THE POWER TO KNOW.

SAS® Inventory Optimization 5.1

Data Administration Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2011. *SAS® Inventory Optimization 5.1: Data Administration Guide*. Cary, NC: SAS Institute Inc.

SAS® Inventory Optimization 5.1: Data Administration Guide

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

Electronic book 2, November 2013

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

Contents

Recommended Reading

- *SAS Data Integration Studio: User's Guide*

For a complete list of SAS books, go to support.sas.com/bookstore. If you have questions about which titles you need, please contact a SAS Book Sales Representative:

SAS Books
SAS Campus Drive
Cary, NC 27513-2414
Phone: 1-800-727-3228
Fax: 1-919-677-8166
E-mail: sasbook@sas.com
Web address: support.sas.com/bookstore

Chapter 1

Introduction to SAS Inventory Optimization

Welcome to SAS Inventory Optimization

What is SAS Inventory Optimization?

SAS Inventory Optimization is an analytic and optimization solution that takes product demand and generates optimal inventory replenishment policies. You can plan inventory and orders to fulfill the forecasted demand, and maintain optimized stock levels over a single, dual, and multiple echelon distribution network, improve customer satisfaction, and reduce total costs.

SAS Inventory Optimization is a Java client application. The dynamic user interface of the solution caters to the needs of different users within an organization. You can view, review, interpret, and analyze results of the inventory optimization process. You can also perform what-if scenario analysis, review suggested order quantities, and then submit them to the enterprise resource planning (ERP) systems.

Functional Components of SAS Inventory Optimization

SAS Inventory Optimization includes the following main functional components:

data warehousing
: consists of the ongoing ETL jobs that update the data warehouse with changes from the source system.

analytics
consists of the ETL jobs for the inventory optimization processes.

Java applications
consists of the SAS Inventory Optimization middle tier and client tier.

These components are seamlessly integrated to offer a true end-to-end solution. The SAS Inventory Optimization solution takes data from various source systems and applies appropriate analytical methods to provide information for advanced business decisions.

How Does SAS Inventory Optimization Help You?

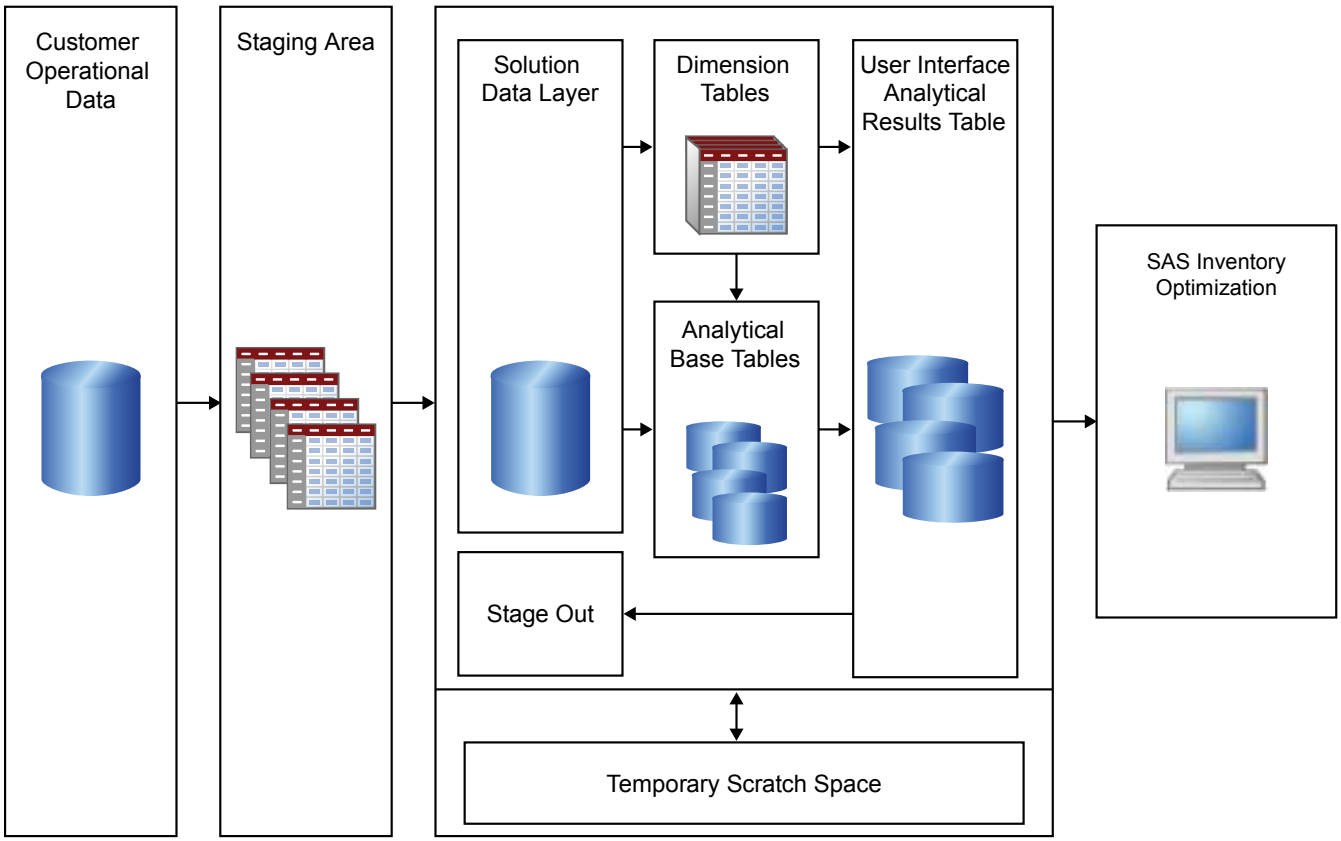
Major uses of SAS Inventory Optimization include the following:

- consume organizational demand forecasts and align supply chain costs to set optimal inventory placement for best service levels at the lowest possible costs
- provide input to plan replenishment activities that include inventory rebalancing

Back-End Data Flow for Loading and Accessing Data

The following figure shows the general flow for the back-end data.

Figure 1.1 *Back-End Data Flow*



The general data flow works as follows:

1. Source data is stored at the data source layer in the customer operational data system.
2. Tables in the staging area are populated from the source data.
3. The solution data layer (SDL) is loaded with the data from the staging area. The recommended method to load data into the SDL is as follows:
   a. Identify data structures in the SDL that need to be loaded to meet your business needs.
   b. Identify and locate source systems for the data by mapping the SDL data and the source system data.
   c. Extract, cleanse, and transform the source system data and consolidate the data into the staging area that is designed to load data into the SDL. The data in the staging area is validated to ensure that all interdependencies or relationships in the SDL are defined.
   d. Load lookup (reference) tables that store code values that are required in the SDL.
   e. Load data into the SDL in accordance with a specific sequence. For more information about the sequence, see "Loading of Lookup Tables" on page 8.

   Note: All the earlier steps are site-specific. After the SDL is loaded, SAS Inventory Optimization provides ETL jobs that use the SDL data for further processing. Initial jobs are run to load the configuration tables and to set up the environment for running the next set of ETL jobs, which are the ongoing jobs. One of the configuration table stores the base period — the time between two replenishment orders that can be placed. The base period is customizable and you can specify whether the base period must be week, month, or quarter.
4. The ongoing jobs are run to load or update the SAS Inventory Optimization tables in accordance with the base period. The jobs perform the following main tasks:
   a. Load dimension tables with the latest attributes from the SDL.
   b. Run back-end processes for inventory optimization to load the analytical base tables.
   c. Load the application tables (also called as user interface analytical result tables - UIARTs).
   d. Load the tables in the Stageout library to store the results of the inventory optimization process. These results can be updated in the source system.

   Note: The temporary library Scratch is used as an intermittent storage area.
5. After the SAS Inventory Optimization tables are loaded, the latest data is displayed on the interface of the SAS Inventory Optimization application.

   For more information about the data management tasks, see Chapter 2, "Data Administration and Management of SAS Inventory Optimization," on page 5.

Related SAS Software

Overview

Many features that are not found in SAS Inventory Optimization are available in other SAS solutions or in SAS products that are used with this SAS solution. If you do not find a feature that you need in this software, you might find it in one of the following SAS solutions or products.

SAS Inventory Replenishment Planning

SAS Inventory Replenishment Planning enables you to manage your inventory levels while fulfilling the customer service levels by providing optimal inventory policies and recommending when and how much to order.

SAS Inventory Replenishment Planning enables you to create what-if scenarios by using different parameters or forecasts, and to determine the impact on policies and projected customer service levels and costs.

SAS Enterprise Data Integration Server

SAS Enterprise Data Integration Server is an application that enables you to manage ETL process flows. These tasks are sequences of steps for the extraction, transformation, and loading of data. SAS Enterprise Data Integration Server enables you to do the following:

- specify metadata for sources, such as tables in an operational system
- specify metadata for targets, such as tables and other data stores in a data warehouse
- create jobs that specify how data is extracted, transformed, and loaded from a source to a target

Where to Go for More Information

Online Help

For information about how to operate your software, select **Help** ⇨ **Help Contents** from within the application.

For information about the version of the software that you are running, select **Help** ⇨ **About SAS Inventory Optimization** from within the application.

SAS Technical Support Services

As with all SAS products, the SAS Technical Support staff is available to respond to problems and answer technical questions about SAS Inventory Optimization.

Chapter 2

Data Administration and Management of SAS Inventory Optimization

Overview

After you complete the post-installation tasks, the SAS Inventory Optimization solution is ready for use. You can now perform the following data-related tasks:

1. Select the base period.
2. Perform the pre-initial data load tasks.
3. Perform the ongoing data tasks.

For more information about the general data flow, see "Back-End Data Flow for Loading and Accessing Data" on page 2.

Selecting the Base Period

The base period for the SAS Inventory Optimization solution must match the planning frequency of your organization. You can specify the base period as day, week, month, or quarter.

Pre-Initial Data Load Tasks

Overview

Before you load initial data in the SAS Inventory Optimization solution, perform the following tasks:

1. Confirm that libraries are created.
2. Run the initial jobs.
3. Load all tables in the solution data layer (SDL) library.
4. Customize the global parameter values.
5. (Optional) Update the metadata for the inventory optimization tables.
6. Set or modify the job parameter values.

Confirm That Libraries Are Created

After you complete the installation of the SAS Inventory Optimization solution, ensure that all the libraries are created. The default path for the libraries is as follows:`<SASCONFIG>/Lev<N>/AppData/SASInventoryOptimization/data`. To view the libraries through the SAS Management Console, select **Data Library Manager** ⇨ **Libraries**. Here is a list of the libraries that are created:

ABT
: stores tables that are specific to analytical solutions.

Control
: stores tables that contain control parameters. These parameters are used to run the solution jobs. These tables are created and loaded by the initial one-time jobs. The control parameters can be customized.

DIM
: stores dimension tables that are loaded with the latest attributes from the SDL tables.

Scenario
: stores tables that contain information about the scenarios that are created.

Scratch
: stores intermediate tables that are used in various processes.

SDL
: stores tables that contain the source data. Also, contains tables to store the forecasted data that is used by the inventory optimization process. You must load these tables with customer-specific data.

Stageout
: stores solution data that must be updated in the source system (for example, optimization policies, approved orders, and promoted settings).

TSDB
: stores tables that are loaded or updated by the user. The library is used for concurrent usage of tables from the user interface and back end. The tables in this library are not physically present but are appended to the iodb database that is created during installation.

UIART
: stores application tables that are loaded through ETL processes. These tables contain information that are required to be displayed in the user interface of the application.

Load All Solution Data Layer Tables

Set Up the Solution Data Layer

The solution data layer (SDL) tables contain source data from the source system in a format that is uniform and complete. The SDL tables ensure that the source data format facilitates their accurate import into the tables of SAS Inventory Optimization.

