THE POWER TO KNOW.

# SAS® Service Parts Optimization 4.31
## Administrator's Guide

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Insitute Inc. 2011. *SAS® Service Parts Optimization 4.31: Administrator's Guide*. Cary, NC: SAS Institute Inc.

**SAS® Service Parts Optimization 4.31: Administrator's Guide**



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

1st electronic book, August 2011

# Contents

# Recommended Reading

- *SAS Data Integration Studio: User's Guide*
- *SAS Forecast Server: Administrator's Guide*
- *SAS Management Console: User's Guide*

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

SAS Publishing Sales
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 Service Parts Optimization



## Welcome to SAS Service Parts Optimization

### *What is SAS Service Parts Optimization?*

SAS Service Parts Optimization enables you to forecast demand for service parts based on historical consumption data. You can plan inventory and orders to fulfill the

forecasted demand, and manage the introduction of new parts and new product launches into the service chain.

SAS Service Parts 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 forecasting and inventory optimization processes. 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 Service Parts Optimization*

SAS Service Parts 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 forecasting and inventory optimization processes.

  SAS Forecast Studio that is integrated with SAS Service Parts Optimization enables you to perform advanced analytical functions.

Java applications
: consists of the SAS Service Parts Optimization middle tier and client tier.

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

### *How Does SAS Service Parts Optimization Help You?*

SAS Service Parts Optimization provides advanced analytical capabilities for service organizations. Major uses of SAS Service Parts Optimization include the following:

- provide demand and inventory forecasts
- provide input to set inventory levels
- provide input to plan replenishment activities
- perform what-if analysis

SAS Service Parts Optimization provides automation and analytical sophistication to the forecasting and inventory optimization processes. You can generate millions of forecasts in a reasonable turnaround time. Also, you can calculate inventory replenishment policies for parts distribution systems and thereby maintain adequate stock levels. These capabilities enable you to reduce costs, increase revenues, and improve customer satisfaction and brand image.

## Using the Documentation

### *Purpose*

The Administrator's Guide describes the deployment, and the system and data administration tasks for the SAS Service Parts Optimization solution. This guide provides information about the following topics:

- system architecture
- installation and post-installation tasks
- data architecture and data flows
- initial and incremental data loading tasks
- system administration tasks
- troubleshooting instructions

## *Required Skill Sets*

To install, configure, administer, and use the SAS Service Parts Optimization solution, the following individuals with the necessary skill sets are required for each particular piece of administration and use:

System Administrator
: must have the skills to perform installation and configuration of the SAS Intelligence Platform and the SAS Service Parts Optimization solution on the required operating system. For example, to install the SAS Intelligence Platform on the Windows operating system, the administrator must have the following skills:

  - be an administrator of the computer
  - be familiar with concepts of the operating system
  - know how to create folders
  - know how to run DOS .bat files
  - be familiar with application server concepts

  The system administrator must use the SAS Management Console software to maintain the metadata for servers, users, and other global resources that are required by the solution.

Data Administrator
: must maintain the solution's data and perform other solution administration to enable users to analyze data. The administrator must be able to create ETL processes and programs that are used by the solution.

SAS Service Parts Optimization Solution User
: must understand the data to be analyzed, the requirements for analysis, and the results of data analyses.

## *Organization of this Guide*

The Administrator’s Guide provides an overview of the solution and its architecture, and explains the steps to set up and manage the solution. These steps can be categorized under the following high-level tasks:

- installation and post-installation
- data administration and management
- system administration

This guide is organized as follows:

Introduction
: introduces you to your SAS solution and explains how the guide is organized and presented. Also, provides additional resources for you to explore if you need more information about your SAS solution.

Installation and Post-Installation
: provides the steps to install the SAS Service Parts Optimization solution and also provides the post-installation tasks.

Data Administration and Management
: provides the data administering tasks that you must perform to load data for the first time and also covers the ongoing data tasks.

System Administration and Maintenance
: discusses the necessary system administration and maintenance tasks such as administering users.

Appendixes
: contains additional information that is referenced from within sections of the documentation.

# SAS Service Parts Optimization Architecture

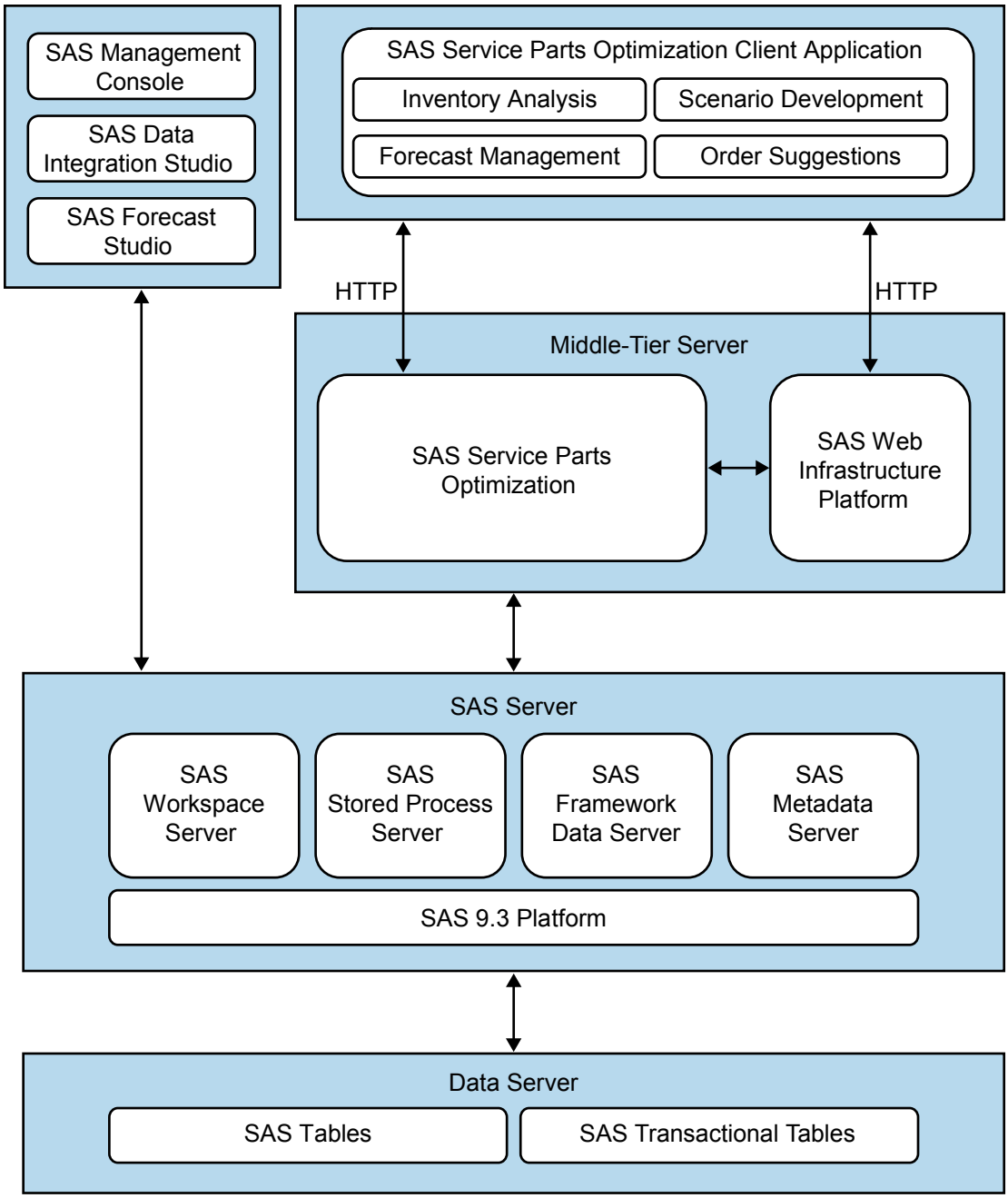
## *Overview of the Solution Architecture*

SAS Service Parts Optimization has an N-tier architecture that consists of the following tiers:

- client
- middle
- server
- data

The following figure shows the generic division of responsibilities across the different tiers.

