition to Simulation and Partial modes, the Run Type for Fact Data parameter also supports an All or None mode. If Run Type for Fact Data is A (All or Nothing) and one or more errors is encountered with the set of records being loaded, then the system writes no records to the target table.

Discovery Mode

If the Discovery Mode system parameter is set to Yes, then the system automatically writes a record to the Business Unit / Item Association table (DDR_R_ITEM_BSNS_UNT_ASSC). If the Discovery Mode system parameter is set to No, then the system validates that the business unit is authorized to sell this item. If the business unit and item combination is not defined, the system writes a record to the error table.

Perform Duplicate Check

The Perform Duplicate Check system parameter determines what to do if duplicate data is received in a single run. For example, if two records exist in the Sales and Returns

interface table for the same Business Unit/Item/Day, then:

- If the Perform Duplicate Check system parameter is set to No, the last record processed overwrites any previous records. No errors are reported.
- If the Perform Duplicate Check system parameter is set to Yes, the system reports any duplicates as errors and writes records to the error table.

Common Validations

The following common validations are used when processing fact data.

Unit of Measure

- Data that reports quantity information must have an associated transaction unit of measure. If unit of measure is left blank, it is assumed to be Each (EA).
- DSR provides you with the option to have an alternate unit of measure that you want to see quantities expressed.

For example, the same product is often sold in different container sizes. Soda may be sold by the carton, which contains twelve 12-ounce can, as well as individual two-liter bottles. In addition to knowing the units sold, you may also wish to know total volume (in liters) that was sold.

- When an alternate unit of measure is used, a conversion factor must be set up between the transaction unit of measure and the alternate unit of measure; otherwise, the system will treat this as an error and load the corresponding error table.

Currency

- Every manufacturer and retailer organization has an associated currency.
- Fact data that reports monetary information is assumed to be in the retailer's currency and must be converted to the manufacturer's currency.
- An exchange rate must be set up between the retailer's currency and the manufacturer's currency for the transaction date, even if the currency codes are the same; otherwise, the system treats this as an error and loads the corresponding error table.

For example, you would need to load conversion factor of 1 to go from the Retailer's currency code of USD to the Manufacturer's currency code of USD.

- If you are loading sales forecast, you should load the future date currency exchange rates for your forecast transaction dates.

Item Number

- Retailer SKU Number (RTL_SKU_ITEM_NBR) is a required field for all fact data process flows.
- Global Item ID is not required if a one-to-one relationship exists between retailer SKU number and Global Item ID.
- If a one-to-many relationship exists between Retailer SKU Number and Global Item ID, then the Global Item ID is required.
- The Global Item ID must be valid in the Manufacturer's Item table (DDR_R_MFG_SKU_ITEM table).
- Business Unit and Item must be active for the given transaction date.
- Global Item ID/Type must be valid (appear in the DDR_R_MFG_SKU_ITEM table). If receiving competitor item sales, ensure that these items are set up in the Manufacturer's item table.
- The item must be in effect (active) for the transaction date being processed.

Business Unit / Item

- Business Unit must be active (that is, effective) for the given transaction date.
- Retailer SKU Item number must be valid (it must exist in the DDR_R_MFG_SKU_ITEM table).

Location

- Inventory Location must be valid (it must exist in the DDR_R_INV_LOC table).
- If the retailer provides no inventory location, then the business unit code should be populated as the Inventory Location code to make the load possible. A null Inventory Location code is reported as an error.

Fact Data Process Flows

The following fact data process flows are used in the Demand Signal Repository implementation process.

Sales & Returns Process Flow

The Retail Sales & Returns process flow (SLS_RTRN_INF_STG_TO_TGT) loads units sold, units returned, and their related monetary value and cost. Data can be reported at the business unit level (for example, store) or at a higher level in the organization for a given day or week. Sales reported at an aggregate level are allocated to the lowest level

(by business unit per day).

Process Flow Name	SLS_RTRN_INF_STG_TO_TGT
Source Tables	DDR_I_RTL_SL_RTN_ITEM
Target Tables	DDR_I_RTL_SL_RTN_ITEM

Shipments Process Flow

The Shipments process flow (SHIPMENT_INF_STG_TO_TGT) loads shipped units and monetary amounts for a given retail business unit per item per day.

The following fields are required:

- Record ID, Business Unit, Retail SKU Item Number, Unit of Measure, Transaction Date
- Global ID and Global ID Type if Retail SKU Item Number is associated to more than one Global ID/Global ID Type

The common validations described earlier are performed for shipment data. No additional shipment specific validations are performed.

Process Flow Name	SHIPMENT_INF_STG_TO_TGT
Source Tables	DDR_I_RTL_SHIP_ITEM
Target Tables	DDR_B_RTL_SHIP_ITEM_DAY

Orders Process Flow

The Orders process flow (ORDERS_INF_STG_TO_TGT) loads order quantities and monetary amounts for a given business unit per item per day. Orders (like Shipments) are reported against a specific store or distribution center. No allocation logic is supported for order data.

The following fields are required:

- Record ID, Business Unit, Retail SKU Item Number, Unit of Measure, Transaction Date
- Global ID and Global ID Type if Retail SKU Item Number is associated to more than one Global ID/Global ID Type

The common validations described previously are performed for shipment data. No

additional shipment-specific validations are performed.

Process Flow Name	ORDERS_INF_STG_TO_TGT
Source Tables	DDR_I_RTL_ORDR_ITEM
Target Tables	DDR_B_RTL_ORDR_ITEM_DAY

Item Inventory Process Flow

The Item Inventory process flow (INVNTORY_INF_STG_TO_TGT) loads on-hand, received, in-transit, back-ordered, and quality hold quantities, net costs, and retail amounts for a given business unit/item/location/day. Item Inventory, such as Shipments and Orders, is reported against a specific store or distribution center; therefore, no allocation logic is supported for order data.

The following fields are required:

- Record ID, Retail Business Unit, Retail SKU Item Number, Inventory Location, Transaction Date
- Global ID and Global ID Type if Retail SKU Item Number is associated to more than one Global ID / Global ID Type1

The following key validations are required:

- Inventory location must be valid for the business unit.
- The common validations described previously are performed for Item Inventory.

Process Flow Name	INVNTORY_INF_STG_TO_TGT
Source Tables	DDR_I_RTL_INV_ITEM
Target Tables	DDR_B_RTL_INV_ITEM_DAY

Promotion Plans Process Flow

The Promotions Plans process flow (PROMOPLN_INF_STG_TO_TGT) loads information about the promotions that are in effect at a store for an item during a given time period, as well as whether or not the promotion includes an associated price discount. No allocation logic is supported for promotions plans.

The following fields are required:

- Record ID, Retail Business Unit, Retail SKU Item Number, Promotion Type, Promotion From Date, Promotion To Date, Promotion Price Amount
- Global ID and Global ID Type if Retail SKU Item Number is associated to more than one Global ID/Global ID Type

The common validations described previously are performed for shipment data. No additional shipment-specific validations are performed.

Process Flow Name	PROMOPLN_INF_STG_TO_TGT
Source Tables	DDR_I_PRMTN_PLN
Target Tables	DDR_B_PRMTN_PLN

Sales Forecast Process Flow

The Sales Forecast process flow (FORECAST_INF_STG_TO_TGT) loads forecasted sales units and their related monetary value. Data can be reported at the business unit level (for example, store) or at a higher level in the organization for a given day or week. Sales reported at an aggregate level are allocated to the lowest level (business unit/day).

The following fields are required:

- Record ID, Forecast Number, Forecast Type, Location Identifier, Location Identifier Flag, Retail SKU Item Number, Unit of Measure, Transaction Date
- Global ID and Global ID Type if Retail SKU Item Number is associated to more than one Global ID/Global ID Type

The following key validations are required:

- Authorized Business Unit check
- Location ID Flag: B (business unit) or O (organization)
- Period Type: D (day) or W (week)

If the Discovery Mode system parameter is set to No, the system validates that the business unit is authorized to sell (and, therefore, report sales forecast) for this item. If the business unit and item combination is not defined, the system writes a record to the error table. If the Discovery Mode system parameter is set to Yes, the system automatically writes a record to the Business Unit / Item Association table.