Create empty SDL tables by running the **`sdl_ddl.sas`** script file that is provided as a part of the installable. For the Windows operating environment, the script is available at **`!SASROOT\invoptsrv\sasmisc`**. If you install SAS in the default folder, then the script is available at **`C:\Program Files\SAS\SASFoundation\9.3\invoptsrv\sasmisc`**.

For the UNIX operating environment, the script is available at **`!SASROOT/misc/invoptsrv`**.

The tables Forecasted_Demand and Intermittent_Demand contain the forecasted data that will be used by the inventory optimization process. This data is not directly available in the source system and needs to be loaded in the tables. Ensure that the base period that you choose to load the data in these tables is the same as that of the inventory optimization process.

Loading Sequence of Solution Data Layer

The loading of the solution data layer (SDL) must follow a sequence to ensure that all the table dependencies are taken care of. The following table provides the loading sequence of SDL tables. The sequence consists of groups. You can load the tables that belong to a group simultaneously or in any order. However, you must load all the tables of a group before loading tables from the next group, that is, the load must follow a groupwise ascending sequence.

Table 2.1 *Loading Sequence of Solution Data Layer Tables*

Group Number	Table Name
1	• Item_Category • Location • Lookup_Master • Network_Model • Route_Type_Ref
2	• Employee • Item • Lookup_Detail • Organization

Group Number	Table Name
3	• BOM • Customer • Facility • Item_Price • Item_Substitute • Item_Succession • Vendor
4	• Customer_Order • Facility_X_Item • Purchase_Order • Period_type • Route • Vendor_Facility_Item
5	• Back_Order_Summary • Dispatch • Facility_Item_Demand • Forecasted_demand • Intermittent_demand • Facility_Item_Inventory • Facility_Item_X_Network • Network_X_Route • Pipeline_Inventory • Receipts • Route_X_Item

Loading of Lookup Tables

The solution data layer (SDL) library has two reference or lookup tables that store code values and their descriptions. These value and description pairs are required by the attribute tables or are used to display information in the user interface. The ETL code derives the description of the code columns from the lookup detail table. Hence, you must ensure that the required code values are present in the lookup detail tables.

The following two reference tables exist for lookup:

- Lookup master table (Lookup_Master):

  This table contains the SDL table name and the _CD column name that is present in the SDL table.

  Table 2.2 *A Sample Record in the Lookup Master Table for Facility Type Code*

TABLE_NAME	COLUMN_NAME
Facility	Facility_Type_CD

- Lookup detail table (Lookup_Detail):

  This table stores details of the code columns, that is, the code values and their descriptions.

Table 2.3 *Sample Records in the Lookup Detail Table for Facility Type Code*

TABLE_NAME	COLUMN_NAME	CODE_VALUE	CODE_DESC
Facility	Facility_Type_CD	100	IT COMPANY
Facility	Facility_Type_CD	110	SERVICE BASED
Facility	Facility_Type_CD	120	PRODUCT BASED
Facility	Facility_Type_CD	130	CALL CENTER

The following table provides the table name and its mandatory code columns that must be loaded.

Table 2.4 *Table and Its Mandatory Code Column Names*

Table Name	Code Column Name
Facility	Facility_Type_CD
Facility_X_Item	• Policy_Type_CD • Service_Type_CD
Item	• Item_Group_CD • Item_Status_CD • Item_Type_CD • Make_Or_Buy_CD
Item_Price	Price_Type_CD

Run the Initial Jobs

The initial jobs are to be run only once, after you install the solution. The initial jobs load the initial tables that are required before loading data for the solution.

Note: After you run the first job, see "Customize the Global Parameter Values" on page 10, before proceeding with the other jobs.

Run the jobs in the following sequence:

1. 01_CREATE_GLOBAL_PARAMETER_LIST
2. 02_CREATE_CONTROL_TABLES
3. 03_CREATE_TIME_PERIOD_DATA
4. 04_CALENDAR_HIERARCHY1

5. 05_CALENDAR_HIERARCHY2
6. 06_TIME_DIM
7. 07_CREATE_TIME_VIEWS
8. 08_CREATE_USER_TABLES

For more information about the job location and job details, see Appendix 2, "ETL Job Details," on page 31.

After you load the initial jobs, the following main tasks are accomplished:

- All the global parameters are loaded.
- The control tables and user tables are created. Control tables store the controlling or key information about SAS Inventory Optimization The Control.Job_Status table contains the status for some of the ETL jobs. You can review the statuses of the jobs in this table to verify that the jobs ran successfully.

  User tables are the intermediate tables that are created either in the transactional database or in the UIART library. These tables are used to store information that is displayed in the user interface and received through the interface.
- The time dimension table is created that occupies an important place in every data warehouse. In SAS Inventory Optimization, the time dimension table is populated only once before running the ongoing ETL jobs. The table supports the hierarchy calendar Day → Week → Month → Quarter → Year. You can specify the start date and the number of years for which the table is to be generated. To view the default values of the parameters that are used to populate the time dimension table, see "Job Parameter Table" on page 28. With the time dimension table, views for week, month, quarter, and year are also created, which are required for running the ETL jobs.

All these tasks ensure that the configuration of the overall environment for SAS Inventory Optimization is complete.

Customize the Global Parameter Values

Some of the ETL processes populate the analytical base tables (ABTs) and user interface analytical result tables (UIARTs). These processes use certain global parameter values for correctly extracting data from the solution data layer (SDL). All the global parameters are listed in the Control.Global_Parameter_List table.

For example, to extract the latest data from the SDL, the following condition is used in the ETL process:

```
WHERE VALID_TO_DTTM = &GLOBAL_HIGH_DTTM_VALUE
```

Here, the macro variable Global_High_DTTM_Value holds the parameter value:

```
"01JAN5999:00:00:00"DT
```

All global parameters are created after you run the initial jobs. The parameters are set to their default values. You can customize these parameters as per your requirement.

To view a list of the global parameter settings, see "Global Parameter Table" on page 21.

Update the Metadata for the Inventory Optimization Tables

Perform this task only if you choose a base period other than week and if you want to manage data by using SAS Data Integration Studio.

The default base period for the inventory optimization process is set to week. The tables that are used by the inventory optimization process use WK in the name and physical name of their metadata to signify that the base period is week. For example, the table IO_WK_Arc_Data_ABT contains WK.

If you modify the base period to day, month, or quarter, then using SAS Data Integration Studio, you must modify WK in the name and physical name of the tables to DAY, MTH, or QTR respectively. For example, if you specify month as the base period, then the table name IO_WK_Arc_Data_ABT must be changed to IO_MTH_Arc_Data_ABT.

Set or Modify the Job Parameter Values

Job parameters are the control parameters for ETL jobs. You must set up these parameters for running the jobs. All parameters have a specific value or a default value that you can change as per your requirement.

To set or modify the value of a job parameter:

1. In SAS Data Integration Studio, right-click the job, and from the pop-up menu, select **Properties**. The Properties dialog box appears.
2. Click the **Parameters** tab. If parameters are defined for the job, then a list of all the parameters is displayed.
3. Select the parameter to be modified and click **Edit**. The Edit Prompt dialog box appears.
4. Click the **Prompt Type and Values** tab and in the **Default value** box, type the new value.
5. Click **OK** twice to close both the dialog boxes. The parameter value is changed.

In the **Parameters** tab, you can edit the existing parameters, if required. To view the jobs for which parameters need to be specified, see "Job Parameter Table" on page 28.

Ongoing Data Tasks

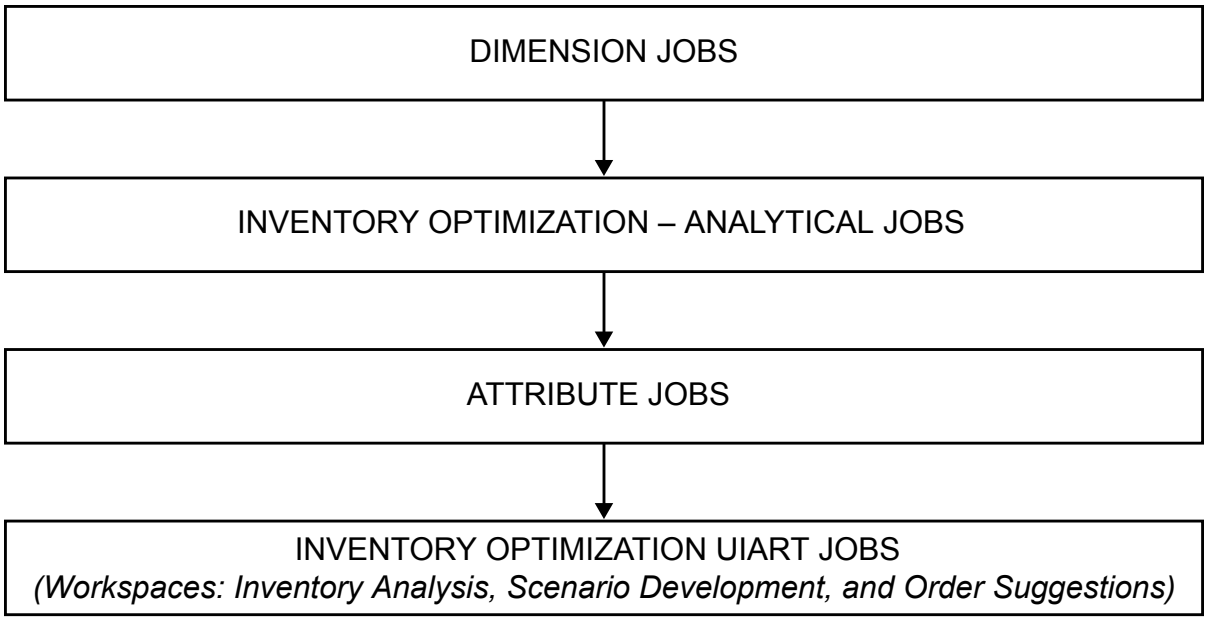
Overview

After you perform the pre-initial data load tasks, you can run the solution-specific jobs. These jobs need to be run periodically based on the base period. The jobs can run on a daily, weekly, monthly, or quarterly basis. Some jobs that run on a daily basis ensure that any settings that are updated through the user interface are reflected on the interface on the next day.

During the nightly, weekly, monthly, or quarterly offline periods, you must run or schedule the ETL jobs to load initial or incremental data. Also, load the back-end jobs to run the solution-specific tasks.

The following figure shows the generic flow of the SAS Inventory Optimization jobs.

Figure 2.1 *Basic Job Sequence*



The subsequent sections provide an overview of each of the jobs. For more information about each job, see Appendix 2, "ETL Job Details," on page 31.

Overview of Dimension Jobs

Attributes of the facility and item dimension tables might undergo change if there is an update to an existing facility or item, or if a new facility or item is added. When the dimension jobs run for the first time, the jobs load all the required attributes from the SDL. For ongoing loads, the jobs handle loading of incremental data by using extract transformation.

Note: Extract transform is a user-written transform that is provided with the ETL package of the solution. The transform extracts only incremental records from the source table.

The facility dimension table describes details of the distribution center warehouse with its five-level organization hierarchy. The facility attributes consist of facility ID, name, type code, and open and closure date time. The organization attributes consist of organization ID, name, and internal or external indicator.

The item dimension table describes the item with its ten-level item category hierarchy. The item attributes consist of item ID, name, description, group code, type code, status code, pack type code, sales introduction or discontinued data, brand name, and make or buy code. The table also stores item indicators, such as whether the item is a finished good, an assembly item, an item bundle, and so on.

Inventory Optimization: Analytical Jobs

Overview of Inventory Optimization

The inventory optimization back-end process in SAS Inventory Optimization provides answers to the following questions:

- When should you place orders to restock inventory?