***Figure 1.1*** *SAS Service Parts Optimization Architecture*




The subsequent sections explain the purpose and components of each tier.

## *Client Tier*

### *Overview*

The client tier is responsible for user interaction, navigation flows, preparation of a call for user choice, and displaying computational results.

### *Client-Tier Components*

SAS Service Parts Optimization requires the following applications:

SAS Management Console
: enables you to perform metadata administration activities.

SAS Data Integration Studio
: provides the ability to manage ETL jobs for the extraction, transformation, and loading of data.

The SAS Service Parts Optimization client displays computational results and enables user interaction. The client-side user interface is developed using the rich client platform (RCP), which is an eclipse plug-in-based development platform.

SAS Forecast Studio is bundled with SAS Service Parts Optimization to enable you to perform advanced analytical functions on the forecast results. For more information about SAS Forecast Studio, see *SAS Forecast Studio: User's Guide* and *SAS Forecast Studio: Administrator's Guide*.

## Middle Tier

### Overview

The middle-tier component is responsible for intercepting the client request and providing an execution environment to the client. The middle tier is developed over SAS business intelligence (BI) middle-tier technologies such as Web Infrastructure Platform (WIP) and Platform Services. These technologies provide the middle-tier application with the ability to run requests over SAS servers.

### Middle-Tier Components

SAS Service Parts Optimization requires the following middle-tier components:

SAS Service Parts Optimization Services
: take adequate server-side actions based on client-side user actions.

Web Infrastructure Platform (WIP) Services
: are used for standard operations such as configuration, authentication, e-mail, and so on.

  Middle-tier components use the HTTP-based protocol to communicate with the following WIP-based services:

  - Authentication Service
  - User Info Service
  - Mail Service

The client tier communicates with the middle tier by using HTTP-based communication model.

## Server Tier

### Overview

The server tier hosts SAS application servers that run the requests made by clients. Topological support might include support for non-functional requirements such as fail-over, pooling, and load balancing.

### Server-Tier Components

SAS Service Parts Optimization requires the following server-tier components:

SAS Workspace Server
uses SAS code execution for a number of clients at a time.

SAS Stored Process Server
runs a registered stored process on behalf of the client.

SAS Framework Data Server
accesses SAS transactional database and other Base SAS data sets for better performance.

SAS Metadata Server
sets up various configurations such as stored processes, workspace server, stored process server, framework data server DSNs, and so on.

## Data Tier

### Overview

Data tier consists of the following data storage types:

SAS tables
are stored in Base SAS formats under different libraries. For more information about the libraries, see Library Descriptions on page 18.

SAS transactional tables
are stored in the SAS Transactional Database. The tables contain transactional and application data.

## SAS Service Parts Optimization Modes of Operations

You can operate SAS Service Parts Optimization in two modes, online mode and batch mode.

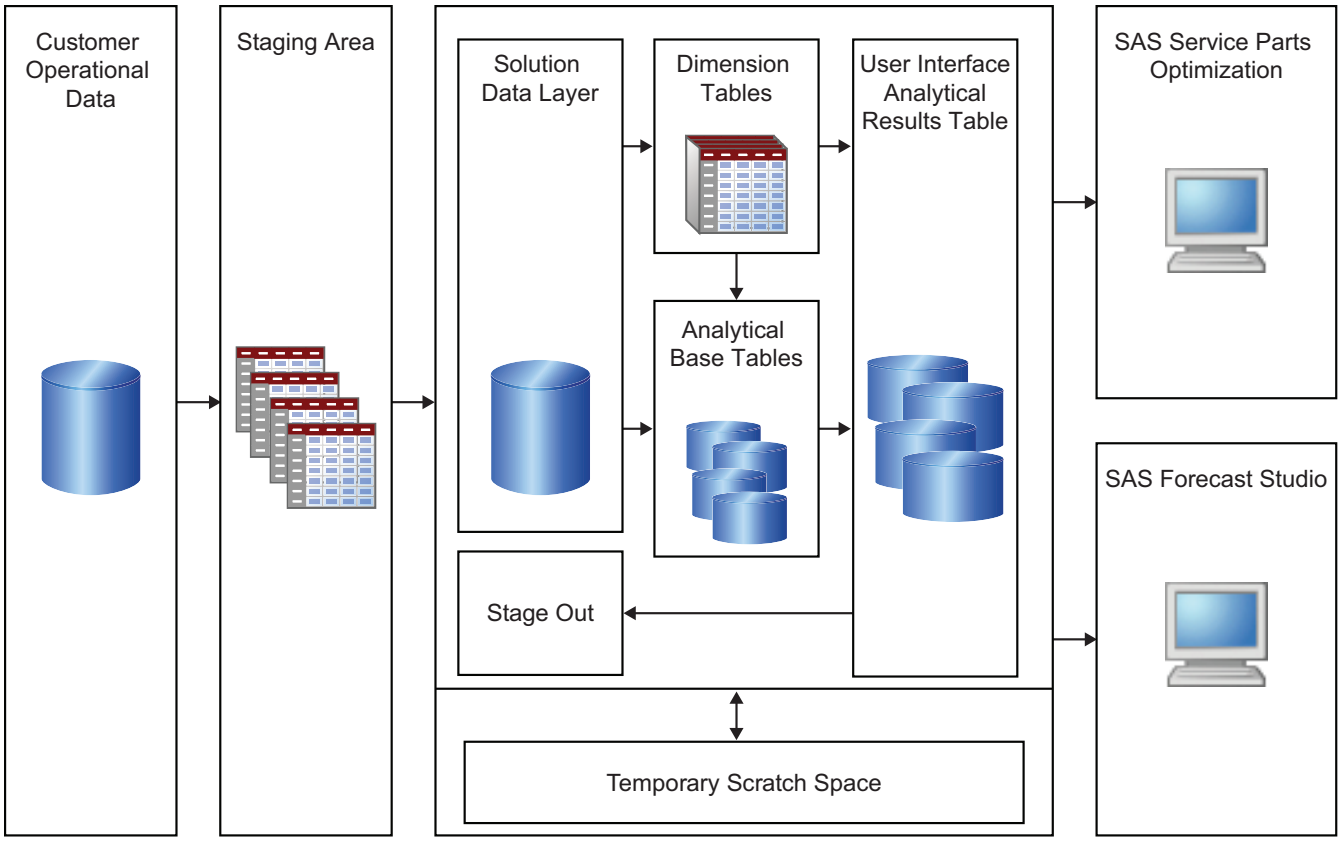
- In online mode, the middle tier, server tier, and data tier are running and you can use and work with the SAS Service Parts Optimization application.
- In batch mode, the ETL jobs to load, forecast, optimize, and report on data are running. The SAS Service Parts Optimization middle tier is shutdown and you cannot work with the client application. However, the SAS system uses ETL jobs to update and maintain the data.