The Location ID flag indicates whether the retailer is reporting data at the business unit level (that is, the lowest level in its organization hierarchy) or at a higher level within the organization. Data reported at a higher level than business unit will be allocated down to the business unit level based on the pattern established in the Organization

Allocation table. · If the Location ID flag is not populated, the system assumes it is a B (business unit) and performs no allocation.

The Period Type flag indicates whether the retailer is reporting data weekly or daily. Data reported weekly will be allocated down to day, based on the pattern established in the Time Allocation table. If the Period Type flag is not populated, the system assumes it is a D (day) and performs no allocation.

Process Flow Name	FORECAST_INF_STG_TO_TGT
Source Tables	DDR_I_SLS_FRCST_ITEM
Target Tables	DDR_B_SLS_FRCST_ITEM_DAY

Goals and Targets Process Flow

The Goals and Targets process flow (GOALS_THD) loads goals, thresholds, and external measures. Goals, thresholds, and external measures can be loaded for any level of the retail organization hierarchy, manufacturer item hierarchy, or retailer item hierarchy. These measures can also be loaded for any time period (year, month, quarter, week, or day).

The following fields are required:

- Record ID
- Manufacturer's Org
- Retailer Org Code and Level
- Mfg Item Hierarchy Code and Level
- Time Hierarchy Code and Level
- Retailer Item Hierarchy Code and Level
- Measure Name, Type and Value

If any of these fields is not populated, the system writes it to the error table.

Note: When setting goals or thresholds, or defining external measures based on item hierarchy, you must choose either the manufacturer's or the retailer's item hierarchy, but not both.

The following key validations are required:

- Measure Type Name:

EXTERNAL, GOAL, THRESHOLD

- Measure Name:
GROSS MARGIN, % RETURNS, CATEGORY MONETARY SALES, DEDUCTION AMT, DEDUCTION BALANCE, IN STOCK %, INVENTORY COVER MAX, INVENTORY COVER MIN, INVENTORY TURNS, INVOICE ACCURACY, ITEM DATA ACCURACY, ITEM DATA SYNCHRONIZATION, LATE PAYMENT, MONETARY SALES, NO OF DEDUCTIONS, ON TIME DELIVERY, ON-HAND INVENTORY, ORDER CHANGE %, ORDER CYCLE TIME, PAYMENT DAYS, PERFECT ORDER %, RETAIL GROSS MARGIN (%), SALES FORECAST ACCURACY (%), SERVICE LEVEL, UNIT SALE
- Organization level:
ORGANIZATION, CHAIN, AREA, REGION, DISTRICT, BU (or NA)
- Manufacturer Item Hierarchy Level:
COMPANY, GROUP, DIVISION, CLASS, SUBCLASS, ITEM, SKU (or NA)
- Retailer Item Hierarchy Level:
COMPANY, GROUP, DIVISION, DEPARTMENT, CLASS, SUBCLASS, ITEM, SKU (or NA)
- Time Hierarchy Level:
YEAR, QUARTER, MONTH, WEEK, DAY

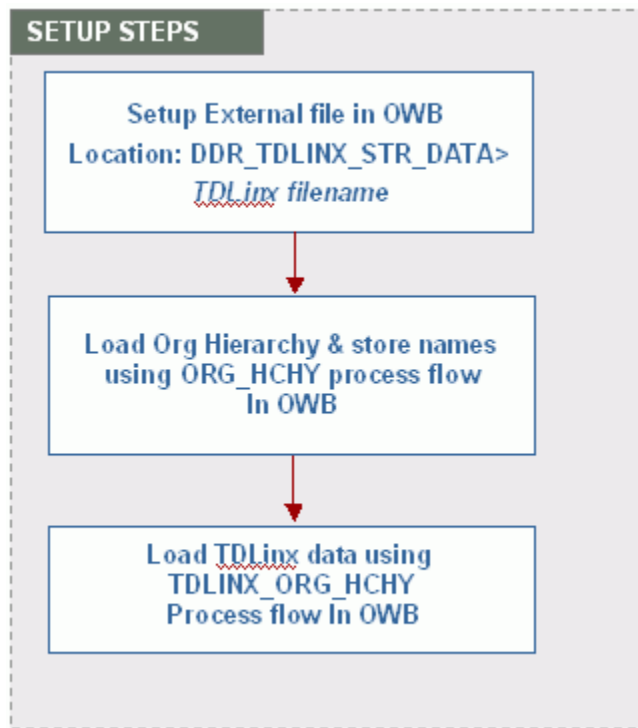
Note: When loading goals or thresholds, or when importing external measures, you must use one or more of the appropriate organization, item, or time hierarchy level codes listed. NA indicates that the organization, manufacturer item, or retailer item hierarchy is not applicable for the goal, threshold, or external measure.

Process Flow Name	GOALS_THD
Source Tables	DDR_I_GOAL_TARGET
Target Tables	DDR_B_GOAL_TARGET

Loading Syndicated Consumption Data

The following diagram illustrates the required setup steps for loading syndicated

consumption data:



Setup the TDLinx File in OWB

Before loading syndicated store data, you must provide a .csv file using the filename created under External File location (such as DDR_TDLINX_STR_DATA.csv) in strict TDLinx format, as specified in Nielsen's TD Data Dictionary. The file must be a location named DDR_TDLINX_FILE_LOCATION, and the input file is expected to be a csv file with placeholders for all values.

Load Org Hierarchy and Store Names

From Oracle Warehouse Builder, load the Org reference data into the DDR_R_Org table using the ORG_HCHY process flow. The following tables are loaded with the Org hierarchy reference data and TDLinx store names using the ORG_HCHY process flow:

- DDR_R_ORG
- DDR_R_ORG_CHAIN
- DDR_R_ORG_AREA
- DDR_R_ORG_REGION
- DDR_R_ORG_DISTRICT

- DDR_R_ORG_BSNS_UNIT

Load TDLinx Data using the TDLINX_ORG_HCHY Process Flow

From Oracle Warehouse Builder, use the OWB_TDLINX_ORG_HCHY process flow to load TDLinx store data into the DDR_R_ORG_BSNS_UNIT table.

The data can be loaded in discrete steps by first running the Map to populate the interface table and then running Map to load the reference table. The table DDR_I_ORG_BSNS_UNIT in DDR_SYND_STR_DATA_FI_MAP is populated with the data in the TDLinx file. Validations like duplicate records check, referential integrity check, NOT Null check and others will be done by map DDR_ORG_HCHY_IF_MAP.

Implementing User-Defined Fields

The process to load Item and Organization Hierarchies is:

1. Create the source files, or modify existing files, in the required format and include 20 User-Defined Attribute Fields for:
 - Manufacturer Item Hierarchy
 - Manufacturer SKU Item
 - Retailer Item Hierarchy
 - Retailer SKU Item
 - Organization Hierarchy
 - Organization Business Unit
2. Load data to appropriate interface files.
3. Run MAP or Process Flow to move data to target tables.
 - Correct any errors.
 - Rerun the MAP or Process Flow.

Oracle Warehouse Builder (OWB) contains the MAPs and Process Flows to load data from the interface tables into DSR. After loading the data into the interface tables, run the MAPs or process flow to move the data from the interface tables into the target or error tables. The new MAP DDR_MFG_RTL_ORG_HCHY_ATTR_MAP loads the Manufacturer Item, Retailer Item and Organization Attributes to their target or error tables.

The new process flow MFG_RTL_ORG_HCHY_ATTR loads the Manufacturer Item,

Retailer Item and Organization Attributes to their target or error tables. Process Flows can be used, instead of MAPs, to load the data in stages to facilitate troubleshooting.

- FILE_INF process flows move data from the source to the interface or error table.
- ERR_TO_INF process flows move corrected data from the error table to the interface table.
- MAIN_PF process flows move corrected data from the interface table to the target table.