- What should be the appropriate inventory level?
- What is the projected customer service level?
- How should you replenish inventory to reduce costs and increase turns?

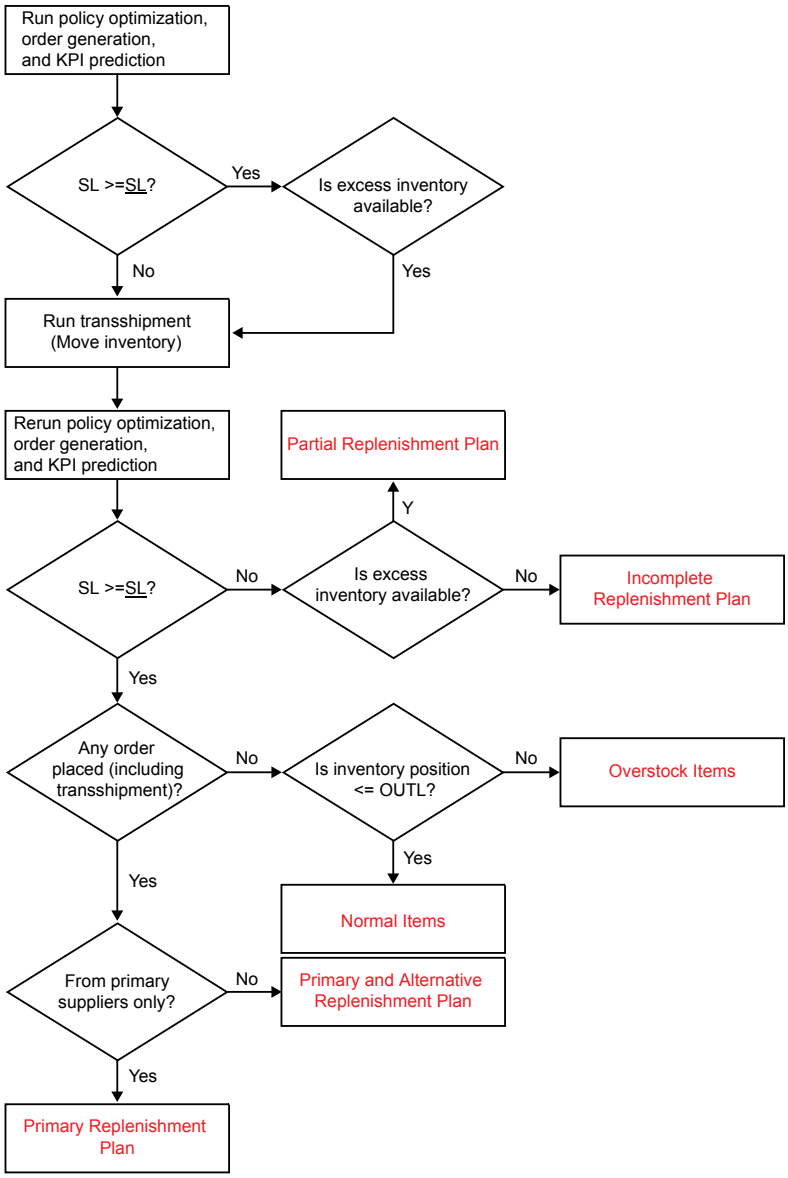
The inventory optimization process handles the following conditions:

- item succession, wherein the inventory data is populated such that the inventory of the predecessor can be merged with that of the successor items
- repaired parts, wherein the inventory data of the repaired parts can be added to the current inventory data
- back orders, wherein demand data can be adjusted for the back orders
- kit or bundled items, wherein demand can be specified for both, a pack or a bundle of items, and for individual items that constitute the bundle
- multiple vendors, wherein lead time calculation for the vendor-facing node is based on the business share of each vendor.

The process provides optimized inventory replenishment policies and order quantities. Policy optimization is performed on the basis of specific constraints such as target service levels and user-defined inputs such as lead time and inventory cost; thereby, enabling users to maintain optimized stock levels over a multi-echelon service network, improve customer-satisfaction, and reduce total costs.

The following figure explains order generation process for SAS Inventory Optimization.

Figure 2.2 *The Order Generation Process*



Legend

SL	Average Projected Service Level
SL	Lower Bound of Targeted Service Level
OUTL	Order-up-to Level

Inventory Optimization Batch Process Jobs

The inventory optimization batch process jobs are categorized as follows:

- jobs to create input tables for storing node, arc, demand, and inventory details that are required for the inventory optimization process
- jobs to run the inventory optimization process
- jobs to populate tables that are required for displaying information about the user interface of the solution

The inventory optimization batch process uses the following information from the SDL:

- forecasted values of the customer demand
- node (facility and item pair) information such as lead time, service level requirement, and unit holding cost in a network
- network structure information or arcs between predecessors and successors
- the amount of inventory of an item to arrive at a facility for a period in the planning horizon

To run the inventory optimization batch process:

1. Specify the base period parameter in the Control.Global_Parameter_List table.
2. Run the jobs to create input tables for the inventory optimization process.
3. Run the inventory optimization process jobs. Any warning or error messages that occur when the process runs are stored in a table. The workflow of the inventory optimization process jobs is as follows:
    a. For all facility and item pairs, the process runs the MIRP procedure for policy optimization, order generation, and key performance indicator (KPI) prediction. All the facility and item pairs are grouped into two categories:
        - Facility and item pairs with the average projected service level value higher than or equal to the lower bound of the target service level. These facility and item pairs are further split into two subcategories:
            - facility and item pairs with excess inventories
            - facility and item pairs without excess inventories
        - Facility and item pairs with average projected service level value less than the lower bound of the target service level.

    b. You can choose to run the transshipment module for the facility and item pairs with excess inventories. The transshipment module is also run for facility and item pairs with average projected service level value less than the lower bound of the target service level value.

        The transshipment module recommends the inventory to be transported from the facility and item pairs with excess inventory to the facility and item pairs in shortage, within each inventory pool. After the move, the inventory pipeline with excess facility and item pairs is updated with results that are suggested by transshipment.

    c. The inventory optimization process runs the MIRP procedure again for policy optimization, order generation, and KPI prediction, based on the updated inventory pipeline after transshipment.
4. Run the jobs to populate the tables that are required to display information about the user interface of the solution.

For more information about each job, see Appendix 2, “ETL Job Details,” on page 31.

Attribute Jobs

The attribute tables for facility and item are loaded periodically with the latest incremental data from the dimension tables. The attribute jobs load the required attribute description from the dimension and lookup tables and display the information about the user interface of the solution.

Here is a list of attributes that are loaded in the attribute tables:

- facility attributes
  - facility ID
  - facility name
  - facility type code and its description
  - organization ID
  - organization name
  - facility location hierarchy
- item attributes
  - item ID
  - item name
  - finished goods and assembly indicator
  - item type code and its description
  - make or buy code and its description
  - item status code and its description
  - item group code and its description
  - item category hierarchy

For more information about all the tables and their columns, see *SAS Inventory Optimization: Data Dictionary*.

For more information about each attribute job, see Appendix 2, "ETL Job Details," on page 31.

Inventory Optimization: UIART Jobs

Overview

The UIART jobs are categorized into jobs for the following workspaces:

- Inventory Analysis
- Scenario Development
- Order Suggestions

Inventory Analysis Workspace

You can view details of inventory metrics in the Inventory Analysis workspace. If you have a daily replenishment, then you need to decide whether you want to view these details about a weekly basis or a monthly basis by updating the GL_DAY_METRIC_LOAD global parameter. The ETL jobs calculate inventory metrics and then display the values on the user interface of the solution.

Inventory metrics are calculated in three steps:

1. Metrics or key performance indicators (KPIs) and their attributes are calculated. For inventory optimization, the SAS Inventory Optimization process calculates the following metrics:
   - lead time

- service level downstream
- service level upstream
- demand
- inventory cost
- inventory quantity
- item price
- inventory turns
- reorder level
- order-up-to level
- safety stock

All the metrics are calculated for the base period and are derived for the period under consideration. For example, if the base period is week, then start date and end date of the previous week are retrieved, and the metrics are calculated over this period. For daily replenishment, the metrics are calculated on a daily basis.

For more information about calculation of each metric,see "Calculations of Inventory Metrics and Thresholds" on page 55.

2. The inventory metric periodic table is loaded. The periodic table is required for detailed view of the inventory on the user interface. The periodic table stores aggregated and definite metric values for different facility and pairs in a data table. A history table is also loaded that stores the history of all the actual metric values before the run date of the base period. For daily replenishment, the metrics are aggregated based on the value of the GL_DAY_METRIC_LOAD global parameter.
3. The inventory metric daily tables are loaded. To review the performance of the service supply chain, information in the Inventory Analysis workspace must be monitored and analyzed per day. This requires that the information be loaded and updated on a daily basis. The daily tables with inventory metric details include the following information:

   Alerts
   : include the alert limits for lead time, service level, and demand metrics. You can specify and control these limits for different item categories and facilities.

   Metric analytical result table (ART)
   : includes the average metric values with their actual, target, and threshold values for every period.

   Item category hierarchical data
   : includes the aggregated metric values for all items and facilities, grouped by item categories.

   Facility-level data
   : contains the aggregated metric values for all items, grouped by the corresponding facilities. Using this information, the data table displays metric values for multiple items, single facility combination on the user interface of the solution.

   Network-level data
   : includes the number of networks that contain a facility.

Scenario Development Workspace

You can develop and work with scenarios in the Scenario Development workspace. Scenario analysis enables you to conduct what-if analysis and solve operational

dilemmas. For example, you can analyze the impact of change in service level, lead time, demand, and unit cost on the inventory cost and other cost metrics. You can perform four types of analysis. Each analysis is used for different purposes.

- Service level sensitivity analysis:

  In this type of scenario, you can select a facility, group of items, and specify the range of service level. This analysis uses the MIRP procedure to provide optimized inventory cost for the selected range of service level, for the specified facility and group of items.

- Customer-facing facility analysis:

  By performing this analysis, you can maximize the target service level for a selected group of items and customer-facing facilities, subject to the inventory cost. You can create a scenario by selecting items and customer-facing facilities. You can also specify the budget constraint and the minimum service level for the selected group of facility and item pairs. The ESLOPT procedure is used for optimizing the service level for customer-facing facilities. You can view three types of metric values on the user interface:

  - current values. Current metric values are obtained by running the procedure with the current service level value as specified in the SDL and without any budget constraint.
  - optimized values. Optimized metric values at all the selected customer-facing facilities are obtained by running the procedure with specified budget and minimum service level constraints. If you do not provide a budget, then the total cost across the selected items and facilities is based on the current service level target is taken as the budget. If a minimum service level value is not specified, then the procedure runs without a minimum service level constraint.
  - new values. By default, new metric values contain the optimized values. You can override the optimized service level values by specifying new values through the user interface. With these new values, service level values for the remaining facility and item pairs are optimized.

- Internal facility service level analysis:

  In this type of scenario, you can select items, facilities, and networks for analysis. The entire network that is associated with the selected items and facilities is analyzed. The analysis uses the MIRP procedure to optimize service levels at all internal facilities. The procedure uses the service level values for the customer-facing facilities. You can view three types of metric values:

  - current values. Current metric values are obtained by running the MIRP procedure with objective EVALISL for the current service levels that are specified in the SDL.
  - optimized values. Optimized metric values are obtained by running the procedure with objective OPTISL.
  - new values. By default, new metric values contain the optimized values. You can override the optimized service level values by specifying new values through the user interface. With the new values, the other metric values are evaluated. The evaluated values are obtained by running the MIRP procedure with objective EVALISL.

- Ad hoc analysis:

  Ad hoc analysis enables you to evaluate the impact of input parameters such as lead time, service level, projected demand, and unit cost of an item on policy parameters and costs. You can create input sets with different combinations of these input

parameter values and then compare the current and optimized values in each set. The MIRP procedure with objective EVALISL evaluates the cost metrics for the changed values of metrics such as service level and lead time.