  *Note:* A few ETL jobs update the UIART library. Ensure that the period of the job run is scheduled for a maintenance period that does not affect request processing.

## Back-End Data Flow for Loading and Accessing Data

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

***Figure 1.2*** *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 "Load All Solution Data Layer Tables " on page 19.

*Note:* All the earlier steps are site-specific. After the SDL is loaded, SAS Service Parts 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 Service Parts 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 forecasting and 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 Service Parts Optimization tables are loaded, the latest data is displayed on the interface of the SAS Service Parts Optimization application.

   You can refine a set of forecast results from SAS Service Parts Optimization by using the SAS Forecast Studio client. The results can be later viewed through the SAS Service Parts Optimization user interface.

   For more information about the data management tasks, see Chapter 3, "Data Administration and Management of SAS Service Parts Optimization," on page 17.

# SAS Service Parts Optimization Clients

## *Overview*

SAS Service Parts Optimization manages a separate session for each user who logs on. The options that are available on the user interface depend on the profile with which you log on.

SAS Forecast Studio is integrated with SAS Service Parts Optimization and enables you to perform advanced analytical processes on a set of forecast results. The outcome of the processes can be further viewed through the SAS Service Parts Optimization user interface. The following sections describe these client applications and their purpose.

Access to the clients is controlled through security settings that are set up for the users. For more information, see "Administer Users" on page 38.

## *SAS Service Parts Optimization Rich Client*

The SAS Service Parts Optimization rich client enables you to manage forecasts, analyze inventory, develop scenarios, and suggest orders.

The client is installed on a computer by running a client installation program. The client starts a session of its own. All tasks are performed under this session. The session ends when you exit the application or when the middle tier shuts down.

### SAS Forecast Studio Client

SAS Forecast Studio provides a graphical interface to the forecasting and time series analysis procedures. The SAS Forecast Studio client provides a user-friendly graphical interface to view forecasts. You can view forecasts that reflect the real scenarios of the business. You can also improve the forecasting performance across all products and locations, at any level of aggregation.

## Related SAS Software

### Overview

Many features that are not found in SAS Service Parts 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 Forecast Studio

SAS Forecast Studio is a client application that provides market-driven planning through accurate demand forecasting. Given a time-stamped data set, SAS Forecast Studio accumulates time-stamped data. SAS Forecast Studio then performs the following tasks:

1. forms a fixed-interval time series
2. diagnoses the time series
3. creates a list of candidate model specifications
4. fits each candidate model specification to your time series
5. generates forecasts
6. selects the most appropriate model specification
7. creates a forecast score
8. generates forecasts from the forecast score
9. evaluates the forecast

In short, SAS Forecast Studio automatically generates large volumes of reliable forecasts to support your planning and business decisions.

### SAS Inventory Optimization

SAS Inventory Optimization 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 Optimization 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 extract, transform, and load (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 Service Parts 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 Service Parts Optimization.

*Chapter 2*

# Installation and Post-Installation Tasks for SAS Service Parts Optimization



## Default File Locations

The default physical locations that are described in this guide are created during the standard configuration process. The SAS Consulting services representatives working at your site can change these locations when they customize your setup to meet your requirements. These representatives can then provide documentation of the locations for your site.

A standard file system location that is referred to in this guide is the SAS configuration directory (written as **<SASCONFIG>**). <SASCONFIG> refers to the directory path where the SAS configuration process writes the finalized software image.

Different files are written to this location. In a Windows operating environment, the default location for <SASCONFIG> is as follows: **C:\SAS\<ImageName>**, where **<ImageName>** represents the site-specific name for your software bundle as defined in the SAS plan file that is part of your order. For UNIX, this location can vary and your SAS Consulting services representative can help you to determine the default file location for your site.

## SAS Intelligence Platform Documentation

Because SAS Service Parts Optimization is designed to work with the SAS Intelligence Platform, it is strongly recommended that you read the SAS Intelligence Platform documentation, which is located at **http://support.sas.com/documentation/onlinedoc**, before you install SAS Service Parts Optimization. That documentation

provides pre-installation tasks and instructions to guide you through a typical installation of the SAS Intelligence Platform.

# Deployment Outline

To deploy SAS Service Parts Optimization:

1. Work with your SAS representative to develop a plan and designate the host computers for your deployment of SAS Service Parts Optimization.

   *Note:* While you create the software order, ensure that you include the following components in the order:

   - SAS Service Parts Optimization Server
   - SAS Data Integration Server
   - SAS Enterprise Business Intelligence Server

2. Install SAS Service Parts Optimization on each host computer as outlined in your plan.xml file. During installation, ensure the following:

   - Specify the e-mail address that can be used by the SAS Service Parts Optimization application to send e-mails.
   - SAS Service Parts Optimization environment can connect to any SAS environment where the SAS Service Parts Optimization middle tier is deployed. SAS environments are defined in the sas-environment.xml file.

3. Complete the post-installation tasks. For more information, see "Post-Installation Tasks" on page 14.

# Post-Installation Tasks

## *Update the Batch File*

To run the ETL jobs in a batch by using the SAS Batch Server, update the batch file by removing the **noxcmd** option.

To do so:

1. Browse to the SAS Batch Server folder. By default, the path is **<SASCONFIG>\Lev1\SASApp\BatchServer**.
2. Edit the batch file.

   *On a Windows Operating Environment*

   a. Open the sasbatch.bat file in edit mode.

   b. In the file, search for the text **–noxcmd** and comment out this text.

```
rem CMD_OPTIONS="-noxcmd -lrecl 32767"
```

   *On a UNIX Operating Environment*

   a. Open the sasbatch.sh file in edit mode.

b. In the file, search for the text **–noxcmd** and comment out this text.

```
rem CMD_OPTIONS="-noxcmd -lrecl 32767"
```

For example, consider the following text:

```
# CMD_OPTIONS="-noxcmd -lrecl 32767"
```

### *Update the SAS Scripts*

On a UNIX operating environment, you need to update certain SAS script files to ensure that the tables that are created by the SAS Workspace Server and the SAS Stored Process Server have the necessary write permissions.

Specify the umask setting of 002 in the following SAS scripts:

- **<SASCONFIG>**/Lev**<N>**/SASApp/BatchServer/sasbatch_usermods.sh
- **<SASCONFIG>**/Lev**<N>**/SASApp/PooledWorkspaceServer/PooledWorkspaceServer_usermods.sh
- **<SASCONFIG>**/Lev**<N>**/SASApp/StoredProcessServer/StoredProcessServer_usermods.sh
- **<SASCONFIG>**/Lev**<N>**/SASApp/WorkspaceServer/WorkspaceServer_usermods.sh

### *About fds.parms File*

The fds.parms file stores the user name and password for the framework server database. For the ETL jobs to run successfully, the file is mandatory. During installation, the file is automatically copied to the location <SAS Config>\Lev<N>\AppData\SASServicePartsOptimization.