Setup Flow – Item Clusters, Retailer Clusters and Customer Promotion Plan

The process to load Item Clusters, Retailer Clusters and Customer/Retailer Promotion Plans is:

1. Create the source files, or modify existing files, in the required format and include 20 User-Defined Attribute Fields for:
 - Item Clusters
 - Retailer Clusters
 - Customer/Retailer Promotion Plans
2. Load data to the appropriate interface files.
3. Run MAP or Process Flow to move data to Target Tables.
 - Correct any errors.
 - Rerun the MAP or Process Flow.

Note: The existing MAPs for Item Clusters, Retailer Clusters and Customer Promotions have been modified to include the attributes loaded from user-defined fields. The existing process flows for Item Clusters, Retailer Clusters and Customer Promotions have also been modified to include the attributes.

Process Flows and Maps

The Process Flows and Maps for loading Retailer Item, Manufacturer Item, Organization, Item Clusters, Retail Clusters and Customer Promotions are below. Items in bold are used for loading the Attributes from the User-Defined Fields.

Process Flow	Map	Entity Loaded
DDR_MFG_RTL_ORG_HCHY	DDR_MFG_RTL_ORG_HCHY_ATTR_MAP	Attributes for Manufacturer & Retailer Organization Hierarchies
ORG_HCHY	DDR_ORG_HCHY_IF_MAP	Organization Hierarchy
MFG_ITM_HCHY	DDR_MFG_ITEM_HCHY_IF_MAP	Manufacturer Item Hierarchy
RTL_ITM_HCHY	DDR_RTL_ITEM_HCHY_IF_MAP	Retailer Item Hierarchy
ORG_HCHY	DDR_ORG_HCHY_IF_MAP	Organization Hierarchy
ITM_CLUSTER	DDR_ITEM_CLSTR_IF_MAP	Item Clusters
RTL_CLUSTER	DDR_RTL_CLSTR_IF_MAP	Retail Clusters
MFG_PRMTN	DDR_MFG_PRMTN_IF_MAP	Manufacturer Promotion

Manufacturer Promotions

Manufacturers are responsible for loading promotion data into the interface table. Interface tables are provided along with DSR. Users are free to use these tables as templates for designing their own interface tables.

The Organizations and SKUs included in the Manufacturer Promotion detail file must be loaded into the Reference tables prior to loading the promotion detail. Duplicate records are moved to an error table with only 'Duplicate' error message. That is, no further checking is done on these records.

When associating promotions to sales data:

- The Promotion Status Field is informational only and is not used in the mapping process.
- Only promotions that the Manufacturer wants considered in the mapping process should be loaded into DSR. DSR attempts to map POS data to any relevant Manufacturer Promotion regardless of the Promotion Status field.

- When the POS data for a SKU is aggregated to provide a daily total, if one unit of an SKU is sold on promotion and 99 units are not sold on promotion, the POS data aggregated by SKU, Store and Day will be flagged as promoted for that SKU.

Manufacturer Promotion Tables and Process Flows

Manufacturer Promotion dimension data is loaded into interface table `DDR_I_MFG_PRMTN`, and the source file must contain:

- Promotion Code
- Promotion Name
- Promotion Type
- Dates must be in `yyyymmdd` format

Data is then loaded into the target table, `DDR_R_MFG_PRMTN`, by running one of the following:

- `DDR_MFG_PRMTN_IF_MAP`
- `MFG_PRMTN` Process Flow

Manufacturer Promotion detail data is loaded into interface table `DDR_I_MFG_PRMTN_DTL`. The Manufacturer Promotion detail source file must contain:

- Promotion Code
- Retail Organization Code
- Manufacturer SKU Item Number
- Promotion From (Start) Date
- Promotion End Date
- Dates should be in `yyyymmdd` format

The promotion details are then loaded into the target table, `DDR_R_MFG_PRMTN_DTL`, by running one of the following:

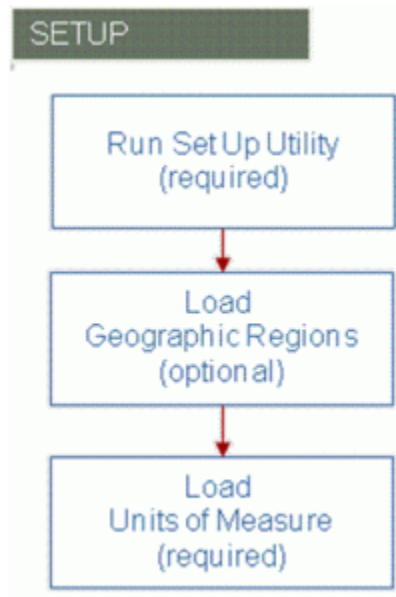
- `DDR_MFG_PRMTN_IF_MAP`
- `MFG_PRMTN` Process Flow

Run the process flow `PRM_ASSC>MFG_PRMTN_ASSC` after data successfully loads to target tables. Enter the parameter for number of days prior to the current date that sales

should be considered for association to the promotions.

Once data is loaded and the Manufacturer Promotions are associated with the Customer's sales, run the Process Flow DSR_OTHER_PROCESS_FLOWS to refresh the Materialized Views.

Third Party Distributors



In order to implement an Third Party Distributors, you need to complete the following steps:

1. Run the set up utility to load any new pre-seeded values.
2. Define any geography regions or sub-regions in which the distribution centers reside.
3. Load any new units of measures.

Step 1: Run Setup Utility

Use the DDR_SETUP_UTIL.SETUP setup utility to load pre-seeded data into DSR. This includes:

- System parameters
- Error messages
- Lookup values

The set up utility will load the organization type of "DST" into the look up master which is required for the third party distributor feature to be used.

Step 2: Load Geographic Regions

Use the GEOGRPHY_RGN_HCHY process flow to load the appropriate geographic regions. If you want to analyze distributor data by geographic region or sub-region, you'll need to make sure they are set up prior to loading the distribution organization. If you are already using geographical regions and sub-regions to analyze customer data, you may already have the necessary regions and sub-regions set up.

Step 3: Load Units of Measure

Use the ITM_UOM process flow to set up the required UOM codes. Note that if you have already loaded unit of measure codes for manufacturer and retailer items, chances are that you already have the necessary UOM codes set up. If, however, a distributor item uses a unit of measure that's not in use by either the manufacturer or retailer, then you'll need to set that unit of measure up as well.

Manufacturer Shipments

In order to implement Manufacturer Shipments, you need to complete the following steps:

1. Run the set up utility to load any new pre-seeded values.
2. Define any geography regions or sub-regions in which the distribution centers reside.
3. Load any new units of measure.

Step 1: Run Setup Utility

Use the Manufacturer Shipments utility to load pre-seeded data into DSR. This includes:

- System parameters
- Error messages
- Look up values

The set up utility will load the organization type of "MFG" into the look up master which is required for the Manufacturer Organization to be loaded. The DSR set up utility is a PL/SQL Procedure (DDR_SETUP_UTIL.SETUP).

Step 2: Load Geographic Regions

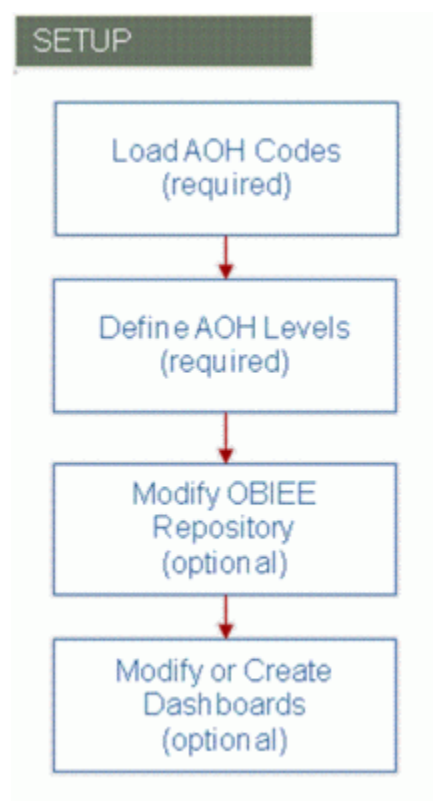
Distribution Centers may be associated to a geographic sub-region which aggregates to a geographic region, and are used to analyze business units (stores or distribution center) by geography. Use the process flow GEOGRPY_RGN_HCHY to load Geographic regions.