For more information about calculations of the scenario-specific metrics, see "Calculations of the Scenario-Specific Metrics" on page 63.

Order Suggestions Workspace

SAS Inventory Optimization provides information about how much to order and when to receive the order. Recommendations about the quantities to be ordered from the primary and alternate channels for all facility and item pairs are also provided.

The order quantities from the primary channel are available as an output of the MIRP procedure. Order quantities from the alternate channels are available as an output of the transshipment process. Based on the order quantities that are recommended by the underlying process, the facility and item pairs are grouped into the following categories:

- Normal. Items in this category have the following features:
  - The average projected service level is higher than or equal to the lower bound of the target service level.
  - Orders are not placed and no transshipments are received.
  - Inventory positions are within the range of reorder level and order-up-to level.
- Overstock. Items in this category have the following features:
  - The average projected service level is higher than or equal to the lower bound of the target service level.
  - Orders are not placed and no transshipments are received.
  - Inventory positions are higher than the order-up-to level.
- Primary. Items in this category have the following features:
  - The average projected service level is higher than or equal to the lower bound of the target service level.
  - Orders are placed from the primary suppliers only and no transshipments are received.
- Primary and Alternative. Items in this category have the following features:
  - The average projected service level is higher than or equal to the lower bound of the target service level.
  - Transshipments are received from the inventory pool.
- Partial. Items in this category have the following features:
  - The average projected service level is less than the lower bound of the target service level.
  - Excess inventories are available in the inventory pool and shipping is possible. However, the transshipment module does not suggest the orders because they are not cost effective. You can ignore the cost impact and receive inventory from the inventory pool to fix the shortage problem.
- Incomplete. Items in this category have the following features:
  - The average projected service level is less than the lower bound of the target service level.

- Excess inventories are not available in the inventory pool or the delivery time is not feasible. Even if you are willing to pay the high cost of orders, the inventory is not sufficient to fix the shortage problem. You need to order the items from locations that are outside the inventory pool or from external vendors to fix the shortage problem.

The facility and item pairs in the primary, primary and alternative, partial, and incomplete categories are called as low stock facility and item pairs. For each such facility and item pair, the system generates a unique identifier (ID). These IDs consist of seven characters followed by a sequential eight-digit number (for example, Primary00000001, PrimAlt00000001, Partial00000001, and Incompl00000001).

In case of multiple vendors, for the vendor-facing facility and item pairs, the order quantity from the primary source is split among the vendors based on their supply share percentage.

Orders can be edited through the user interface. You can modify the order amount of an order, split an order, modify the transfer mode, and assign a vendor to an order.

Note: If your system uses firebird 2.0 with sasts engine and the transfer mode contains double-byte character set data, then editing the order can cause right truncation error. In such a scenario, you might not be able to save the edited order.

Appendix 1

Global and Job Parameter Table Details

Global Parameter Table

The following table lists all the global parameters. You can change these parameters as per your requirement.

Table A1.1 *Global Parameters and Their Descriptions*

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
History management process	GLOBAL_ HIGH_DTTM_ VALUE	Specifies a globally conventional high date value (datetime format) to identify valid records. You can change this date value as per your requirement.	"01JAN5999:00: 00:00"DT

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
Extract, transform, and load (ETL) process	GL_SHORT_ YES	Specifies a short value to indicate yes.	Y
	GL_SHORT_NO	Specifies a short value to indicate no.	N
	GL_WK_BASE_ PERIOD	Specifies a short value to indicate week as the base period.	W
	GL_MTH_ BASE_PERIOD	Specifies a short value to indicate month as the base period.	M
	GL_QTR_BASE_ PERIOD	Specifies a short value to indicate quarter as the base period.	Q
	GL_BASE_ PERIOD	Specifies a short value to indicate the base period for inventory optimization. The possible values are as specified in the GL_WK_BASE_PERIOD, GL_MTH_BASE_PERIOD, and GL_QTR_BASE_PERIOD parameters. You must set this parameter to your selected base period value. Ensure that this value is similar to the value in the Base_Period_Flg column of the SDL.Network_Model table.	W
	GL_SHORT_ BOTH	Specifies a short value to indicate that the facility is both internal and external	B
	GL_SHORT_ EXTERNAL	Specifies a short value to indicate that the facility is external.	E

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
Extract, transform, and load (ETL) process	GL_ FRAMEWORK_ DB	Specifies whether SAS Framework Data Server is used. *Note:* You must always set this parameter to N because MySQL is used as a transactional database.	N
	GL_DB_ HOSTNAME	Specifies the host name where 3rd party database is installed.	"puneisblademis 1.apac.sas.com"
	GL_DB_DSN	Specifies the data source name for connecting to transactional database server.	'iodb'
	GL_DB_ DATABASE	Specifies the transactional database server.	'iodb'
	GL_DB_AUTH_ DOMAIN	Specifies the authentication domain for the transactional database server.	'mysqlauth'
Extract, transform,and load (ETL) process	GL_SHORT_ INTERNAL	Specifies a short value to indicate that the facility is internal.	I
	GL_DAY_ BASE_PERIOD	Specifies a short value to indicate day as the base period.	D
	GL_SAS_ META_CFG_ PATH	Specifies the path of the framework server parameter file. By default, the file is saved at the location **<SASCONFIG>**\Lev**<N>**\AppData \SASInventoryOptimization during the installation process.	For a Windows operating environment: **C: ** For a UNIX operating environment: **Home/**

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
Inventory optimization process	GL_PLANNING_HORIZON	Specifies the number of base periods (any value > 0) for which policy parameters, key performance indicators, or both are computed for inventory optimization. You must set this parameter to a value >= max (lead time as per base period) + 1.	6
	GL_IO_BATCH_CURR_DATE	Specifies the current date that is used to populate the inventory data and demand data for inventory optimization. The default date is the current date. However, a specific date can be provided (for example, "01JAN2011"D). You must change this parameter value to the date from when your inventory optimization process starts.	date()
	GL_MIRP_REPLICATIONS	Specifies the number of simulation replications to be used in inventory optimization	200
	GL_SERVICE_TYPE	Specifies the service type that is used to calculate the service level. The possible values are FR for fill rate and RR for ready rate. You must modify this parameter as per your selected service type.	FR
	GL_SERVICE_TYPE_DESC	Specifies the description for the service type that is selected in the GL_SERVICE_TYPE global variable.	Fill rate

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
Inventory optimization process	GL_NODE_ PRICE_TYPE_ CD	Specifies the price type code to be used in node data for inventory optimization. You must modify this parameter to the price type code as per your data.	STD
	GL_IO_MIN_CV	Specifies the minimum value of coefficient of variation that is related to demand forecast. If the coefficient of variation for a demand forecast value is less than the specified minimum value of coefficient of variation, then the MIRP procedure outputs a warning message. The variance of the demand value is increased to meet the value in this parameter. You must modify this parameter as per your requirement.	0.01
	GL_IO_MAX_ CV	Specifies the maximum value for coefficient of variation that is related to demand forecast. If the coefficient of variation for a demand forecast value is greater than the specified maximum value of coefficient of variation, then the MIRP procedure outputs a warning message. The variance of the demand value is decreased to meet the value in this parameter. You must modify this parameter as per your requirement.	1
	GL_POOLING_ NETWORK_FLG	Specifies whether the network is a pooling network by comparing the value of the network flag with the value of this variable.	P

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
Inventory optimization process	GL_IO_KIT_ITEMS_IND	Indicates whether the kit items are to be handled in the inventory optimization process or not. If you want to handle kit items, then you must specify this parameter value as 1, else the value must be 0.	1
	GL_IO_COMPARE_FCST_ACT_IND	Indicates whether forecasted demand is to be compared with the actual demand or not. If actual demand is greater than the forecasted demand, then the actual demand will be used in the inventory optimization process. If you want to compare the forecasted demand with the actual demand, then you must specify this parameter value as 1, else the value must be 0.	1
	GL_IO_RUN_TRANSSHIPMENT_IND	Indicates whether the transshipment process is to be run or not. If you want to run the transshipment process, then you must specify this parameter value as 1, else the value must be 0.	1
	GL_IO_USE_MAX_ORDER_QTY_IND	Indicates whether the maximum order quantity constraint is to be used in the inventory optimization process or not. If you want to use the constraint, then you must specify this parameter value as 1, else the value must be 0.	1
	GL_IO_USE_SCEN_PROMOTED_VAL_IND	Indicates whether the promoted values of a scenario are to be used in the inventory optimization or not. If you want to use the promoted values, then you must specify this parameter value as 1, else the value must be 0.	1
	GL_USE_BACKORDER	Specifies whether back orders are to be added to the item demand for the first period or not. If you want to add back orders, then you must specify this parameter value as 1, else the value must be 0.	Y

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
User interface	GL_ CURRENCY_ SYMBOL	Specifies the symbol or abbreviation for currency. You must modify this parameter as per your currency.	USD
	GL_IO_ METRICS_ HISTORY_ PERIOD	Specifies the number of base periods of historical data to be displayed in the Inventory Analysis workspace. You must modify this parameter as per your requirement.	2
	GL_DEMAND_ THRESHOLD_ COEFF	Specifies the coefficient value (a positive number < 3) for the given Confidence Interval (CI). The coefficient value is used to calculate the lower and upper bounds for the forecasted demand. You can modify the coefficient value as per your requirement.	1.96
User interface	GL_IO_ITEM_ CAT_LVLNO	Specifies the number of levels for item categories to be displayed in the Inventory Analysis workspace. You must change this parameter as per the number of hierarchical levels in the item category.	3
	GL_LOOSE_ CNTRL_TYPE	Specifies the loose receipt control type that is used to determine the period boundaries for calculating the service level.	LOOSE
	GL_TIGHT_ CNTRL_TYPE	Specifies the tight receipt control type that is used to determine period boundaries for calculating the service level.	TIGHT
	GL_RECEIPT_ CNTRL_TYPE	Specifies receipt control type (possible values are LOOSE or TIGHT) that is used to determine period boundaries for calculating the service level. You can modify this parameter as per your requirement.	LOOSE
	GL_SERVICE_ THRESHOLD	Specifies the threshold to calculate the lower and upper bounds of service level. You can modify this threshold value as per your requirement.	0.02

Category (Parameter Used In)	Parameter Name	Parameter Description	Default Value
User interface	GL_MAX_ BUDGET	Specifies a high value for the inventory cost budget to be used for customer-facing facility analysis in the Scenario Development workspace.	1.00E+15
	GL_MIN_ DECREASE_ COST_PCT	Specifies the maximum percentage up to which the unit cost of an item can be decreased in ad hoc analysis in the Scenario Development workspace. You can modify this parameter value as per your requirement.	10
	GL_DAY_ METRIC_LOAD	Specifies the base period that is used to perform aggregation of the metrics that are displayed on the Inventory Analysis workspace when daily replenishment is selected.	W

Job Parameter Table

The following table lists the jobs and their parameters that need to be specified.

Table A1.2 *Job Parameters and Their Descriptions*

Job Name or Directory	Parameter Name	Parameter Description	Default Value
03_CREATE_TIME_ PERIOD_DATA	CAL_YR_ START_DT	Specifies the start date of the year from which data would be generated.	'01JAN2000'D
	CAL_NO_OF_ YRS	Specifies the number of years for calendar time dimension.	15
	SUMM_NO_OF_ YRS	Specifies the number of years for which the summarized time dimension is to be populated. This is used in long-term forecasting.	15
	LOADING_ DTTM	Specifies the date time value to be loaded.	'01JAN2000:00:00 :00'DT

Appendix 2

ETL Job Details

Overview

You can access all ETL jobs from SAS Data Integration Studio. The basic location of all ETL jobs except the initial one-time jobs for SAS Inventory Optimization is as follows: **`/Products/SAS Inventory Optimization/SAS Inventory Optimization 5.1`**.