*Note:* Make sure you don't change the content of the fds.parms file.

### *Create an Environment to Work with Stored Processes in SAS Forecast Studio*

The Forecast Management workspace enables users to select time series to be reforecasted in SAS Forecast Studio. Two stored processes are provided to copy the revised forecast results from SAS Forecast Studio to the SAS Service Parts Optimization tables. You must create an environment that enables users to work with the stored processes in SAS Forecast Studio. This is a one-time activity.

To create the environment to work with stored processes in SAS Forecast Studio:

1. Open SAS Management Console.
2. In the **Application Management** folder, click **Forecast Server**. The SAS Forecast Server Plug-in window appears.
3. Log on to SAS Forecast Server by using the appropriate credentials.
4. In the SAS Management Console, right-click **Forecast Server** and from the pop-up menu, select **New Environment**. The Create Environment window appears.
5. Type an appropriate name and description for the environment.
6. Ensure that the **Host name** box displays **SASApp - Logical Workspace Server**.

7. In the **Location** box, type the path to the environment that you create.

   For a Windows operating environment, the default path is **<SASCONFIG>**/Lev**<N>**/AppData/ForecastServer/4.1.

   For a UNIX operating environment, the default path is **<SASCONFIG>**\Lev**<N>**\AppData\ForecastServer\4.1.

   *Note:* This location contains the environment that you create. Ensure that no other environment exists in the specified location.

8. In the **Reports folder** box, type **/Products/SAS Service Parts Optimization/Service Parts Optimization 4.3/STORED_PROCESSES** and click **OK**. The new environment is created and displayed in the right pane.

   *Note:* The **Reports folder** box must contain the path to the folder where the stored process is stored.

*Chapter 3*

# Data Administration and Management of SAS Service Parts Optimization



## Overview

After you complete the post-installation tasks, the SAS Service Parts 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.
4. Administer users.

For more information about the general data flow, see "Back-End Data Flow for Loading and Accessing Data" on page 7.

# Selecting the Base Period

The base period for the SAS Service Parts Optimization solution must match the planning frequency of your organization. You can specify the forecasting base period as week, month, or quarter. For inventory optimization, you can also specify day as the base period.

# Pre-Initial Data Load Tasks

## *Overview*

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

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

## *Confirm That Libraries Are Created*

After you complete the installation of the SAS Service Parts Optimization solution, ensure that all the libraries are created. The default path for the libraries is as follows:**`<SASCONFIG>/Lev1/AppData/SASServicePartsOptimization/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.

Forecast
: stores tables that contain time series details for facility and item pairs with low accuracy forecast results. A table is created per forecast group. It also stores tables that contain the time series that are selected from the user interface to be revised in SAS Forecast Studio.

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. You must load these tables with customer-specific data.

SPO_ER
: stores tables with information about user-defined events that are specific to each forecast group.

SPO_MR
: stores user-defined custom models of a catalog.

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 spodb 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 Service Parts 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\spartoptmva\sasmisc`**. If you install SAS in the default folder, then the script is available at **`C:\Program Files\SAS\SASFoundation \9.2\spartoptmva\sasmisc`**.

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

### *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 3.1*** *Loading Sequence of Solution Data Layer Tables*

| Group Number | Table Name |
|---|---|
| 1 | • Forecast_Group<br>*Note:* Ensure that each forecast group contains at least one time series.<br>• Forecast_Subgroup<br>• Item_Category<br>• Location<br>• Lookup_Master<br>• Network_Model<br>• Route_Type_Ref<br>• Time_Period |
| 2 | • Employee<br>• Item<br>• Lookup_Detail<br>• Organization<br>• Time_Period_Assoc |
| 3 | • BOM<br>• Customer<br>• Facility<br>• Item_Price<br>• Item_Substitute<br>• Item_Succession<br>• Vendor |
| 4 | • Customer_Order<br>• Facility_X_Item<br>• Purchase_Order<br>• Route<br>• Vendor_Facility_Item |

| Group Number | Table Name |
|---|---|
| 5 | • Back_Order_Summary<br>• Dispatch<br>• Facility_Item_Demand<br>• Facility_Item_Ind_Variable<br>• Facility_Item_Inventory<br>• Facility_Item_X_Network<br>• Forecast_Group_Item_Detail<br>• Network_X_Route<br>• Pipeline_Inventory<br>• Receipts<br>• 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 3.2*** *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 3.3*** *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 3.4*** *Table and Its Mandatory Code Column Names*

| Table Name | Code Column Name |
|---|---|
| Facility | Facility_Type_CD |
| Facility_X_Item | • Policy_Type_CD<br>• Service_Type_CD |
| Forecast_Group | Forecast_Group_CD |
| Item | • Item_Group_CD<br>• Item_Status_CD<br>• Item_Type_CD<br>• 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. You can access these jobs from the following location in SAS Data Integration Studio:

**`/Products/SAS Service Parts Optimization/Service Parts Optimization 4.3/CONTROL/INITIAL_ONE_TIME_JOBS`**

Run the jobs in the following sequence:

1. 01_CREATE_GLOBAL_PARAMETER_LIST
2. 02_CREATE_CONTROL_TABLES
3. 03_CREATE_FCST_ABT
4. 04_CREATE_TIME_PERIOD_DATA
5. 05_CALENDAR_HIERARCHY1
6. 06_CALENDAR_HIERARCHY2
7. 07_TIME_DIM
8. 08_CREATE_TIME_VIEWS
9. 09_CREATE_USER_TABLES
10. 10_POPULATE_HPF_PREFERENCES
11. 11_POPULATE_HPF_PREFERENCE_OPTIONS_VALUE
12. 12_POPULATE_HPF_PREFERENCE_FG_TABLE
13. 13_CREATE_EVENTS_TEMPLATE

14. 14_CREATE_TIME_GRAIN_FG

For more information about each job, see Appendix 2, "ETL Job Details," on page 63.

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 Service Parts 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 Framework Data Server 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 Service Parts 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 61. With the time dimension table, views for week, month, quarter, and year are also created, which are required for running the ETL jobs.
- A control table is populated with all the combinations of HPF parameter values that are used to display information in the user interface.
- All events that are used by the forecasting batch process are updated in a table.

All these tasks ensure that the configuration of the overall environment for SAS Service Parts 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 49.

## *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.

### *Update the Control.FCST_Offdays_Peakdays_List Table*

Perform this task only if you choose day as the base period.

To run on a daily basis, the inventory optimization process needs daily forecasted demand data. This daily data cannot be generated. You must update the Control.FCST_Offdays_Peakdays_List table that enables the internal processes to divide the weekly forecast values into daily forecast values to be used by the inventory optimization process.

For every facility and item pair, you must specify the non-working days (off days) in a week, days with peak demand (peak days), and percentage based on the actual total demand of the week for the peak days in the control table. The control table includes the following columns of information:

- Facility_Rk: This column contains the retained key for a facility. If the value is 0, then it indicates that the values in the other columns are default values and are applicable for all facilities.
- Item_Rk: This column contains the retained key for an item. If the value is 0, then it indicates that the values in the other columns are default values and are applicable for all items.
- Offday1, Offday2: These two columns specify holidays or off days within a week. The values in the columns could be any number between 1 and 7 where 1 denotes Sunday, 2 denotes Monday, and so on. For example, Off day1 = 3 signifies that Tuesday is an off day.
- Peakday1, Peakday2: These two columns specify days that have peak demand. The values in the columns could be any number between 1 and 7, where 1 denotes Sunday, 2 denotes Monday, and so on.
- Peak day1_PCT, Peakday2_PCT: These two columns specify the percentage values of the total demand of the week on the peak days.