Note: If you want to analyze manufacturer data by geographic region or sub-region, you'll need to make sure they are set up prior to loading the manufacturer's organization. If you are already using geographical regions and sub-regions to analyze customer data, you may already have the necessary regions and sub-regions set up.

Step 3: Load Units of Measure

The last step is to load your units of measure using the process flow ITM_UOM.

Alternate Organization Hierarchy



In order to implement an Alternate Organization Hierarchy, you need to complete the

following steps:

1. Determine how many different Alternate Organization Hierarchies you need and load an AOH code for each one used. It's likely that you will only have one or two sources of Alternate Organization Hierarchy Data.
2. Define how many levels each Alternate Organization Hierarchy has and the name of each level.
3. Modify the OBIEE repository to expose each Alternate Organization Hierarchy and their appropriate levels.
4. Modify existing dashboards to use an Alternate Organization Hierarchy or create new dashboards that use the Alternate Organization Hierarchy.

Step 1: Load Alternate Organization Hierarchy Code

The first step is to load an Alternate Organization Hierarchy Code. An AOH Code uniquely identifies each Alternate Organization Hierarchy in use. For example, the value of 'SDP' can be assigned to an alternate organization hierarchy received from a syndicated data provider, or the value of 'CBO' can be assigned to an AOH that represents the Customer's Buying Organization. When you execute the AOH Attributes process flow, the system reads the source table and populates the target table listed here.

There is no prerequisite data which needs to be loaded prior to loading the Alternate Organization Hierarchy Code.

Process Flow Name:	AOH_ATTR
Source Table(s):	DDR_I_AOH_ATTR
Target Table(s):	DDR_R_ORG_HCHY
Dependent Table(s):	None
Required Fields:	AOH Code AOH Name
Key Validations:	There are no additional validations beyond checking for the required fields and reporting any duplicate records. Duplicate record checking is standard within DSR. A duplicate record error will occur when the same AOH Code appears more than once in the interface table.

Step 2: Define Alternate Organization Hierarchy Levels

This is a planning step, and the outcome determines what changes (if any) are needed to the DSR repository. For each source of alternate organization hierarchy information, you need to determine the maximum number of levels to be exposed. The sample Alternate Organization Hierarchy contains four levels plus the business unit. The existing Alternate Organization Hierarchy structure supports up to 10 levels, plus business unit. If you have an Alternate Organization Hierarchy over 10 levels (not including business unit), then additional data model changes will be required.

Next you will need to assign names to each level in the Alternate Organization Hierarchy. In a ragged hierarchy, level names may be generic. For example, in the sample Alternate Organization Hierarchy level names are simply called, level 1, 2, 3, and so on. In a more fixed taxonomy, you may want to assign more specific names to each level, like "National Account", "Regional Account", or "Planning Account".

Then you'll want to determine what account attributes should be exposed at each level. In this example, we have exposed the same set of attributes at each level, but you may choose to expose different attributes at each level. In addition to the attributes shown here, there are a set of 20 user-defined attributes (10 character and 10 numeric), which can be exposed and named according to their use.

Step 3: Modify OBIEE Repository

The third step is to modify the OBIEE repository based on the requirements defined in step two. DSR exposed four levels of an Alternate Organization Hierarchy. If additional levels are needed, or if attributes need to be renamed, someone with OBIEE administration experience will perform the following changes:

1. Create a materialized view (MV) containing level five alternate organization hierarchy information. Existing alternate organization hierarchy materialized views can be used as a model.
2. Add the newly created MV to the connection pool in the physical layer under the Alternate Organization Hierarchy dimension.
3. Add a new child object between Level four and the Business Unit in the business model and mapping layer. The business unit should always be the last level in the Alternate Organization dimension.
4. Drag and drop the needed columns over from the physical layer to the Business Model and Mapping layer. For example, Level five Account ID and Level five Name.
5. Drag and drop the desired columns into the presentation layer and rename accordingly.

6. Once you have completed changes to the repository they must be deployed to the BI server.
7. Finally, the existing Reference Data Complete Refresh process flow need to be modified to include the refresh of the new MV created in the first step.

For more information on updating , see Configuring the Alternative Organization Hierarchy in OBIEE

Step 4: Modify or Create Dashboards

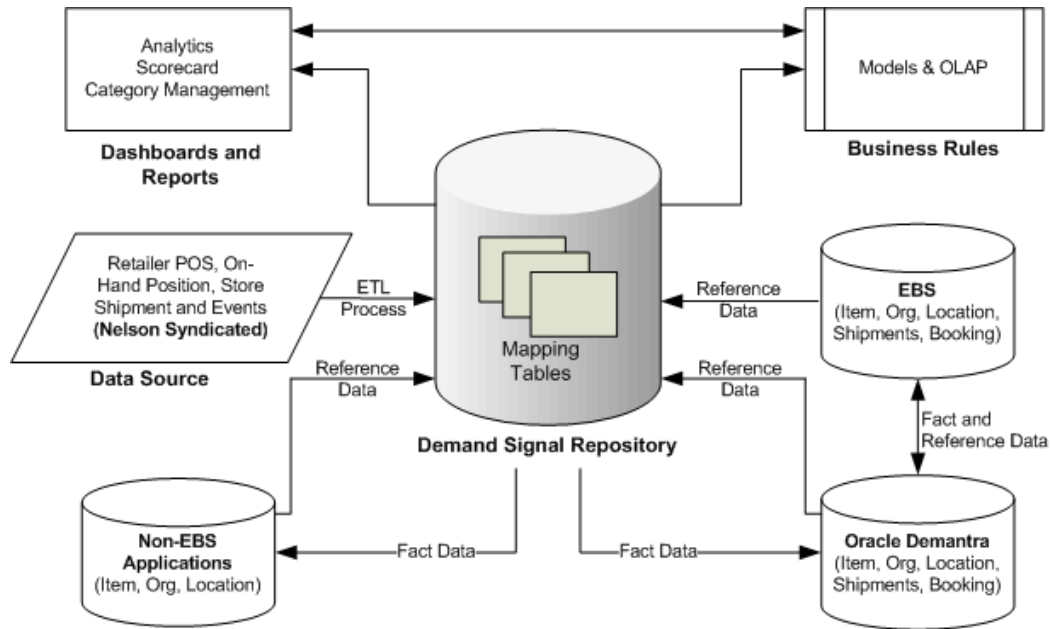
The final set up step is to modify existing dashboards or create new dashboards which use the Alternate Organization Hierarchy. Any of the existing out-of-the-box DSR queries which use the customer organization hierarchy can be modified to use a desired level within the Alternate Organization Hierarchy.

Demand Signal Repository Integration

Business Process Task Flow

The Oracle Demand Signal Repository application uses the following business process task flow:

1. An Administrator loads cleansed data from various sources, including Retailer Point of Sale, Retailer Inventory, and syndicated into the Demand Signal Repository.
2. Mapping tables are updated with newly loaded data and synchronized with existing data.
3. Oracle Warehouse Builder performs initial mapping of Organization- and Geography-level members for inclusion and synchronization with EBS items and locations for EBS implementations.
4. The Oracle Data Integrator is used for mapping Organization, Geography, and Product members for non-EBS implementations.
5. The Demand Signal Repository exports data required for analytical applications at the required levels. For Demantra, item, organization, and site (Shipto) at the day or week level are required using Demantra's internal identifiers for EBS implementations.
6. Analytical applications upload the data from Demand Signal Repository for analysis within the application.