The ETL jobs are placed under specific folders in this location. The subsequent topics explain the ETL jobs and provide only the specific location for the jobs. For example, the location for the initial one-time jobs is given as **`CONTROL/INITIAL_ONE_TIME_JOBS`**. This folder structure is available under the basic location of the ETL jobs.

Note: The input requirements that are mentioned for each job are directly used by that job. You do not need to provide any inputs or perform any prerequisite tasks.

Initial ETL Job Sequence

The following table lists the sequence of the ETL jobs that are used in SAS Inventory Optimization. You must follow this sequence when you are planning to run the jobs for the first time, but after you run the initial one-time jobs. Later, each job is to be run either daily or on the basis of the base period. For more information about the initial one-time jobs, see "Initial One-Time Jobs" on page 35.

The value that is displayed in `monospace` format is the directory or container for the subsequent jobs. The Predecessors column lists the job numbers of the jobs that must be run before a particular job is run.

Table A2.1 *ETL Job Sequence*

Job Number	Job Name	Predecessors	Comments
Dimension Jobs			
IO_JOBS/01_DIMENSION_JOBS			
1	FACILITY_DIM	None	None
2	ITEM_DIM	None	None
Inventory Optimization Jobs			
IO_JOBS/02_ANALYTICS_JOBS/IO/A_CREATE_IO_ABT			
3	01_POPULATE_NODE_DATA	Need inputs	None
4	02_POPULATE_ARC_DATA	None	
5	03_POPULATE_DEMAND_DATA	Need inputs	
6	04_POPULATE_INVENTORY_DATA	None	
IO_JOBS/02_ANALYTICS_JOBS/IO/B_IO_PROCESS			
7	01_POLICY_ORDER_KPI_1	3–6	None
8	02_TRANSSHIPMENT	7	
9	03_POLICY_ORDER_KPI_2	8	
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/01_ATTRIBUTE_JOBS			
10	FACILITY_ATTR	None	None
11	ITEM_ATTR		

Job Number	Job Name	Predecessors	Comments
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/02_METRICS_CALC			
12	01_GET_LEAD_TIME	None	None
13	02_GET_SL_DOWNSTREAM		
14	03_GET_SL_UPSTREAM		
15	04_GET_DEMAND_ACT		
16	05_GET_DEMAND_TGT		
17	06_GET_COST_ACT		
18	07_GET_COST_TGT		
19	08_GET_KPI_PREDICT_MIN_MAX		
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/03_METRICS_BASE_PERIOD			
20	01_IO_METRICS_PERIODIC_ART_HIST	None	None
21	02_IO_METRICS_PERIODIC_ART_LOAD		
22	03_IO_METRICS_PERIODIC_ART_REPLACE		
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/04_METRICS_DAILY			
23	01_IO_METRICS_ALERT_SETTINGS_LOAD	None	None
24	02_IO_METRICS_ALERT_SETTINGS_REPLACE		
25	03_IO_METRICS_ART_LOAD		
26	04_IO_METRICS_ART_VENDOR_DETAILS		
27	05_IO_METRICS_ART_REPLACE		
28	06_IO_METRICS_ITEM_FACITY_NETWORK		
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/05_SCENARIO_BASE_PERIOD			
29	SCENARIO_UI_TABLES	None	None

Job Number	Job Name	Predecessors	Comments
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/06_ORDERS_BASE_PERIOD			
30	01_CREATE_LOCK_PLAN_SETTING	None	Run this job only once, after you populate all tables of the SDL library.
31	02_CREATE_SKU_BUCKETS	9, 11, 24	None
32	03_POPULATE_DRP_TABLE	31	
33	04_POPULATE_TRANSFER_COST_DETAILS	32	
34	05_POPULATE_SUBSTITUTE_ITEM	33	
35	06_POPULATE_ORDER_RESULT	34	
36	07_POPULATE_REPL_BY_PERIOD_LOOKUP	35	None
37	08_POPULATE_ORDER_DETAIL	36	
38	09_POPULATE_REPL_PLAN_DETAIL	37	
39	10_POPULATE_REPL_PLAN_METRICS	38	
40	11_CREATE_REPL_PLAN_SUMMARY	39	
41	12_POPULATE_BUYER_ORDER_RESULT	40	
42	13_POPULATE_BUYER_REPL_PLAN_SUMMARY	41	
IO_JOBS/02_ANALYTICS_JOBS/IO/C_UI_ART_JOBS/07_ORDERS_OPTIONAL			
43	LOAD_ORDER_DETAILS	39	Run this job only if you want to promote all locked orders.

Initial One-Time Jobs

The following table provides an overview of the initial one-time jobs. These jobs need to be run only once, after you install the solution.

Table A2.2 *Initial One-Time Jobs*

Job Name	Job Description	Input Requirements
01_CREATE_GLOBAL_PARAMETER_LIST	This job populates the Control.Global_Parameter_List table and globally initializes all the parameters.	None
02_CREATE_CONTROL_TABLES	This job creates and populates the control tables by using the script files **`load_spo_io_control_data.sas`** and **`create_spo_fcst_control_table.sas`**. These tables are used by the forecasting and inventory optimization process. This job also creates the Control.Job_Status table.	None
03_CREATE_TIME_PERIOD_DATA	This job generates data for Time_Period, Time_Period_Assoc, Time_Period_Assoc_Type, and Period_Type tables of the solution data layer (SDL). The job also creates hierarchies by using associations between the various time period entities. • General Calendar: Day → Week • Summarized Calendar: Day → Month → Quarter → Year	You can specify parameters such as start date and number of years for which data is to be generated, as values of the parameters for the job.
04_CALENDAR_HIERARCHY1	This job reduces the levels in the calender hierarchy1 (Day → Week) and stores the output in the Scratch.Calendar_Flatten_Hier1 table.	The tables SDL.Time_Period and SDL.Time_Period_Assoc are required.

Job Name	Job Description	Input Requirements
05_CALENDAR_ HIERARCHY2	This job reduces the levels in the calender hierarchy2 (Day → Month → Quarter → Year) and stores the output in the Scratch.Calendar_Flatten_ Hier2 table.	The tables SDL.Time_Period and SDL.Time_Period_Assoc are required.
06_TIME_DIM	This job populates the time dimension table DIM.Time_Dim.	The tables Scratch.Calendar_Flatten_ Hier1 and Scratch.Calendar_Flatten_ Hier2 are required.
07_CREATE_TIME_VIEWS	This job creates week, month, quarter, and year time views from the DIM.Time_Dim table.	The tables DIM.Time_Dim DIM.Week_Dim, DIM.Month_Dim, DIM.Quarter_Dim, DIM.Year_Dim are required.
08_CREATE_USER_ TABLES	This job creates the following initial tables that are required for populating the user interface: • UIART.User_Mapping • TSDB.User_Preferences • TSDB.IO_Metrics_ Alertsettings • TSDB.Scenarios • TSDB.Scenario_Params • TSDB.Evaluate_Forecast_ Parameter • UIART.Evaluated_ Timeseries_Result • TSDB.Forecast_Data_ Param	None

Dimension Jobs

The following table provides an overview of the dimension jobs. These jobs must be run whenever the input tables that are required by these jobs are either loaded for the first time or updated later.

Table A2.3 *Dimension Jobs*

Job Name	Job Description	Input Requirements
FACILITY_DIM	This job populates the dimension table for a facility. The job extracts the latest records from the facility table, and then loads the attributes in the Facility_Dim table.	The tables SDL.Facility, SDL.Location, and SDL.Organization are required.
ITEM_DIM	This job populates the dimension table for an item. The job extracts the latest records from the source item table, and then loads the attributes in the Item_Dim table.	The tables SDL.Item and SDL.Item_Category are required.

Inventory Optimization Jobs

Input Data Preparation Jobs

The following table provides an overview of the jobs for preparing data for the inventory optimization process. These jobs must be run as per the base period.

Table A2.4 *Input Data Preparation Jobs*

Job Name	Job Description	Input Requirements
01_POPULATE_NODE_DATA	This job populates the tables ABT.IO_**<BASE_PERIOD>**_Node_Data_ABT and ABT.IO_**<BASE_PERIOD>**_Node_Data_Hist_ABT for the inventory optimization process.	The following tables are required: • SDL.Network_Model • SDL.Item • SDL.Item_Price • SDL.Facility_X_Item • SDL.Route • SDL.Network_X_Route • SDL.Route_X_Item • SDL.Facility_Item_X_Network • SDL.Item_Succession • SDL.BOM

Job Name	Job Description	Input Requirements
02_POPULATE_ARC_DATA	This job populates the arc data in the table ABT.IO_**<BASE_PERIOD>**_ARC_Data_ABT for inventory optimization process.	The following tables are required: • SDL.Network_Model • SDL.Item • SDL.Facility_X_Item • SDL.Route • SDL.Network_X_Route • SDL.Route_X_Item • SDL.Facility_Item_X_Network • SDL.Item_Succession • SDL.BOM
03_POPULATE_DEMAND_DATA	This job populates the demand data in the table ABT.IO_**<BASE_PERIOD>**_Demand_Data_ABT for inventory optimization process.	The following tables are required: • SDL.Network_Model • SDL.Item • SDL.Item_Price • SDL.Facility_X_Item • SDL.Facility_Item_X_Network • SDL.Item_Succession • SDL.Back_Order_Summary • SDL.BOM • ABT.**<BASE_PERIOD>**_Aggregated_Forecast_ART • DIM.Time_Dim

Job Name	Job Description	Input Requirements
04_POPULATE_ INVENTORY_DATA	This job populates the inventory data in the tables ABT.IO_**<BASE_PERIOD>** _Inventory_Data_ABT, ABT.IO_**<BASE_PERIOD>** _Opening_Inventory_ABT, and ABT.IO_**<BASE_PERIOD>** _Pipeline_Inventory_ABT for inventory optimization process. If you want to merge the inventory of the predecessor with that of the successor, then in the user-written code for this job, specify the UseItemSuccession parameter to GL_Short_Yes, else specify the parameter to GL_Short_No.	The following tables are required: • SDL.Network_Model • SDL.Item • SDL.Facility_X_Item • SDL.Facility_Item_ Inventory • SDL.Pipeline_Inventory • SDL.Item_Succession • SDL.BOM • DIM.Time_Dim

Inventory Optimization Process Jobs

The following table provides an overview of the jobs that are used for running the inventory optimization process. These jobs must be run as per the base period.