Consider some examples based on the following sample control table.

***Table 3.5*** *Sample Control Table*

| Facility_rk | Item_rk | Off day1 | Off day2 | Peak day1 | Peak day 1_pct | Peak day2 | Peak day2_pct |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 7 | | | | |
| 1 | 0 | 6 | | | | | |
| 2 | 1 | 1 | 7 | 2 | 25 | 3 | 21 |

Example 1: As per the values in the last row of the table, the off days are 1 and 7, which means that Sunday and Saturday are the non-working days. Peak demand occurs on day 2, which is Monday, and it accounts for 25% of the total demand of the week. The other peak demand occurs on day 3, which is Tuesday, and is around 21% of week.

Example 2: For the combination of Facility_rk = 2 and Item_rk = 2, the first row from the table is considered. Off days are 1 and 7, Sunday and Saturday, and the other values are not specified. Hence, the demand is split equally among the other five days.

### *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 61.

## Ongoing Data Tasks

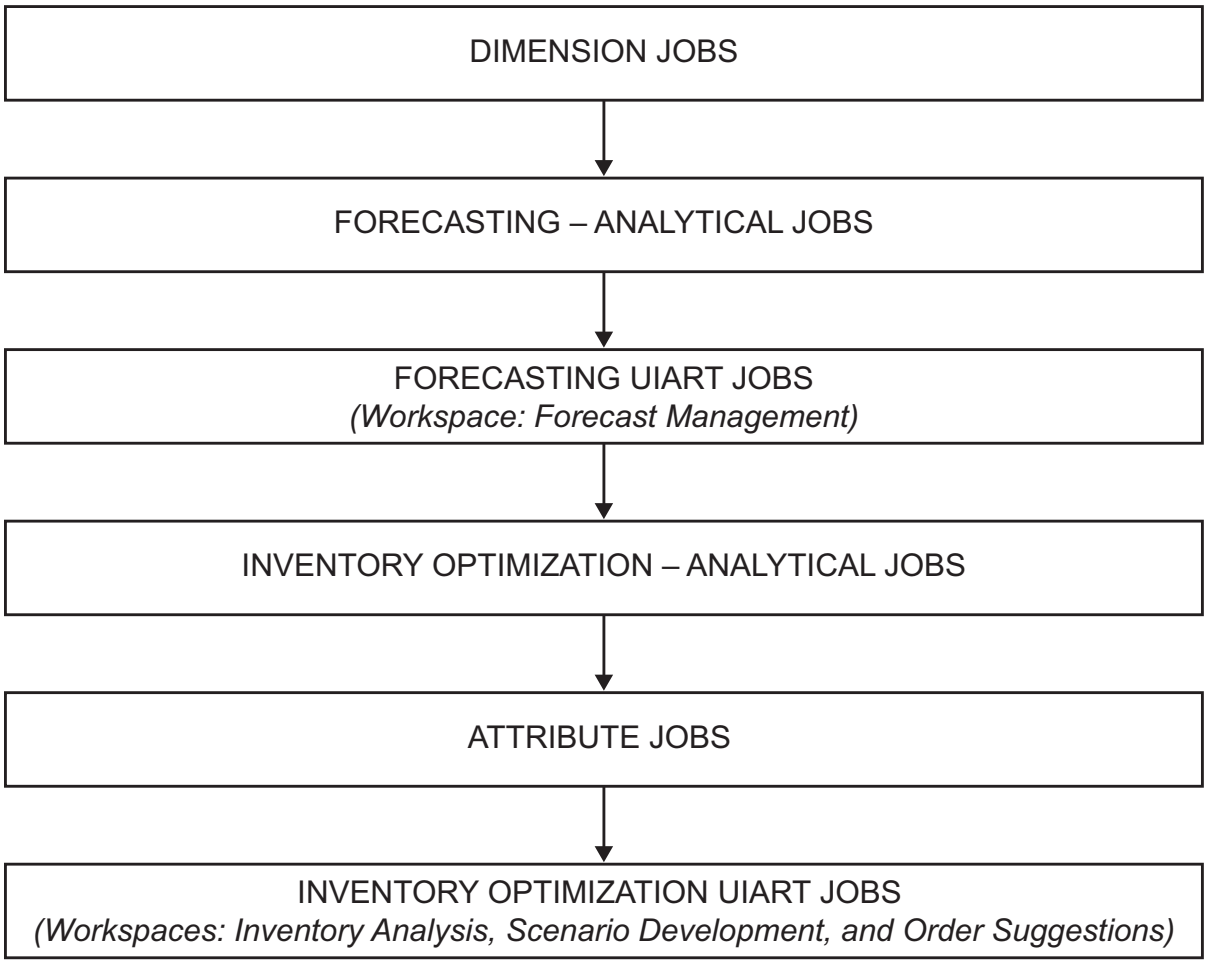
### *Overview*

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 Service Parts Optimization jobs.