Web Services

DSR Web Services enable sharing DSR data with complementary applications such as:

- Demantra Demand Management
- Manufacturer applications
- Retailer applications

Demand Signal Repository shares demand data with manufacturer applications at both the lowest level of detail and the aggregate level. A web service for each group of facts in the data model enables an application to select and retrieve data according to the dimensions of time, manufacturer item hierarchy, retailer organizational hierarchy (selected levels), and location. These web services provide cleansed retail point of sale, on-hand inventory and other data at daily store level, or aggregated to a higher level as needed. The following table describes each of the supported Web Services:

Web Service	Description
getSalesReturnItem	Returns detailed data from the Retailer Sales Return Item Day fact table.
getMrktItemSales	Returns detailed data from the Market Item Sales Fact table.

getPrmtinPlan	Returns detailed data from the Promotion Plan fact table.
getRtlInvItmDat	Returns detailed data from the Retail Inventory Item Day fact table.
getRetailerOrderItem	Returns detailed data from the Retailer Order Item Day fact table.
getSaleFreicastItem	Returns the detailed data from the Sales Forecast Item fact table.
getRetailerShipItem	Returns the detailed data from the Retailer Ship Item fact table.
getAgrSalesReturnItem	Returns aggregate data from the Retailer Sales Return Item Day fact table.
getAgrMrktItemSales	Returns aggregate data from the Market Item Sales fact table.
getAgrPrmtinPlan	Returns aggregate data from the Promotion Plan fact table.
getAgrRtlInvItmDat	Returns aggregate data from the Retail Inventory Item Day fact table.
getAgrRetailerOrderItem	Returns aggregate data from the Retailer Order Item Day fact table.
getAgrSaleFreicastItem	Returns aggregate data from the Retailer Order Item Day fact table.
getAgrSaleFreicastItem	Returns aggregate data from the Sales Forecast Item fact table.
getAgrRetailerShipItem	Returns aggregate data from the Retailer Shipment Item fact table
getFileName	Returns the name of the files for the specified job.
getDDRXmlFile	Returns the streamed xml file data for the specified file name.

The Demand Signal Repository provides a web services interface for external applications to retrieve data at detailed and aggregate levels. These services correspond to the major fact tables in the system. They support the most common repetitive data

access scenarios:

DSR Data Types	Detailed Data Access	Aggregate Data Access
Sales (POS) Returns	getSalesReturnItem	getAgrSalesReturnItem
Sales (actual, average, normal, price cut, ad) ACV	getMrktItemSalesWeek	getAgrMrktItemSalesWeek
Retail Promotions	getPrmtinPlan	getAgrPrmtinPlan
On-Hand Inventory, In-Transit Inventory, Backorders, Quality Hold Merchandise, Receipts	getRtlInvItmDat	getAgrRtlInvItmDat
Store Orders	getRetailerOrderItmDay	getRetailerAgrOrderItmDay
Forecasts	getSaleFreicastItmByDay	getAgrSaleFreicastItmByDay
Shipments	getRetailerShipItemDay	getAgrRetailerShipItemDay

Web Services returns aggregated (lowest levels are added into any higher levels in dimension hierarchies) and detailed (lowest dimension hierarchy level maintained in Fact table) fact data. Output data is in XML format stored in an XML file. For detailed level Web Services, output data is generated based on the specified hierarchy level input given by the calling application. Generated Output data is at lowest level of the dimension hierarchy stored in Fact tables. For the aggregate level Web Services, aggregated output will be generated based on the web service call for the specified hierarchy level and expected aggregation hierarchy level.

External application requests data by specifying the specific dimensional level and its values from the supported hierarchies:

Hierarchy	Supported Dimensions
Time	Day
	Gregorian Month
	Gregorian Year

Time	Day
	Business Week
	Business Month
	Business Quarter
	Business Year
Item	SKU
	Item
	Sub Class
	Class
	Group
	Division
Organization	BU
	Retail Chain
	Retail Org
	Retail Cluster
Location	Address
	Location
	City
	State
	Country

Loading TDLinX Store/Outlet Data

The TDLinX Store/Outlet adapter enables the import of syndicated Consumer Goods data from Nielsen. Nielsen syndicated store data is available in a format called TDLinX. Manufacturers use this data to analyze Retailer store level and market level data.

The adapter loads the supplied .csv file, which must follow Nielsen's TDLinX formats strictly: column order, length, position as specified in the TDLinX Data Dictionary. The input file must be a csv file with placeholders for all values. The adapter performs the necessary translations prior to populating the associated DSR interface table. The Import Data procedure maps fields using the column names defined within the TDLinX source file to the matching columns in the interface table.

Loading TDLinx Data using the TDLinx Store/Outlet Adapter:

1. Copy the DDR_TDLINX_STR_DATA.csv file (data in strict TDLinx format) to the location specified by the DDR_TDLINX_FILE_LOCATION parameter.
2. Load Org data into the DDR_R_ORG table using the TDLINX_ORG_HCHY process flow.
3. Load the Org Hierarchy data tables (DDR_R_ORG_CHAIN, DDR_R_ORG_AREA, DDR_R_ORG_REGION, DDR_R_ORG_DISTRICT and DDR_R_BSNS_UNIT) using the TDLINX_ORG_HCHY process flow.
4. Load Store data (DDR_R_BSNS_UNIT), from TDLinx data using the TDLINX_ORG_HCHY process flow.

To move the data from the TDLinx file in separate steps:

1. Use the DDR_TDLINX_STR_FI_MAP process flow to move the data to the DDR_I_ORG_BSNS_UNIT Interface table.
2. Use the DDR_ORG_HCHY_IF_MAP process flow to move the data from Interface table to the DDR_R_ORG_BSNS_UNIT target table.
3. Use the ORG_HCHCY_ERR_TO_INF process flow to move data from DDR_E_ORG_BSNS_UNIT error table to the Interface table.

Enterprise Resource Planning to Demand Signal Repository Integration

Demand Signal Repository integrates with Enterprise Resource Planning (ERP) applications, so that these applications can share data. This integration facilitates:

- Data transfer from a variety of sources. For example, point-of-sale and syndicated data is synchronized and available for feeding EBS or Manufacturer point applications.
- Internal identifiers of EBS reference data, for example, items, customers, and so on, are synchronized with Demand Signal Repository data and used during export.
- Internal identifiers of Manufacturer reference data, for example, items, customers, and so on, are synchronized with Demand Signal Repository data and used during export.

Inputs to Demantra from EBS and Demand Signal Repository

EBS Data

Location Hierarchy	Loc11	Loc22	Loc33	
Item Hierarchy	ItemX	Cat11	Cat22	Cat44

DSR Data

Location Hierarchy	Loc11	Loc2	Loc3	Loc4
Item Hierarchy	ItemY	Cat1	Cat2	

DSR Retailer DC Shipment data

	Item	Location	Week1	Week2
DSR Retailer DC Shipment	ItemY	Loc1	10	20

Mapping

- EBS ItemX = DSR ItemY
- EBS Loc11 = DSR Loc1

Outputs from Demantra to EBS Applications

Demantra export integration profile for ASCP:

	Item	Location	Week1	Week2
DSR Retailer DC Shipment	ItemX	Loc11	10	20

Demand Signal Repository Manufacturer Data Synchronization

Use the Demand Signal Repository integrations to synchronize the Manufacturer reference data to the corresponding data from point-of-sale, syndicated sources, and retailer inventory.

Inputs to manufacturer's point application from Demand Signal Repository and manufacturer systems:

Manufacturer Reference Data

Location Hierarchy	Loc11	Loc22	Loc33	
Item Hierarchy	ItemX	Cat11	Cat22	Cat33

DSR Data

Location Hierarchy	Loc1	Loc2	Loc3	Loc4
Item Hierarchy	ItemY	Cat1	Cat2	

DSR Retailer DC Shipment Data

	Item	Location	Week1	Week2
DSR Retailer DC Shipment	ItemY	Loc1	10	20

Mapping

- Manufacturer ItemX = DSR ItemY
- Manufacturer Loc11 = DSR Loc1

Demand Signal Repository to Demantra Demand Management Integration

By integrating Demantra Demand Management with Demand Signal Repository, users are able to:

- Load data from Demand Signal Repository for further analysis using the statistical capabilities of Demantra Demand Management.
- Forecast point-of-sale data received at the lowest levels of Demand Management.
- Start the Demantra forecast engine with store-level data, aggregated to site and weekly point-of-sale levels.