Table A2.5 *Inventory Optimization Process Jobs*

Job Name	Job Description	Input Requirements
01_POLICY_ORDER_KPI_1	This job runs the inventory optimization process before transshipment and populates the following tables: • ABT.MIRP_**<BASE_PERIOD>**_Node_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_ARC_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_Demand_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_Inventory_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_OPT_Message_ART • ABT.MIRP_**<BASE_PERIOD>**_OUT_Beforetrans	The following tables are required: • ABT.IO_**<BASE_PERIOD>**_Node_Data_ABT • ABT.IO_**<BASE_PERIOD>**_ARC_Data_ABT • ABT.IO_**<BASE_PERIOD>**_Demand_Data_ABT • ABT.IO_**<BASE_PERIOD>**_Inventory_Data_ABT • Control.IO_Batch_Attributes

Job Name	Job Description	Input Requirements
02_TRANSSHIPMENT	This job runs the transshipment process and populates the following tables: • TRANS_**<BASE_PERIOD>**_Node_Data_ABT • TRANS_**<BASE_PERIOD>**_Nodearc_Data_ABT • TRANS_**<BASE_PERIOD>**_Alertdata_IN_ABT • TRANS_**<BASE_PERIOD>**_Alertdata_Type_ABT • Transshipment_**<BASE_PERIOD>**_ART • Transshipment_Cost_**<BASE_PERIOD>**_ART • Transshipment_**<BASE_PERIOD>**_Summary_ART • TRANS_**<BASE_PERIOD>**_Shipin_Period_Summary • TRANS_**<BASE_PERIOD>**_Shipout_Summary_ART • TRANSSHIP_**<BASE_PERIOD>**_Shipin_Summary	The following tables are required: • ABT.MIRP_**<BASE_PERIOD>**_Node_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_ARC_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_Demand_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_Inventory_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_OUT_Beforetrans

Job Name	Job Description	Input Requirements
03_POLICY_ORDER_KPI_2	This job runs the inventory optimization process after transshipment process and populates the tables ABT.MIRP_**<BASE_PERIOD>**_Predict_KPI_ART. and ABT.MIRP_**<BASE_PERIOD>**_Predict_KPI_HIST_ART	The following tables are required: • ABT.MIRP_**<BASE_PERIOD>**_Node_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_ARC_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_Demand_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_Inventory_Data_ABT • ABT.MIRP_**<BASE_PERIOD>**_OUT_Beforetrans • TRANS_**<BASE_PERIOD>**_Shipin_Period_Summary • TRANS_**<BASE_PERIOD>**_Shipout_Summary_ART

Inventory Optimization UIART Jobs

Attribute Table Creation Jobs

The following table provides an overview of the jobs that create the item and facility attribute tables. These jobs must be run as per the base period.

Table A2.6 *Attribute Table Creation Jobs*

Job Name	Job Description	Input Requirements
FACILITY_ATTR	This job creates the facility attribute table UIART.Facility_ATTR that is used to display attribute details for facilities on the user interface. The job also loads code descriptions for the facilities from the lookup detail table.	The tables DIM.Facility_Dim, SDL.Lookup_Detail, and ABT.MIRP_<BASE_PERIOD>_Predict_KPI_ART are required.

Job Name	Job Description	Input Requirements
ITEM_ATTR	This job creates the item attribute table UIART.Item_ATTR that is used to display attribute details for items on the user interface. The job also loads code descriptions for the items from the lookup detail table.	The tables DIM.Item_Dim, SDL.Lookup_Detail, and ABT.MIRP_<BASE_PERIOD>_Predict_KPI_ART are required.

Metric Calculation Jobs

The following table provides an overview of the jobs that calculate the metrics or key performance indicators (KPI). These jobs must be run as per the base period.

Table A2.7 *Metric Calculation Jobs*

Job Name	Job Description	Input Requirements
01_GET_LEAD_TIME	This job loads the intermediate UIART tables Get_Lead_Time_ACT_TGT and Get_Lead_Time_Min_Max with the following lead time attributes from SDL and ABT sources: • ACTUAL_LEAD_TIME • TARGET_LEAD_TIME • LEAD_TIME_MAX • LEAD_TIME_MIN	The tables SDL.Purchase_Order, SDL.Receipts, ABT.MIRP_<BASE_PERIOD>_Predict_KPI_Hist_ART, and ABT.MIRP_<BASE_PERIOD>_Node_Data_ABT are required.
02_GET_SL_DOWNSTREAM	This job loads the intermediate UIART table SL_Downstream with the following downstream service level attributes from SDL and ABT sources: • ACTUAL_SL_DOWNSTREAM • TARGET_SL_DOWNSTREAM	The tables SDL.Customer_Order, SDL.Dispatch, ABT.MIRP_<BASE_PERIOD>_Predict_KPI_Hist_ART, and ABT.IO_<BASE_PERIOD>_Node_Data_ABT are required.
03_GET_SL_UPSTREAM	This job loads the intermediate UIART table Get_SL_Upstream with following upstream service level attributes from SDL and ABT sources: • ACTUAL_SL_UPSTREAM • TARGET_SL_UPSTREAM	The tables SDL.Purchase_Order, SDL.Receipts, ABT.MIRP_<BASE_PERIOD>_Predict_KPI_Hist_ART, and ABT.IO_<BASE_PERIOD>_Node_Data_ABT are required.

Job Name	Job Description	Input Requirements
04_GET_DEMAND_ACT	This job loads the intermediate UIART table Get_Demand_ACT with the actual demand from the facility and item demand. The actual demand is calculated as sum of the order quantity from the Facility_Item_Demand table in the required time period.	The tables DIM.Time_Dim and SDL.Facility_Item_Demand are required.
05_GET_DEMAND_TGT	This job loads the intermediate UIART tables TGT_Demand_Temp and Get_Demand_TGT with the target demand from the ART source. The target demand KPI is calculated as the summation of the external and internal demand mean from the output of the MIRP procedure.	The tables ABT.MIRP_<BASE_PERIOD>_Predict_KPI_Hist_ART and ABT.MIRP_<BASE_PERIOD>_Predict_KPI_ART are required.
06_GET_COST_ACT	This job loads the intermediate UIART table Get_Inventory_Cost_ACT with the following actual metrics from the inventory and item price sources: • Actual inventory quantity • Actual inventory cost • Item price	The tables DIM.Time_Dim, SDL.Facility_Item_Inventory, and SDL.Item_Price are required.
07_GET_COST_TGT	This job loads the intermediate UIART tables TGT_Cost_Temp and Get_Inventory_Cost_TGT with the following target metrics from the SDL and ART sources: • Inventory cost • Reorder level • Order-up-to level • Safety stock	The tables ABT.MIRP_<BASE_PERIOD>_Predict_KPI_ART, ABT.MIRP_<BASE_PERIOD>_Predict_KPI_Hist_ART, and SDL.Item_Price are required.

Job Name	Job Description	Input Requirements
08_GET_KPI_ PREDICT_MIN_MAX	This job loads the intermediate UIART tables Get_KPI_Min_Max_Period and Get_KPI_Min_Max_History with following upper and lower bound attributes from the intermediate UIART and ART sources: • SL_LB • SL_UB • Demand_UL • Demand_LL	The tables ABT.MIRP_<BASE_ PERIOD>_Predict_KPI_ART and ABT.MIRP_<BASE_ PERIOD>_Predict_KPI_Hist_ ART are required.

Inventory Metrics Periodic Jobs

The following table provides an overview of the jobs that populate the inventory metrics periodic table. These jobs must be run as per the base period.

Table A2.8 *Inventory Metrics Periodic Jobs*

Job Name	Job Description	Input Requirements
01_IO_METRICS_ PERIODIC_ART_HIST	This job integrates all the metric calculations and stores the results in the IO_Metrics_Periodic_ART_ HIST table. This table stores the history data, current data, and forecasted data for all periods. The following metrics are loaded in the target history table UIART.IO_Metrics_Periodic_ ART_HIST: • LT_ACT • LT_TGT • DEMAND_ACT • DEMAND_FORECAST • SL_UPSTREAM_ACT • SL_UPSTREAM_TGT • SL_DOWNSTREAM_ACT • SL_DOWNSTREAM_TGT • COST_ACT • COST_TGT	The following tables are required: • ABT.IO_<BASE_ PERIOD>_NODE_DATA_ ABT • UIART.GET_LEAD_ TIME_act_tgt • UIART.SL_ DOWNSTREAM • UIART.GET_SL_ UPSTREAM • UIART.GET_DEMAND_ ACT • UIART.GET_DEMAND_ TGT • UIART.GET_ INVENTORY_COST_ACT • UIART.GET_ INVENTORY_COST_TGT • ABT.MIRP_<BASE_ PERIOD>__Predict_KPI_ Hist_ART

Job Name	Job Description	Input Requirements
02_IO_METRICS_ PERIODIC_ART_LOAD	This job loads data in the IO_Metrics_Periodic_ART_ Temp table only for the required period in accordance with the global parameter GL_IO_Metrics_History_ Period. This parameter specifies the number of periods of historical data to be displayed in the Inventory Metrics workspace.	The table UIART.IO_Metrics_Periodic_ ART_HIST is required.
03_IO_METRICS_ PERIODIC_ART_ REPLACE	This job replaces the IO_Metrics_Periodic_ART table by the _temp Periodic ART table that is already loaded. This replacement avoids any lock on the table when a query is issued from the user interface. The following metrics are loaded in the target table: • LT_ACT • LT_TGT • DEMAND_ACT • DEMAND_FORECAST • SL_UPSTREAM_ACT • SL_UPSTREAM_TGT • SL_DOWNSTREAM_ACT • SL_DOWNSTREAM_TGT • COST_ACT • COST_TGT	The table UIART.IO_Metrics_Periodic_ ART_Temp is required.

Jobs for the Inventory Metrics View and Alert Settings

The following table provides an overview of the jobs that load the inventory metrics ARTs and the alert settings. These jobs must be run on a daily basis.

Table A2.9 *Jobs for the Inventory Metrics View and Alert Settings*

Job Name	Job Description	Input Requirements
01_IO_METRICS_ ALERT_SETTINGS_ LOAD	This job integrates the minimum and maximum values from the intermediate UIART tables to load the IO_Metrics_Alert_Settings_ Temp table. The job populates the upper and lower limit values for the required alert metrics. The job loads the following bound values in the UIART.IO_Metrics_ Alertsettings_Temp table: • LT_UP_CNTRL_LIM • LT_LW_CNTRL_LIM • DEMAND_UP_CNTRL_ LIM • DEMAND_LW_CNTRL_ LIM • SL_UPSTREAM_UP_ CNTRL_LIM • SL_UPSTREAM_LW_ CNTRL_LIM • SL_DOWNSTREAM_UP_ CNTRL_LIM • SL_DOWNSTREAM_LW_ CNTRL_LIM	The following tables are required: • UIART.GET_LEAD_ TIME_min_max • UIART.GET_KPI_min_ max_period • ABT.IO_<BASE_ PERIOD>_NODE_DATA_ ABT
02_IO_METRICS_ ALERT_SETTINGS_ REPLACE	This job replaces the UIART.IO_Metrics_ Alertsettings_UIART table that is located on the forecast server by the corresponding temporary table that is already loaded. This replacement avoids any lock on the table when a query is issued from the user interface. The job only replaces those records in the TSDB.IO_Metrics_ Alertsettings table where `spoid = 0` .	The table UIART.IO_Metrics_ Alertsettings_Temp is required.

Job Name	Job Description	Input Requirements
03_IO_METRICS_ART_LOAD	This job loads the UIART.IO_Metrics_ART_Temp and UIART.IO_Metrics_ART_No_Vendor_Details tables and calculates the average of the metrics that are taken for the required period from the IO_Metrics_Periodic_ART table. The job also calculates the error high and error low columns by considering the upper control and lower control columns from the TSDB.IO_Metrics_Alertsettings table.	The following tables are required: • DIM.Item_Dim • DIM.Facility_Dim • SDL.Lookup_Detail • UIART.IO_Metrics_ART_Temp • TSDB.IO_Metrics_Alertsettings • SDL.Employee • UIART.IO_Metrics_Periodic_ART
04_IO_METRICS_ART_VENDOR_DETAILS	This job adds vendor details to the IO_Metrics_ART_No_Vendor_Details table to populate the UIART.IO_Metrics_ART_Temp_Vendor_DTLS table.	The following tables are required: • SDL.Facility_Item_X_Network • SDL.Vendor_Facility_Item • UIART.IO_Metrics_ART_No_Vendor_Details • SDL.Vendor
05_IO_METRICS_ART_REPLACE	This job replaces the UIART.IO_Metrics_ART table with the IO_Metrics_ART_Temp_Vendor_DTLS table that is already loaded to avoid any lock on the table when a query is issued from the user interface.	The table UIART.IO_Metrics_ART_Temp_Vendor_DTLS is required.