***Figure 3.1*** *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 63.

### *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.

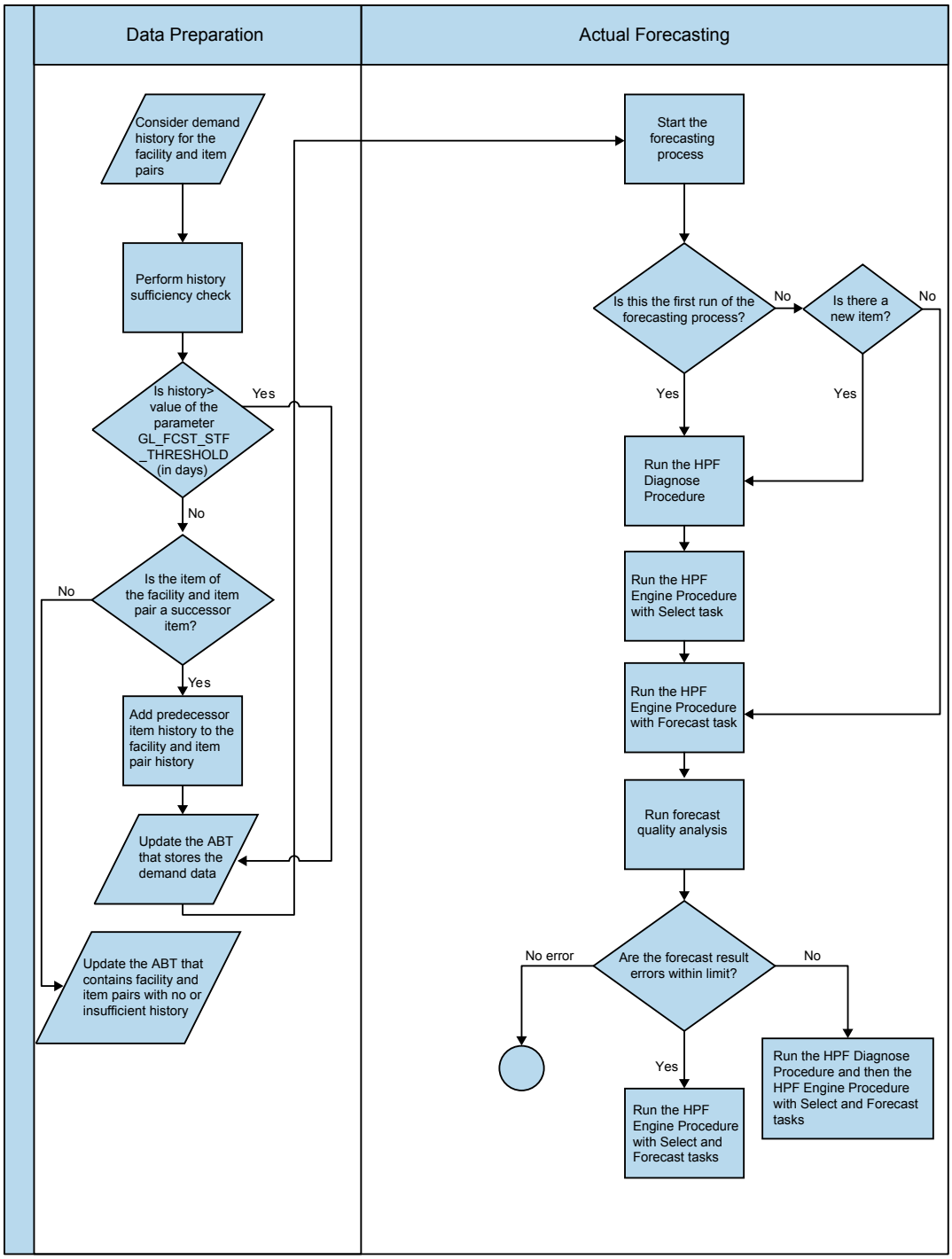
## *Forecasting: Analytical Jobs*

### *Overview of Forecasting*

The forecasting process in SAS Service Parts Optimization enables you to decide the future demand levels at user-defined hierarchies. The forecasting module provides forecast of the demand quantity over the planning horizon with historical demand history as input. The forecasting process can be run either in a batch mode or through ETL jobs.

The following figure explains the forecasting process flow for SAS Service Parts Optimization.

***Figure 3.2*** *The Forecasting Process*




The forecasting results are divided into different categories: normal forecast results, revisited forecast results, and low accuracy forecast results, depending on their accuracy. The forecasting process identifies demand quantity for successor items and displays the forecast results on the user interface of the solution. Results for time series with insufficient or no history, and for time series with intermittent demand are also displayed on the user interface.

### *Forecasting Batch Process Jobs*

The forecasting batch process jobs are categorized as follows:

- initial and subsequent forecasting jobs: The initial and subsequent jobs populate the analytical result tables and include the new-part and short-term forecasting jobs. If the SDL stores historical data for less than the number of days that you have predefined in the global parameter GL_FCST_STF_THRESHOLD, then the new-part forecasting jobs are run. If the historical data is for the predefined number of days or more, then the short-term forecasting jobs run.
- long-term forecasting jobs: If historical data for at least 40% of the planning horizon is available, the long-term forecasting jobs can run.
- jobs to populate user interface analytical result tables: These tables are used for displaying information on the user interface of the SAS Service Parts Optimization application.

The quality analysis jobs of the forecasting batch process classify all the time series into normal, revisited, and low accuracy forecast results and take subsequent corrective measures. The table Control.FCST_QA_Tests stores the standard tests that can be run for quality analysis. You can edit this table and select the tests to be performed.

If you choose to specify your own quality analysis process, you need to set the value of the GL_FCST_RUN_QA_MACRO_IND global parameter as 0. Then, populate the **<Base_Period>**_Forecast_Control_Data**<Count>** tables of the Control library for each run of the forecasting batch process. Here, **<Base_Period>** should be wk, mth, or qtr depending on the base period, and **<Count>** is a unique serial number that is assigned to the forecast group within the base period. For details about the tables, see *SAS Service Parts Optimization: Data Dictionary*.

Depending on your source data, you can choose to optimize the forecasting batch process by specifying values in the following:

- GL_SPO_OPTIMIZATION_FLAG and GL_SPO_USE_SUBGROUP_FLAG global parameters
- Forecast_Groups_Opt_Flag column of the Forecast_Groups_Opt_Flag table

*Note:* If you choose to replace missing values in the actual demand data with *missing*, then these missing values are not included in the quality analysis process.

To run the forecasting batch process:

1. Run the initial forecasting jobs.
    a. Run the demand data accumulation jobs to populate ABTs for short-term forecasting and new-part forecasting. The combined demand data ABT is used as an input by the high-performance forecasting (HPF) engine to generate forecasting results. All forecast results of different hierarchy levels are stored in separate folders.
    b. The main forecasting job diagnoses possible forecast models based on the available demand history, fits, and selects the best model as per the specified criteria.
    c. Monthly, quarterly, and weekly results are combined into a single result table and forecast results history is maintained.
2. Run subsequent forecasting jobs to incorporate any newly introduced time series and to rerun the forecasting process.
    a. Run the incremental demand data accumulation jobs to populate incremental ABTs for short-term forecasting and new-part forecasting.

b. The jobs handle any newly introduced time series and newly added forecast groups.

c. The jobs classify forecast results into normal, revisited, low accuracy, successor items, items with intermittent demand, and items with no history, and perform subsequent corrective measures on the results.

d. Prediagnosed and preselected models are used for forecasting. The quality analysis jobs classify all the time series into normal, revisited, and low accuracy forecast results and take subsequent corrective measures.

e. Monthly, quarterly, and weekly results are combined into result tables for every forecast group for the chosen base period and forecast results history is maintained.

3. Run long-term forecasting jobs. Demand quantities of facility and item pairs to be used for forecasting are aggregated at a yearly level. The long-term forecasting results are obtained based on Regression and Bass model methods.

*Note:* Some time series might not be forecasted due to incorrect parameter settings or lack of sufficient history. You can view these unforecasted time series in the Forecast library. The table names have the following naming convention: **<Base_Period>**_unforecasted_timeseries**<Count>**. Here, **<Base_Period>** should be wk, mth, or qtr depending on the base period and **<Count>** is a unique serial number that is assigned to the forecast group within the base period (for example, wk_unforecasted_timeseries2).

For more information about each job, see Appendix 2, "ETL Job Details," on page 63.

### Forecasting: UIART Jobs

The forecasting UIART jobs update tables that are used to display forecasting data on the user interface of the solution.

The tables in the UIART library contain the overall time series that can be divided into normal, revisited, and low accuracy forecast results. The tables also store time series with insufficient history, intermittent demand, and time series with successor items.

All the actual and forecasted values for the time series provide important statistical parameters, such as root mean square error (RMSE), mean absolute percent error (MAPE), standard deviation (STD), forecasted lower limit, forecasted upper limit, and demand error.

While all the tables are stored in the UIART library, the No_History_Timeseries_Detail, Forecast_Result, and HPF_Preferences_UI tables are stored in the TSDB library for concurrent access of data.