This integration extracts specified data from Demand Signal Repository for downloading into Demantra staging tables at the lowest level of the Demantra hierarchies. By default, data is exported at a weekly level but can be modified by the Administrator to export at other aggregation levels.

Data that is exported from Demand Signal Repository includes:

- Item and location reference data
- Retailer POS Sales data at day or week, store aggregated to Site (Shipto), and Item levels using the Demantra identifying codes
- Retailer POS Forecast at day or week, store aggregated to Site (Shipto), and Item levels using the Demantra identifying codes
- Retailer DC/WHSE Forecast at day or week, Site (Shipto), and Item levels using the Demantra identifying codes
- Retailer Store OnHand Total data at day or week, store aggregated to Site (Shipto), and Item levels using the Demantra identifying codes
- Retailer Store Orders Total at day or week, store aggregated to Site (Shipto), and Item levels using the Demantra identifying codes
- Retailer DC/Whse OnHand data at day or week, Site (Shipto), and Item levels using the Demantra identifying codes
- Retailer DC/Whse Shipments (also known as Store Shipments) at day or week, Site (Shipto), and Item using the Demantra identifying codes
- Retailer DC/Whse Intransit at day or week, Site (Shipto), and Item using the Demantra identifying codes

Demand Signal Repository Data Extract

The Oracle Warehouse Builder map DDR_DMNT_INTG_MAP populates the Demand Management interface staging table BIIO_DSR_SALES_DATA using information from the mapping tables. The process flow pulls 2 years of data and can be:

- Scheduled to run automatically
- Executed by the DSR Administrator
- Run as part of a custom PL/SQL job

Up to two years of data relative to the current System date (one year's data before the system date, and one year's data after the system date) is collected from DSR and loaded. This data includes:

- Retail Sale Return - Only Point of Sale (POS) data.
- Sale Forecast – Two year span of all data.

- Inventory – Two year span of all data for the last day of the week only.
- Retailer Ship – Two year span of DC data
- Retailer Order – Two year span of Point of Sale data

Lowest-Level Item Data

Demantra requires the following fields for each record in the item data:

- A unique code to be used as the identifier for any lowest-level item. This code is provided by mapping in Demand Signal Repository.
- A unique description, which is a short string that serves as a user-friendly name of the item. If no description is available, use the code.
- Additional codes that indicate the membership of this item within all levels of the item hierarchy
- Additional fields that describe this item, as needed.
- Additional fields that specify unit conversion factors for this item, if needed.

Lowest-Level Location Data

Demantra requires the following fields for each record in the location data:

- A unique code to be used as the identifier for any lowest-level location.
- A unique description, which is a short string that serves as a user-friendly name of the location. If no description is available, use the code.
- Additional codes that indicate the membership of this location within all levels of the location hierarchy.
- Additional fields that describe this location, as needed.

Sales Data

Demantra requires the following fields for each record in the sales data:

- The unique code of the item being sold.
- The unique code of the location of the sale.
- The date of the sale.

- The number of units that were sold for this item, location, and date. This field must be numeric.
- Price per unit for this item, at this location and date. This field must be numeric.
- Additional fields, as needed.

Aggregation in Time

You must select the smallest time unit that you will use within Demantra. Correspondingly, you must specify the start of that time unit (such as the starting day of the week) and an aggregation method for Demantra to use when importing data (backwards or forwards). However, it is not necessary to pre-aggregate the raw data in time. The Demantra loading and integration tools can perform that aggregation, if needed. That is, if you import multiple sales records for different dates for the same item-location combination, then Demantra automatically sums them into the time unit to which those dates belong.

Note: Together, the item, location, and date will form the primary key for the sales record. That is, Demantra stores no more than one record for each combination of item, location, and date.

The preseeded hierarchies in Demantra are:

- Item Levels
 - Product Category: Item > Category
 - Product Family: Item > Product Family
 - Demand Class: Demand Class
- Location Levels
 - Zone: Site > Trading Partner Zone > Zone
 - Customer Class: Site > Account > Customer > Customer Class
 - Business Group: Organization > Oper Unit > Bus Grp
 - Legal Entity: Organization > Legal Entity
 - Sales Channel: Sales Channel

Time Considerations

Use the following time considerations when implementing the Demand Signal

Repository to Demantra Demand Management Integration:

- If base time unit = day, then:
 - Gregorian Calendar: Day > Month > Quarter > Year
 - Quarter one starts January 1st
 - Default 4-4-5 Calendar: Day > Week (Default 4-4-5) > Period (Default 4-4-5) > Quarter (Default 4-4-5) > Year (Default 4-4-5)
 - Week 1 starts on the first day of the calendar year that matches the Demantra 'week starts on' setting. For example, if 'week starts on' = Sunday, then Week 1 of 2007 would start on Sunday January 7, 2007.
 - Periods are four, four, then five weeks long, with the pattern starting on Week 1 of the year.
 - Quarters are three periods long, starting on Week one.
 - Years last from Day one of Week one to the day before the first day of the next calendar year that is a 'week starts on' day. For example, Year 2007, assuming 'week starts on' = Sunday, Year 2007 will last from January 7, 2007 through January 5, 2008.
- If base time unit = week, then:
 - Default 4-4-5 Calendar: Week (Default 4-4-5) > Period (Default 4-4-5) > Quarter (Default 4-4-5) > Year (Default 4-4-5)
- If base time unit = month, then:
 - Gregorian Calendar: Month > Quarter > Year > All

Loading DSR Data as Demantra Series

The following DSR data can be loaded as series in Demantra:

- Retailer POS Sales: Retailer POS data aggregated to Site
- Retailer POS Forecast: Retailer POS Forecast aggregated to Site
- Retailer DC/WHSE Forecast: Retailer Distribution Center/ Warehouse Forecast
- Retailer Store OnHand Total by Site: Retailer OnHand Store inventory aggregated to Site
- Retailer Store Orders Total by Site: Retailer Store Orders aggregated to Site

- Retailer DC/Whse OnHand: Retailer OnHand inventory at DC/WHSE
- Retailer DC/Whse Withdrawals by Site: Retailer Withdrawals (Store Shipments) from DC/Whse
- Retailer DC/Whse Intransit from DC/WHSE: Retailer Intransit Shipments from DC/WHSE