Job Name	Job Description	Input Requirements
06_IO_METRICS_ ITEM_FACITY_ NETWORK	This job creates the tables UIART.Item_Category_ HIER<n> (Hierarchy tables are created in accordance with the global parameter GL_IO_Item_Cat_LVLNO), UIART.Network_Facility, and UIART.AllFacilities. These tables are required for displaying information in the Inventory Metrics view of the Inventory Analysis workspace. This job calls the following macros. • %IO_METRICS_ITEM_ HIER_TABLES • %IO_METRICS_ FACILITY_ALL • IO_METRICS_ NETWORK_FACILITY The metrics are calculated for all the periods by using the input table.	The table UIART.IO_Metrics_ART is required.

Jobs for Scenario Development Workspace

The following table provides an overview of the job that loads the tables that are required to display information in the Scenario Development workspace. These jobs must be run as per the base period.

Table A2.10 *Job for the Scenario Development Workspace*

Job Name	Job Description	Input Requirements
SCENARIO_UI_TABLES	This job updates the stageout.Scen_promoted_val data set with latest promoted scenarios values. The job loads the following target output tables: • UIART.Facility_Face • UIART.Item_Vendor • UIART.I_X_F_ATTR • UIART.Item_Buyer_List • UIART.Buyers • UIART.Vendors These tables are required by the Scenario Development workspace with the SDL and ABT source tables.	The following tables are required: • SDL.Facility_X_Item • SDL.Facility_Item_X_Network • SDL.Vendor_Facility_Item • SDL.Employee • SDL.Vendor • ABT.IO_<BASE_PERIOD>_Node_Data_ABT

Note: The user can see the promoted values in stageout.Scen_promoted_val table after the next batch run.

Jobs for the Order Suggestions Workspace

The following table provides an overview of the jobs that load the tables that are used to display information in the Order Suggestions workspace. These jobs must be run as per the base period.

Table A2.11 *Jobs for the Order Suggestions Workspace*

Job Name	Job Description	Input Requirements
01_CREATE_LOCK_PLAN_SETTING	This job creates a table TSDB.Lock_Plan_Setting to store settings that enable automatic locking of the replenishment plans.	The SDL.Facility_X_Item table is required.

Job Name	Job Description	Input Requirements
02_CREATE_ SKU_BUCKETS	This job classifies the facility and item pairs into the following categories for the Order Suggestions workspace: • Normal • Overstock • Primary • Primary and Alternative • Partial • Incomplete The job populates the following tables from the UIART library: • Order_Bucket_Normal_Stock • Order_Bucket_Over_Stock • Order_Bucket_Low_Stock • Order_Bucket_Primary • Order_Bucket_Primary_Alternate • Order_Bucket_Partial • Order_Bucket_Incomplete • Order_Bucket_Type	The input and output tables that are created by the Inventory Optimization process jobs and the following tables are required: • SDL.Lookup_Detail • SDL.Route_Type_Ref • SDL.Vendor • UIART.Item_ATTR • UIART.I_X_F_ATTR • UIART.Facility_ATTR • UIART.Buyers • TSDB.IO_Metrics_ Alertsettings

Job Name	Job Description	Input Requirements
03_POPULATE_DRP_TABLE	This job populates the DRP tables UIART.Repl_By_Period_Normal, UIART.Repl_By_Period_Overstock, and UIART.Repl_By_Period_Lowstock for all the facility and item pairs to be displayed on the Order Suggestions workspace.	The input and output tables that are created by the Inventory Optimization Process jobs and the following tables are required: • SDL.Lookup_Detail • SDL.Route_Type_Ref • SDL.Vendor • UIART.Item_ATTR • UIART.I_X_F_ATTR • UIART.Facility_ATTR • UIART.Buyers • TSDB.IO_Metrics_Allsettings • SDL.Item_Substitute
04_POPULATE_TRANSFER_COST_DETAILS	This job populates the UIART.Order_Transfer_Cost table with the transfer cost details.	
05_POPULATE_SUBSTITUTE_ITEM	This job populates the UIART.Order_Substitute_Item table that is required for displaying information about the Order Suggestions workspace.	
06_POPULATE_ORDER_RESULT	This job populates the UIART.Order_Result table that is required for displaying information about the Order Suggestions workspace.	
07_POPULATE_REPL_BY_PERIOD_LOOKUP	This job populates the UIART_Repl_By_Period_Lookup lookup table that is required for displaying information about the Order Suggestions workspace.	
08_POPULATE_ORDER_DETAIL	This job populates the TSDB.Suggested_Order_Detail, UIART.Suggested_Order_Detail_Hist, and Stageout.Order_Detail tables with details of each order.	
09_POPULATE_REPL_PLAN_DETAIL	This job populates the TSDB.Repl_Plan_Detail table with details of each replenishment plan.	
10_POPULATE_REPL_PLAN_METRICS	This job populates the UIART.Repl_Plan_Metrics table with different metrics for displaying information about the Order Suggestions workspace. Some of the metrics include projected transfer cost, projected holding cost, and projected penalty cost.	
11_CREATE_REPL_PLAN_SUMMARY	This job creates replenishment plans summary table TSDB.Repl_Plan_Summary that is displayed on the Order Suggestions workspace.	

Job Name	Job Description	Input Requirements
12_POPULATE_BUYER_ORDER_RESULT	This job populates the UIART.BUYER_ORDER_RESULT table that is required to display information in the Buyers Portfolio.	
13_POPULATE_BUYER_REPL_PLAN_SUMMARY	This job creates the replenishment plans summary table TSDB.BUYER_REPL_PLAN_SUMMARY that is required for the Buyers Portfolio in the Order Suggestions workspace.	

Job for Loading Order Details in the Stageout Library

The following table provides an overview of the job that loads the order details in the Stageout library. These jobs must be run on a daily basis.

Table A2.12 *Job for Loading the Order Details*

Job Name	Job Description	Input Requirements
LOAD_ORDER_DETAILS	This job loads the order details in the table Stageout.Order_Detail.	The TSDB.Order_Detail table is required.

Appendix 3

Metric Calculations

Calculations of Inventory Metrics and Thresholds

Lead Time

Overview

Lead time is the time to physically deliver an order (full or partial) from the source location to the destination location. The different values of lead time are explained in the following subsections.

Actual Lead Time

Actual lead time is the transit time between the primary supplying channel and the receiving location. An average of lead times is calculated over the period under consideration.

The derivation for the actual lead time is as follows:

Actual lead time = Average (Receipt date – Dispatch date) for the order of a facility and item pair, considering it is a regular order in the primary network.

Consider the order receipts between the start date and end date of the previous period. If there is no delivery during that period, then the actual lead time is missing. Actual lead time values are missing for current and future periods also.

Target Lead Time

Target lead time is the average transportation time that is measured over the past periods between the source and the destination facilities through primary channels. This value is obtained from the source system and is available in the node data.

Lead Time Minimum

Lead time minimum or lead time lower control limit is the minimum time (in days) for an item to be transported to a facility. This value is obtained from the source system and is available in the node data.

Lead Time Maximum

Lead time maximum or lead time upper control limit is the maximum time (in days) for an item to be transported to a facility. This value is obtained from the source system and is available in the node data.

Lead Time Error High

If the lead time actual value is greater than the lead time maximum value, then lead time error high = 1, else the value is 0.

Lead Time Error Low

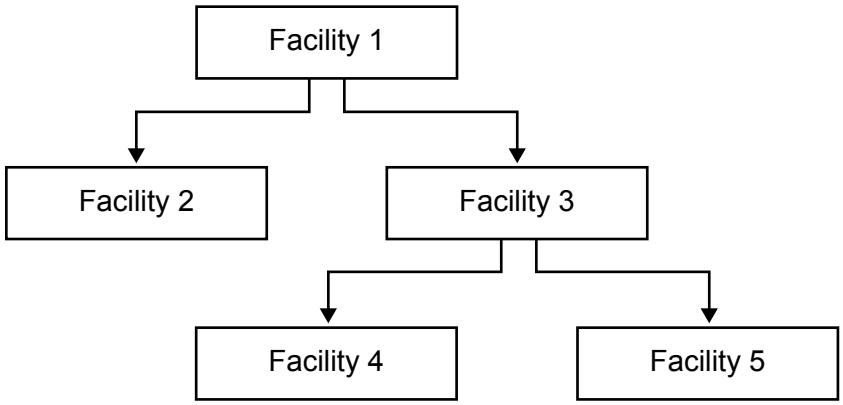
If the lead time actual value is less than the lead time minimum value, then lead time error low = 1, else the value is 0.

Downstream Service Level

Overview

Service level is a measure of the fulfillment of customer demand. Downstream service level is the service level provided by a facility to the downstream facilities or end customers. For example, consider the network in the following figure:

Figure A3.1 *A Sample Network*



The downstream service level for Facility 3 is the service level that is delivered to Facility 4 and Facility 5.

Actual Downstream Service Level

Actual downstream service level is the percentage service level that is calculated from the solution data layer for the facility. The average is calculated over the past period under consideration and is missing for the current and future periods.

The derivation for the actual downstream service level value is as follows:

- Consider all regular orders for an item in a facility where the delivery due date exists in the previous period. Receipt control type (possible values are LOOSE or TIGHT) is used to determine the period boundaries to calculate service level.
    - For facility and item pairs with loose receipt control type, the receipt date must be before or on the period end date of the previous period.
    - For facility and item pairs with tight receipt control type, the receipt date must be before or on the receipt due date.
- To calculate the downstream service level value for facility and item pairs for the period under consideration:
    - If the dispatch quantity is greater than or equal to the customer order quantity, then the downstream service level value = 1
    - If the dispatch quantity is less than the customer order quantity, then the downstream service level value equals the ratio of the dispatch quantity and customer order quantity.
- The actual downstream service level with customer-specific service type value is calculated as follows:
    - If service type = fill rate (FR), then actual service level = downstream service level value as calculated in earlier point.
    - If service type = ready rate (RR), and if downstream service level value = 1, then actual service level = 1; else for any other value of downstream service level, the actual service level = 0.
    - If service type = backorder ratio (BR), then actual service level = 1 - (downstream service level value)

Target Downstream Service Level

The target downstream service level is available in the solution data layer.

Downstream Service Level Upper Control Limit

The downstream service level upper control limit is required for setting alerts on the user interface of the solution. This value (also called upper bound for service level) is customizable.

The derivation for the downstream service level upper control limit is as follows:

Downstream service level upper control limit = (Service level value supplied by the source system) + (Service level threshold specified in the global parameter list)

Downstream Service Level Lower Control Limit

The downstream service level lower control limit is required for setting alerts on the user interface of the solution. This value (also called lower bound for service level) is customizable.

The derivation for the downstream service level lower control limit is as follows:

Downstream service level lower control limit = (Service level value supplied by the source system) - (Service level threshold specified in the global parameter list)

Downstream Service Level Error High

If the downstream service level actual value is greater than the downstream service level upper control limit, then the downstream service level error high value is stored as 1, else 0.

Downstream Service Level Error Low

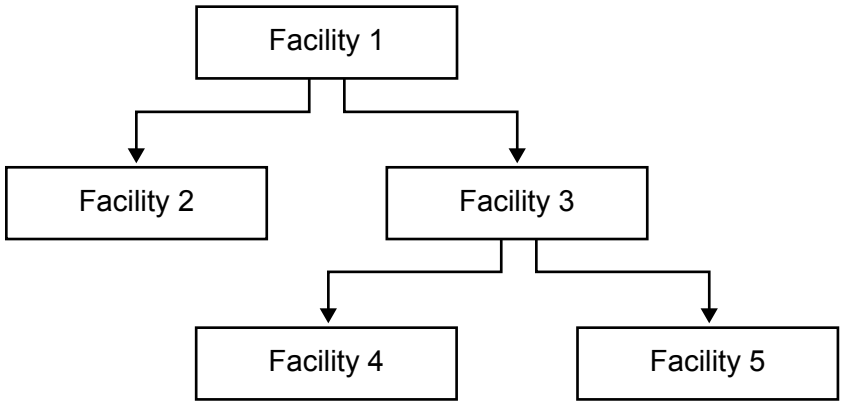
If the downstream service level actual value is less than the downstream service level lower control limit, then the downstream service level error low value is stored as 1, else 0.