The external demand data that is specified through the user interface is stored in a table. Also, default values of various options or parameters that are required for running the forecasting process are stored in a control table. Some of these parameters can be updated from the user interface of the solution. You can modify all parameters directly in the back-end table. The modified parameters and the external demand data are used as input and forecasting process is run again.

*Note:* Specifying REG as the identification order for the ARIMAX model can produce unpredictable results including forecast failures.

## *Inventory Optimization: Analytical Jobs*

### *Overview of Inventory Optimization*

The inventory optimization back-end process in SAS Service Parts 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?

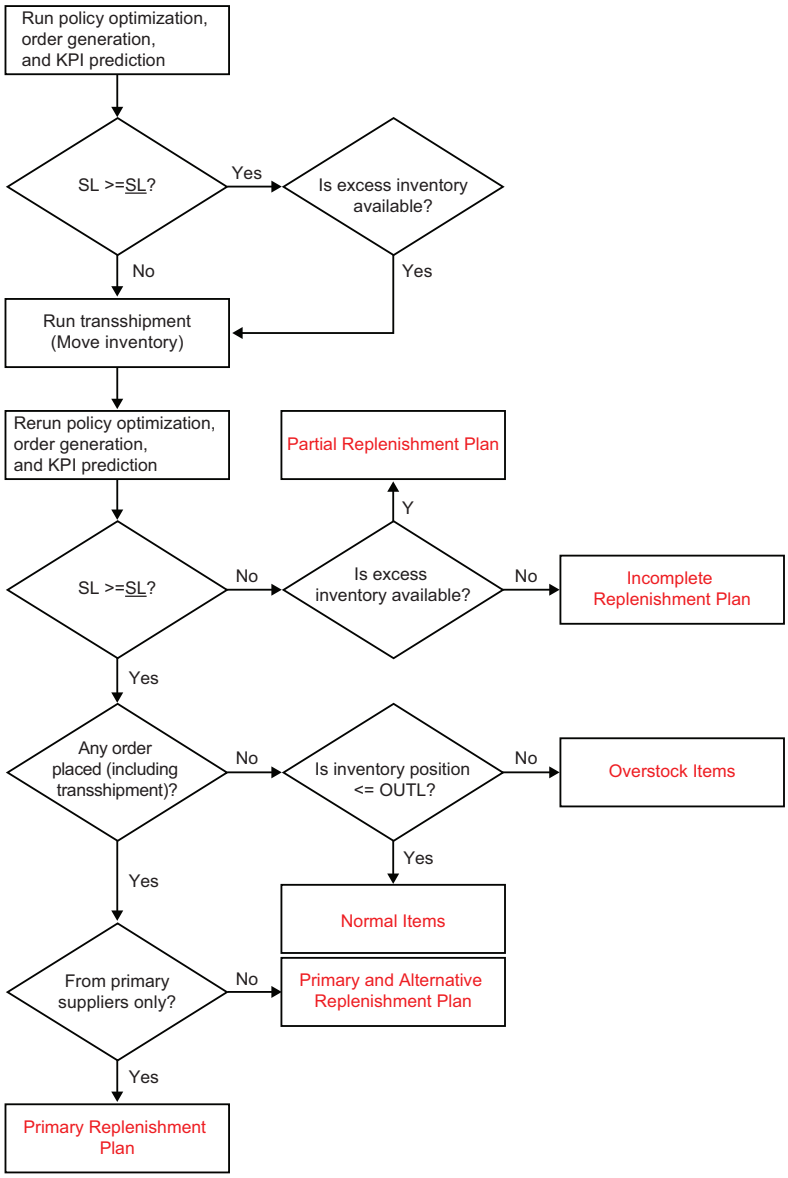
The inventory optimization process handles the following conditions:

- 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 Service Parts Optimization.

***Figure 3.3*** *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 on the user interface of the solution

The inventory optimization batch process requires forecasted values of the customer demand and uses the following information from the SDL:

- 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 on the user interface of the solution.

For more information about each job, see Appendix 2, “ETL Job Details,” on page 63.

### *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 on 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 Service Parts Optimization: Data Dictionary*.

For more information about each attribute job, see Appendix 2, "ETL Job Details," on page 63.

## 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 on 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 Service Parts 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 107.

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.

You can perform ad hoc analysis on the source data and also on the demand data from a SAS Forecast Studio project.

For more information about calculations of the scenario-specific metrics, see "Calculations of the Scenario-Specific Metrics" on page 115.

### *Order Suggestions Workspace*

SAS Service Parts 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.

# Administer Users

Each SAS Service Parts Optimization user is assigned a role that determines the functions that the user can perform in the SAS Service Parts Optimization application.

The following table provides descriptions for the different user roles in SAS Service Parts Optimization.

***Table 3.6*** *SAS Service Parts Optimization Roles and Descriptions*

| Role | Display Name | Description |
|---|---|---|
| Buyer | SPO: Buyer | Performs all tasks within the Order Suggestions workspace. |
| Forecaster | SPO: Forecaster | Performs all tasks within the Forecast Management workspace, except editing the forecasting process parameters. |
| Advance Forecaster | SPO: Advance Forecaster | Performs all tasks within the Forecast Management workspace. |
| Inventory Analyst | SPO: Inventory Analyst | Performs all tasks within the Inventory Analysis and Scenario Development workspaces, except promoting scenarios. |

| Role | Display Name | Description |
|---|---|---|
| Advance Inventory Analyst | SPO:Advance Inventory Analyst | Performs all tasks within the Inventory Analysis and Scenario Development workspaces. |

One of these roles must be assigned to every SAS Service Parts Optimization user.

After you create users for SAS Service Parts Optimization, you must map their IDs (that you create in the SAS Management Console) with their corresponding employee IDs.

To map a user ID with its corresponding employee ID:

1. Browse to the folder that you defined for the UIART library. For Windows operating environment, the default path to this folder is as follows: **`<SASCONFIG>\Lev<N>\AppData\SASServicePartsOptimization\data\uiart`**
2. Open the **`user_mapping.sas7bdat`** file with SAS 9.2. The SAS window appears and displays a table with the following column names:
   - SMCID
   - SPOID
   - USER_ID
   - BUYER_IND
   - EMAILID
3. For the user that you want to map, in the SMCID column, type the user ID that you specified while creating the user.
4. In the SPOID and USER_ID columns, type the user's business key and unique identifier values respectively. You can get these values from the EMPLOYEE_ID and EMPLOYEE_RK columns of the SDL.Employee table.
5. In the BUYER_IND column, if the user is a buyer, type *1*, else type *0* for a non-buyer user.
6. In the EMAILID column, type the user's e-mail ID.
7. Save all changes.

*Chapter 4*

# Administration and Maintenance of SAS Service Parts Optimization



## System Administration and Maintenance

### *Administrative Tools*

The system or solution administrator might use the following administrative tools:

- SAS Management Console. You can use the following plug-ins of SAS Management Console to manage metadata for the SAS Metadata Server:
  - Server Manager
  - Data Library Manager
  - User Manager
  - BI Manager
  - Authorization Manager
- SAS Data Integration Studio. You can use the SAS Data Integration Studio to manage ETL and other jobs, which are sequences of steps for the extraction, transformation, and loading of data.

For more information about the administrative tools, refer to each product's online Help and related documentation.

## Servers for Solution Use

The following table shows the servers that are used by the SAS Service Parts Optimization components.