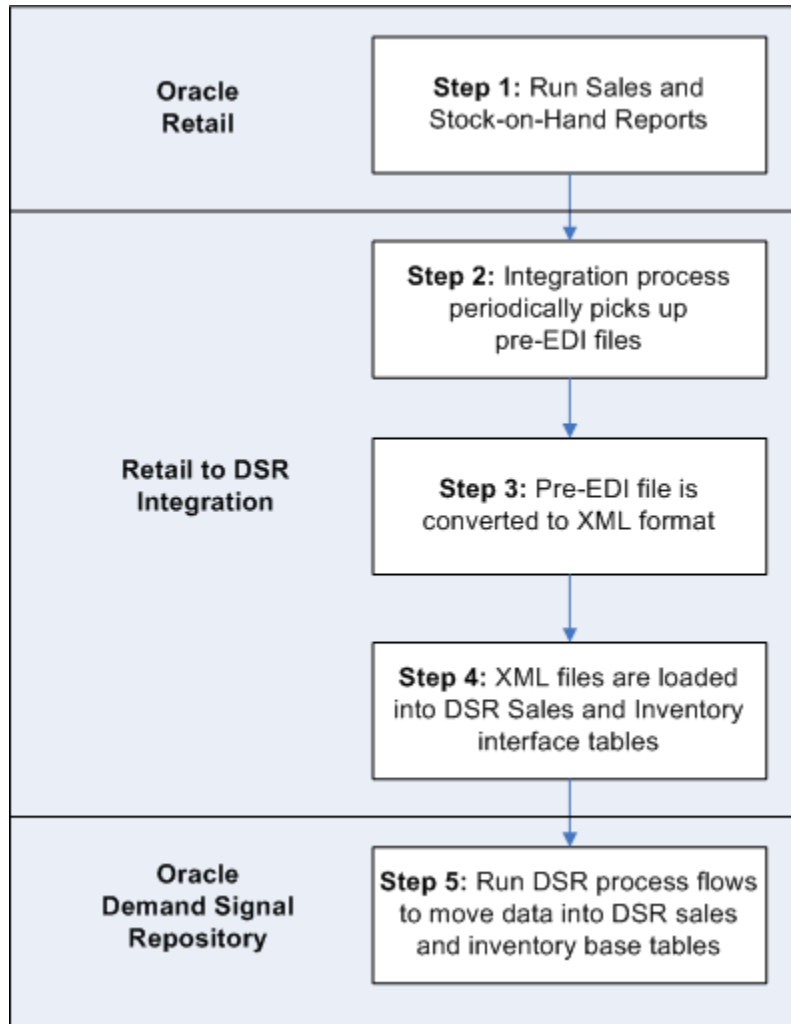
Demand Signal Repository to Retail Merchandising System Integration

The Retail Merchandising System (RMS) Integration provides an out-of-the-box integration of EDI 852 data coming from Oracle's RMS system into Oracle's Demand Signal Repository (DSR) module. This integration allows Consumer Goods Manufacturers to import data from their retail customers using Oracle's Retail Merchandising System (RMS) to Demand Signal Repository.

Oracle Retail produces a report in a flat file pre-EDI format that contains information for each supplier. To secure the data for each supplier Oracle Retail should produce one file per supplier. This report contains item quantities grouped by Stores and Warehouse Locations.

Integration Flow

The following diagram illustrates the RMS to DSR integration flow:



Integration Prerequisites

In order for the RMS integration to function you must have the following prerequisite applications installed and configured:

- Oracle Demand Signal Repository release 12.1
- Oracle Retail Merchandising System version 10.0 (or higher)
- Oracle Data Integrator version 10.1.3.4

RMS Integration Setup and Process

To use the RMS integration, you must first install and configure Demand Signal Repository as well Oracle Data Integrator (ODI). Once these have been installed and configured, you must RMS Integration system parameters in DSR are set appropriately.

Once the set up steps are complete, you will need to load the RMS EDI 852 file into the proper directory and run the DSR integration scenario. Upon successful completion of the scenario, the DSR sales and return interface table and the DSR inventory interface table will be populated. The standard Oracle Warehouse Builder (OWB) process flows are used to process the data in the interface tables and populate the target DSR tables.

RMS Configuration Steps

The productivity data report is a sales audit summary that is sent to specify EDI vendors, giving sales details, current stock on hand for all location, and current in transit quantities for each of the items primarily supplied by that vendor.

This program polls all suppliers that require activity reporting. Those that require daily reports have daily sales information selected from the EDI_DAILY_SALES table, while those requiring weekly reports pull information from the ITEM_LOC_HIST and ITEM_MASTER tables to get the weekly sales data. The store level table ITEM_LOC is queried for current stock levels, and the quantities that are in-transit are derived from the transfer tables.

Use the following steps to configure the RMS to DSR integration:

1. Configure the JAVA_HOME variable.
2. Configure the DDR_R_LKUP_MST DSR lookup table.
3. Create the directory structures as specified in the RETL_DSR_SRC_DIR, RETL_DSR_SUCCESS_DIR, RETL_DSR_FAILED and RETL_DSR_CLASS_DIR parameters.
4. In the ODI Topology Manager, configure the ODI physical architecture for the DSR and XML schemas.
5. Copy the file RETLDLPRD.dtd to the location specified in the RETL_DSR_SRC_DIR parameter.
6. Copy the file DLPRDFileToXMLConverter.class to the location specified in the RETL_DSR_CLASS_DIR parameter.
7. Move the EDI file to the location specified in the RETL_DSR_SRC_DIR parameter.
8. Execute the Integration scenario in ODI.
9. In the Oracle Warehouse Builder, run the Sales and Return Fact Data and Inventory Fact Data process flows.

Configuring the DDR_R_LKUP_MST Lookup Table

The following DSR system parameters are used by the RMS integration. These values can be changed using the Look Up Master ETL process flow.

The Retail Source Directory (RETL_DSR_SRC_DIR) parameter specifies the directory where the RMS EDI (EDIDLPRD) file is stored for upload to DSR. When running the RMS integration scenario, the scenario will look for a file in this directory. If a file is found, the scenario continues; otherwise, the scenario waits indefinitely for an EDI file to show up in this directory.

Parameter Name	Description	Example
RETL_DSR_SRC_DIR	The directory where the RMS EDI File is stored for upload to DSR.	ODI_HOME/retldsr/source
RETL_DSR_CLASS_DIR	The directory where the DLPRDFileToXMLConverter.class File should be kept.	RETL_DSR_HOME\RetailDSRHome
RETL_DSR_SRC_FILENAME	The RMS EDI file name that is loaded into DSR.	EDIDLPRD.edi
RETL_DSR_SUCCESS_DIR	The directory where the RMS EDIDLPRD file is moved after a successful upload.	ODI_HOME/retldsr/success
RETL_DSR_FAILED_DIR	The directory where the RMS EDIDLPRD file is moved if the upload fails.	ODI_HOME/retldsr/failed
RETL_DSR_TO_ADD_R	The admin email address where notifications are sent.	firstname.lastname@oracle.com
RETL_DSR_FM_ADD_R	The Integrator admin email from which notifications are sent.	retl-dsr@integrator.com
RETL_DSR_MAIL_SERVER	Mail Server that sends the ODI email notifications.	mail.oracle.com

Creating Directory Structures

Ensure that the directory structures as specified in RETL_DSR_SRC_DIR, RETL_DSR_SUCCESS_DIR and RETL_DSR_FAILED_DIR exist.

Setting the Location for the RETLDLPRD.dtd Schema in ODI

You must set the location of the RETLDLPRD.dtd schema file in ODI. From the Topology Manager, navigate to the Physical Architecture tab and enter the JDBC location. For example:

```
jdbc:snps:xml?d=RETL_DSR_HOME\RETLDLPRD.dtd &S=RETLDLPRD
```

Configuring the ODI Physical Architecture for the DSR and XML Schemas

You must configure the EBS DSR target database schema so that the integration can connect to connect the DSR installation to ODI. The schema name is typically "DDR". For example:

```
jdbc:oracle:thin:@<server_name>:<port>:<db_name>
```

Running the Integration Scenario

Once the system parameters have been set and an EDI file has been loaded into the directory specified in the system parameters, start Oracle Data Integrator (ODI) and run the Load Retail Items to DSR scenario.

After running the integration scenario, the History tab displays all the steps within the scenario that executed and whether or not they were successful. In the case of an error, the EDI file is moved to the failed directory and an e-mail notification is sent that the scenario failed.