Upstream Service Level

Overview

Service level is a measure of the fulfillment of customer demand. Service level value that is obtained from the supplying facility or external suppliers is stated as upstream service level. For example, consider the network in the following figure:

Figure A3.2 *A Sample Network*



The upstream service level for Facility 2 and Facility 3 is the service level that is obtained from Facility 1.

Actual Upstream Service Level

Actual upstream service level is the percentage service level that is calculated from the solution data layer for the facility. The average is calculated over the past period under consideration and is missing for the current and future periods.

The derivation for the actual upstream service level is as follows:

- Consider all regular orders for an item in a facility where the delivery due date exists between the start date and end date of the previous period. Receipt control type (possible values are LOOSE or TIGHT) is used to determine the period boundaries to calculate service level.
    - For facility and item pairs with loose receipt control type, the receipt date must be before or on the end date of the previous period.
    - For facility and item pairs with tight receipt control type, the receipt date must be before or on the receipt due date.
- The upstream service level value is calculated for facility and item pairs for the period under consideration as follows:

- If the received quantity is greater than or equal to the purchase order quantity, then the upstream service level value = 1.
- If the received quantity is less than the purchase order quantity, then the upstream service level value equals the ratio of the received quantity and purchase order quantity.

- The actual upstream service level with customer-specific service type value is calculated as follows:
  - If service type = fill rate (FR), then actual service level = upstream service level value as calculated in the earlier point.
  - If service type = ready rate (RR) and if upstream service level value = 1, then actual service level = 1; else for any other value of upstream service level, the actual service level = 0.
  - If service type = backorder ratio (BR), then actual service level = 1 - (upstream service level value)

Target Upstream Service Level

Target upstream service level is obtained from the solution data layer.

Upstream Service Level Upper Control Limit

The upstream service level upper control limit is required for setting alerts on the user interface of the solution. This value (also called upper bound for service level) is customizable.

The derivation for the upstream service level upper control limit is as follows:

Upstream service level upper control limit = (Service level value supplied by the source system) + (Service level threshold specified in the global parameter list)

Upstream Service Level Lower Control Limit

The upstream service level lower control limit is required for setting alerts on the user interface of the solution. This value (also called lower bound for service level) is customizable.

The derivation for the upstream service level lower control limit is as follows:

Upstream service level lower control limit = (Service level value supplied by the source system) - (Service level threshold specified in the global parameter list)

Upstream Service Level Error High

If the upstream service level actual value is greater than the upstream service level upper control limit, then the upstream service level error high value is stored as 1, else 0.

Upstream Service Level Error Low

If the upstream service level actual value is less than the upstream service level lower control limit, then the upstream service level error low value is stored as 1, else 0.

Demand

Overview

Demand of an item at a facility is the quantity of that item that is ordered at that facility. Demand is defined as orders for the amount of a particular service part that a consumer or facility wants to purchase at a given price. Demand for an item or a service is

determined by many different factors other than price, such as the price of substitute goods, complementary goods, and so on.

Actual Demand

Actual demand is the sum of the total order quantity that is received over the period under consideration for a facility and item pair. This demand value is calculated for all facility and item pairs from the source table FACILITY_ITEM_DEMAND. Demand values are not available for current and future periods.

The derivation for the actual demand is as follows:

Actual demand = Sum (Order quantity over the period under consideration)

Target Demand

Target demand is the sum of the total order quantity that is forecasted over the period under consideration. This demand value is calculated by using the output of the MIRP procedure, that is external demand mean and internal demand mean.

The derivation for the target demand is as follows:

Target demand = (External demand mean) + (Internal demand mean)

Demand Upper Limit

Demand upper limit (also called demand upper bound) is derived from the output of the MIRP procedure for the period under consideration. The upper bound value is calculated by using the customer-specific global parameter GL_DEMAND_THRESHOLD_COEFF. This global parameter specifies the coefficient value (a positive number < 3) that is used to calculate the lower and upper bounds for forecasted demand.

The derivation for the demand upper limit is as follows:

Demand upper limit =
(EXTERNAL_DEMAND_MEAN+INTERNAL_DEMAND_MEAN) +
(GL_DEMAND_THRESHOLD_COEFF. * (SQRT(EXTERNAL_DEMAND_VAR + INTERNAL_DEMAND_VAR)))

Demand Lower Limit

Demand lower limit (also called demand lower bound) is derived from the output of the MIRP procedure predict KPI for the period under consideration. The lower bound value is calculated by using the customer-specific global parameter GL_DEMAND_THRESHOLD_COEFF. This global parameter specifies the coefficient value (a positive number < 3) that is used to calculate the lower and upper bounds for forecasted demand.

The derivation for the demand lower limit is as follows:

Demand lower limit = (External demand mean + Internal demand mean) - (GL_DEMAND_THRESHOLD_COEFF * (SQRT(External demand variance + Internal demand variance)))

Demand Error High

If the actual demand value is greater than the demand upper control limit, then value for the demand error high is stored as 1, else 0.

Demand Error Low

If the actual demand value is less than the demand lower control limit, then value for the demand error low is stored as 1, else 0.

Inventory Cost

Overview

Inventory cost is defined as the cost of the closing inventory for the previous period. This value is the total aggregated cost of the inventory for all facility and item pairs.

Actual Inventory Cost

Actual inventory cost is an average cost that is calculated over the period under consideration.

The derivation for the actual inventory cost is as follows:

Actual inventory cost = (Average of closing inventory quantity in the period) * (Unit cost (for the previous periods) taken from item price)

Target Inventory Cost

Target inventory cost is the predicted or forecasted inventory cost for the current and future periods.

The derivation for the target inventory cost is as follows:

Target inventory cost = (On-hand mean * Unit cost) where on-hand mean is calculated by the MIRP procedure and the unit cost is used from the item price.

Inventory Quantity

Overview

Inventory units or quantity is defined as the average quantity of the closing inventory for the previous period.

Actual Inventory Quantity

The actual inventory quantity is the total aggregated quantity of inventory for all the facility and item pairs.

The derivation for the actual inventory quantity is as follows:

Actual inventory quantity = Average (Closing inventory quantity for a facility and item pair)

Inventory Turns

Overview

Inventory turns is the ratio of the total demand of items that are sold to the average inventory units of the items, over a given period.

Actual Inventory Turns

Actual inventory turns is calculated as the ratio of the actual demand for a period to the average inventory for that period.

The derivation for the actual inventory turns differs as per the selected base period. The derivation is as follows:

For daily base period

- If the GL_DAY_METRIC_LOAD parameter value is week (W), then inventory turns is calculated as follows:

  Inventory turns = ((Actual demand) * 52)/ (Actual inventory quantity)

- If the GL_DAY_METRIC_LOAD parameter value is month (M), then inventory turns is calculated as follows:

  Inventory turns = ((Actual demand) * 12) / (Actual inventory quantity)

For week base period

Inventory turns = ((Actual demand) * 52) / (Actual inventory quantity)

For month base period

Inventory turns = ((Actual demand) * 12) / (Actual inventory quantity)

For quarter base period

Inventory turns = ((Actual demand) * 4) / (Actual inventory quantity)

Calculations of Replenishment Plan Metrics

The replenishment plan metrics are displayed in the Order Suggestions workspace.

Here is a list of the metrics and their calculations:

Total Order Amount

Total order amount = Total amount of suggested orders from primary channel and alternative channels

Total Number of Orders

Total number of orders = Total number of orders from primary channel and alternative channels, where suggested order quantity > 0

Total Projected Cost

Total projected cost = Projected transfer cost + Projected holding cost + Projected penalty cost

Projected Holding Cost

Projected holding cost = (Projected on-hand after lateral transshipment + Projected future delivery) * Unit holding cost

Projected Transfer Cost

Projected transfer cost = Total transfer cost from primary channel + Total transfer cost from alternate channels

where, transfer cost = fixed ordering cost per order + [lead time in multiples of base period * pipeline cost per unit * order amount]

Projected Penalty Cost

Projected penalty cost = Inventory shortage * Unit penalty cost

Projected Service Level

Projected service level is based on the demand during projection intervals (DDPI).

DDPI = Lead time + Period between replenishments (PBR), if PBR > 1 and policy type is base stock

For other conditions, DDPI = Lead time + 1

If echelon = 1:
Projected service level = Average of the service level over the period DDPI, where the service level is not missing

If echelon > 1:
Projected service level = Average of the service level over the period DDPI excluding period 1, where the service level is not missing

Target Service Level
Target service level = Service level value as specified in the SDL

Service Level Upper Bound
Service level upper bound = Upper threshold value for the service level at a facility and item pair

Service Level Lower Bound
Service level lower bound = Lower threshold value for the service level at a facility and item pair

Calculations of the Scenario-Specific Metrics

The scenario-specific metrics are displayed in the Scenario Development workspace.

The following table lists the metrics, their calculations, and the scenario types where the metrics are used.

Table A3.1 *Scenario-Specific Metric Calculations*

Metric	Metric Calculation	Associated Scenario Types
Inventory units	Inventory units = Order-up-to level	• Customer-facing facility analysis • Internal facility service level analysis • Ad hoc analysis
Inventory Cost	Inventory cost = Order-up-to level * Item price	Customer-facing facility analysis
On-Hand Cost	On-hand cost = On-hand mean * Item price	• Internal facility service level analysis • Ad hoc analysis
On-Hand Holding Cost	On-hand holding cost = On-hand mean * Holding cost amount	• Internal facility service level analysis • Ad hoc analysis
Pipeline Cost	Pipeline cost = Pipeline mean * Pipeline cost amount	Ad hoc analysis

Metric	Metric Calculation	Associated Scenario Types
Total Cost	Total cost = On-hand cost + On-hand holding cost + Pipeline cost	• Service level sensitivity analysis • Internal facility service level analysis • Ad hoc analysis

Glossary

backorder penalty cost
the cost that is incurred when an item is out of stock. This cost might include the cost of emergency shipments, the cost of substituting a less profitable item, or the cost of lost goodwill.

base period
the interval of time in which one inventory replenishment order is allowed.

carrying cost
See holding cost

clustering
the process of dividing a data set into mutually exclusive groups such that the observations for each group are as close as possible to one another, and different groups are as far as possible from one another.

control table
a table containing parameter values that are used for the forecast and inventory optimization analyses. These values are customizable.

fill rate
a service measure that indicates the fraction of demand that is satisfied from on-hand inventory.

holding cost
the cost of keeping items in inventory, which includes the expense that is incurred in running a warehouse, handling inventory, and counting inventory. Holding costs might also include the cost of special storage requirements, deterioration of stock, damage, theft, obsolescence, insurance, taxes, or the opportunity cost of money invested. Also called carrying cost.

MAPE
See mean absolute percent error

mean absolute percent error
the average of the absolute percentage errors. Short form: MAPE.

multi-echelon network
the distribution network that has at least one facility and item pair with more than one echelon level. The echelon level of a facility and item pair represents its relative

position in a network. The echelon level of a pair is equal to the maximum echelon level of all its successor facility and item pairs plus one. If a facility and item pair does not have successors, its echelon level is one.

order-up-to level
the target inventory level.

penalty cost
See backorder penalty cost

planning horizon
the number of periods into the future for which predictions are made.

ready rate
the probability that the on-hand inventory level at the end of a review time period is positive.

reorder level
the inventory level at which a replenishment order should be placed.

RMSE
See root mean square error

root mean square error
the square root of the mean square error. It is used as an estimate of the standard deviation of the response variable. Short form: RMSE.

solution data layer
an intermediate layer of tables provided by the solution to save the customer source data in the required manner.

Index

R

S

T

U