Attribute Mappings

The integration transforms the Oracle Retail Sales and Stock On Hand pre-EDI file to XML format and then maps it to Oracle-DSR Sales and Inventory Facts interface tables. The mapping between the XML elements to table columns is shown below:

Sales Item Mappings

Oracle Retail Source EDIDLPRD File	Oracle Retail Source [XML Element]	Oracle DSR Target DDR_I_RTL_INV_ITEM Columns	Mapping Details
		REC_ID	DDR_I_RTL_INV_ITEM_SEQ.NEXTVAL
TQUTY.Location	<Location>	LOC_IDNT_CD	RTRIM (ITEMQUANTITY.LOCATION)
TITEM.Item	<ItemNum>	GLBL_ITEM_ID	RTRIM (ITEM.ITEMNUM)

TITEM.Ref_Item	<RefItemNum>	RTL_SKU_ITEM_NBR	CASEWHEN (ITEM.REFITEM IS NOT NULL, ITEM.REFITEM, ITEM.VENDORCATA LOGNUMBER)
TITEM.Vendor catalog number	<VendorCatalogNum ber>		
TITEM.Item_Num_ Type	<ItemNumType>	GLBL_ITEM_ID_TYP	RTRIM (ITEM.ITEMNUMTYP E)
TQUTY.Quantity where QuantityDescriptor ='Sold'	<SoldQty>	SLS_QTY	RTRIM(QTYBYLOCA TION.SOLDQUANTI TY)
		PERIOD_TYP_FLAG	Default 'D'
		LOC_IDNT_FLAG	Default 'B'
		ORG_LVL_CD	Default 'O'
THEAD.Report date	<Supplier ReportDate=" YYYYMMDD">	SRC_SYS_DT	TO_DATE (DLPRD.CREATEDA TE,'YYYYMMDD')
FHEAD.File Source	<DLPRD Source=" DLPRD">	SRC_SYS_IDNT	CONCAT('RETL:', DLPRD.SOURCE)
FHEAD.File Create Date	<DLPRD CreateDate=" YYYYMMDD">	TRANS_DT	TO_DATE (SUPPLIER.REPORTD ATE,'YYYYMMDD')

Inventory Item Mappings

Oracle Retail Source EDIDLPRD File	Oracle Retail Source [XML Element]	Oracle DSR Target DDR_I_RTL_INV _ITEM Columns	Mapping Details
		REC_ID	DDR_I_RTL_INV_ITEM_SEQ. NEXTVAL

TQUTY.Location	<Location>	RTL_BSNS_UNIT_CD	RTRIM (ITEMQUANTITY.LOCATION)
TITEM.Item	<ItemNum>	GLBL_ITEM_ID	RTRIM (ITEM.ITEMNUM)
TITEM.Ref_Item TITEM.Vendor catalog number	<RefItemNumber > <VendorCatalog Number>	RTL_SKU_ITEM_NBR	CASEWHEN (ITEM.REFITEM IS NOT NULL, ITEM.REFITEM, ITEM.VENDORCATALOGN UMBER)
TITEM.Item_Num_ Type	<ItemNumType>	GLBL_ITEM_ID_TYP	RTRIM (ITEM.ITEMNUMTYPE)
TQUTY.Location	<Location>	INV_LOC_CD	RTRIM (ITEMQUANTITY.LOCATION)
TQUTY.Quantity where QuantityDescriptor ='On hand'	<OnhandQty>	ON_HAND_QTY	RTRIM(QTYBYLOCATION.ON HANDQUANTITY)
TQUTY.Quantity where QuantityDescriptor ='In Transit'	<InTransitQty>	IN_TRANSIT_QTY	RTRIM(QTYBYLOCATION.IN TRANSITQUANTITY)
TQUTY.UnitCost where QuantityDescriptor ='On hand'	<OnhandQtyUnit Cost>	ON_HAND_NET_COST_AMT	RTRIM(QTYBYLOCATION.ON HANDQTYUNITCOST) * RTRIM(QTYBYLOCATION.ON HANDQUANTITY)
TQUTY.UnitCost where QuantityDescriptor ='In Transit'	<InTransitQtyUnit Cost>	IN_TRANSIT_NET_COST_AMT	RTRIM(QTYBYLOCATION.IN TRANSITQTYUNITCOST) * RTRIM(QTYBYLOCATION.IN TRANSITQUANTITY)
THEAD.Report date	<Supplier ReportDate="YYYYMMDD">	TRANS_DT	TO_DATE (SUPPLIER.REPORTDATE,'YYYYMMDD')
FHEAD.File Source	<DLPRD Source="DLPRD">	SRC_SYS_IDNT	CONCAT('RETL:', DLPRD.SOURCE)

FHEAD.File Create Date	<DLPRD CreateDate=" YYYYMMDD">	SRC_SYS_DT	TO_DATE (DLPRD.CREATEDATE,'YYY YMMDD')
---------------------------	--------------------------------------	------------	----------------------------------------------

Index

A

ACV measures, 1-12, 6-22
allocating data, 1-20
alternate organization hierarchy, 7-3
attribute mappings, 9-17

B

business calendar, 1-19
Business Intelligence Dashboard, 1-2
business units, 2-1

C

calendars, 1-18
 configuring, 8-2
clusters, 1-15
codes
 configuring, 8-2
currencies, 1-20
Customer Organization Hierarchy, 2-2

D

dashboards, 1-2, 4-1
 Manufacturer Promotions, 4-2
 Sales Performance by Account, 4-5
data, 1-3, 7-1
 allocating and aggregating, 1-20
data loading interface tables, 7-18
data model, 2-1
Demand Signal Depository
 data, 1-3

Demand Signal Repository
 Introduction, 1-1
 measures, 1-4

E

enabling third party distributors, 7-35
ERP integration, 9-6
exception criteria, 5-1
exception measures, 1-11
exceptions, 5-1
exceptions dashboard, 5-5

F

fact data, 2-5
 importing, 8-34
fact data process flows, 2-6
Fact Data Process Flows, 8-36
fiscal calendar, 1-19
forecast types, 7-26

G

goals, 1-12, 6-27
Gregorian calendar, 1-18

H

hierarchies, 1-14
 loading, 8-15, 8-24
hierarchies, loading, 7-2

I

implementation

- loading organizations and hierarchies, 8-15
 - populate pre-seed data, 8-2
- implementation overview, 8-1
- importing fact data, 8-34
- integration, 9-1, 9-6
 - RMS, 9-13
 - web services, 9-2
- interface tables, 7-18
- Introduction
 - Demand Signal Repository, 1-1
- item clusters, 1-15
- items
 - loading, 8-24

L

- loading
 - items and hierarchies, 8-24
 - replenishment rules, 7-22
- loading ACV measures, 6-22
- loading hierarchies, 7-2
- loading organizations and hierarchies, 8-15
- lower bound measures, 4-6

M

- manufacturer promotions, 7-29
- Manufacturer Promotions dashboard, 4-2
- manufacturer shipments, 7-42, 8-48
- materialized views, 7-24
- measures, 1-4, 6-1
 - ACV, 1-12
 - exception measures, 1-11
 - lower bound, 4-6
- multiple hierarchies, 1-14

N

- Neilson TDLinx Adapter, 9-5

O

- Operations Scorecard, 4-9
- organization allocations, 7-16
- organization and business units, 2-1
- organizations
 - loading, 8-15
- overview
 - calendars, 1-18

- dashboards and scorecards, 4-1
 - data, 7-1
 - implementation, 8-1
 - measures, 6-1
 - reports and scorecards, 1-13
- Overview, 1-1

P

- planning and advertising calendars, 1-19
- populating pre-seed data, 8-2
- process flows
 - fact data, 2-6
- purpose codes, 7-26

R

- Reference Data Load Process , 2-2
- replenishment rules, 7-22
- reports, 1-13, 4-9
- retail clusters, 1-16
- RMS integration, 9-13
- roles, 3-1

S

- Sales Performance by Account dashboard, 4-5
- Sales scorecard, 4-8
- scorecards, 1-13, 4-1, 4-5
 - sales, 4-8
- Scorecards
 - operations, 4-9
 - Supply Chain, 4-8
- security, 3-1
- setting goals, 6-27
- setting thresholds, 6-28
- Setting Up Codes and Calendars, 8-2
- store clusters, 1-15
- Supply Chain scorecard, 4-8
- syndicated consumption data, 8-41
- syndicated consumption measures, 6-22

T

- TDLinx Adapter, 9-5
- third party distributors, 8-47
- thresholds, 1-12, 6-28
- time allocations, 7-16

U

user defined fields, 7-24
user-defined fields, 8-43
user roles, 3-1
users, 3-1

W

web services, 9-2