***Table 4.1*** *Servers for Solution Components*

| Server | Dependent Components |
|---|---|
| SAS Metadata Server | All SAS components |
| SAS Object Spawner (SAS Workspace Server, Pooled SAS Workspace Server, and SAS Stored Process Servers) | Batch jobs and middle tier |
| SAS Service Parts Optimization Middle Tier | Client tier |
| FrameworkData Server | Middle tier |
| WIP Middle Tier | Middle tier and client tier |
| Remote Services | Middle tier |
| Application Server (JBoss, WLS, or WAS) | Client tier |

## Maintenance Activities

To maintain the architecture, back-end data, and solution client, you must perform the maintenance activities listed in the following table.

***Table 4.2*** *Maintenance Activities for SAS Service Parts Optimization*

| Maintenance Activity | Comments |
|---|---|
| Stop and start the middle-tier server as required for the incremental data flow updates and nightly jobs. | The incremental data update does not cause any problem to the middle tier and client tier. However, if some client task is being performed when data is updated in the UIART tables, then the client task might fail. To avoid failure of the client task, you can stop the middle tier until the data load completes. |
| Maintain a log file for the SAS Service Parts Optimization middle-tier server. | Log files must be archived time to time. The middle-tier application server creates log files that are sorted by date. Log files for workspace server and stored process server are located in the `log` folder. |
| Install updates to software. | None. |
| Back up and restore data, as required by your organization. | Site-specific. |

| Maintenance Activity | Comments |
|---|---|
| Back up the SAS Metadata Server. | For details, see documentation for SAS Intelligence Platform. |

***CAUTION:***

**After you create new users for SAS Service Parts Optimization, you must map their SAS Management Console IDs with the corresponding employee IDs.** For more information, see "Administer Users" on page 38.

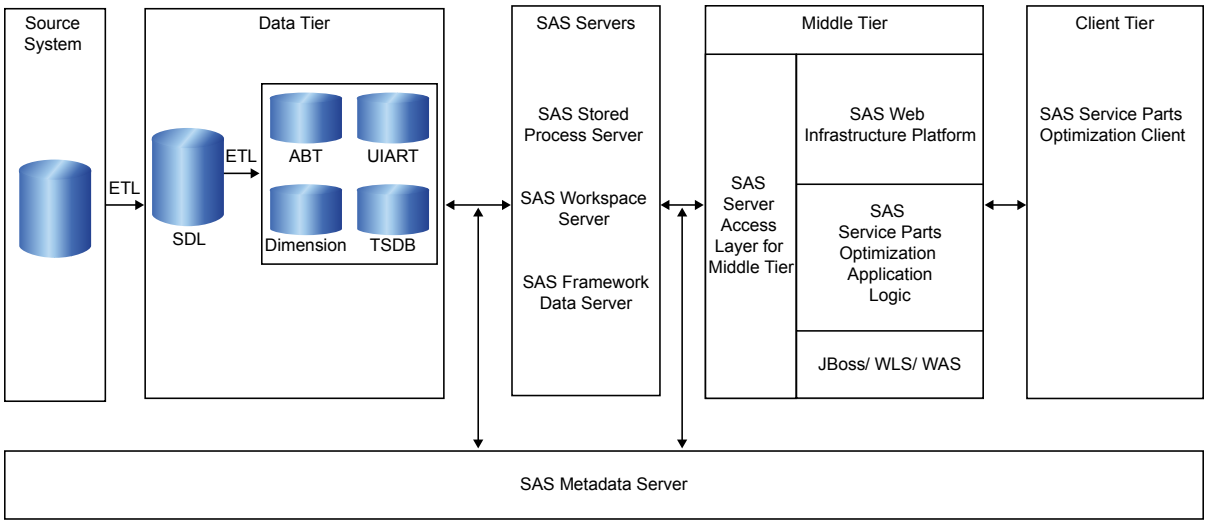
*Note:* If you want to change the name of the SAS Workspace Server to a language other than English, ensure that the spoDatabase.properties file is encoded in UTF-8 format.

# Tier Dependencies

SAS Service Parts Optimization has three functional tiers:

- server tier
- middle tier
- client tier

***Figure 4.1*** *Tier Architecture of SAS Service Parts Optimization*




Legend

| | |
|---|---|
| ETL | Extract, Transform, Load |
| SDL | Solution Data Layer |
| ABT | Analytical Base Table |
| UIART | User Interface Analytical Result Table |
| TSDB | Framework Server Database |
| WLS | WebLogic Server |
| WAS | Websphere Application Server |

Servers and services in a tier might depend on servers on other tiers. The dependency among the tiers is provided in the following tables.

***Table 4.3*** *Tier Dependency Summary*

| Tier | Dependent On |
|---|---|
| Middle tier | Server tier |
| Client tier | Middle tier |

***Table 4.4*** *Tier Dependency Details*

| Server or Service | Dependent On |
|---|---|
| Middle tier | |
| SAS Service Parts Optimization middle tier | 1. Framework data server<br>2. Object spawner<br>3. Remote services<br>4. SAS Web Infrastructure Platform (WIP) services |
| Client Tier | |
| SAS Service Parts Optimization Client | 1. SAS Service Parts Optimization middle tier<br>2. SAS WIP services |

*Note:* When you start any of the dependent server or service, ensure that you start the dependent server or service in the order that is specified in the earlier table.

# Hide or Display Workspaces in the Client Application

You can modify settings in the SAS Management Console to hide or display workspaces in the SAS Service Parts Optimization application. By default, all the workspaces are displayed. The Forecast Management workspace is represented by one property and the Inventory Analysis, Scenario Development, and Order Suggestions workspaces are represented by another property in the SAS Management Console.

To hide or display a workspace in the application:

1. Open SAS Management Console.
2. In **Application Management**, expand **Configuration Manager**.
3. Right-click **Serv Part Opt 4.3** and from the pop-up menu, select **Properties**. The Serv Part Opt 4.3 Properties dialog box appears.

4. Click the **Advanced** tab. The following workspace-related properties are displayed in the Property Name column:
   - Spo.DemandPlanningVisible
   - Spo.InventoryOptimizationVisible
5. To hide a workspace, in the related Property Value box, type *No*.
6. To display a workspace, in the related Property Value box, type *Yes*.

   *Note:* If you specify *Yes* for the Spo.InventoryOptimizationVisible property, then Inventory Analysis, Scenario Development, and Order Suggestions workspaces are chosen to be displayed.

   *Note:* Ensure that you choose to display at least one of the workspaces.
7. Restart the SAS Metadata Server and the SAS WebApp Server. For details, see documentation for SAS Intelligence Platform.

# Migrate the SAS Service Parts Optimization Database

By using the SAS Deployment Wizard, you can perform the main migration of the SAS Service Parts Optimization solution to another 9.3 system. However, you need to manually migrate the SAS Service Parts Optimization database.

To migrate the database:

1. In the computer where you are migrating the solution, stop **SAS Framework Data Server** and **JBoss** services.
2. In the computer from where you are migrating the solution, stop **SAS Framework Data Server** and **JBoss** services.
3. In the computer from where you are migrating the solution, browse to the configuration directory of the framework database server and copy the SPODB.FDB file. The default location is **`<SAS Config>\lev<N>\FrameworkServer\Content`**.
4. In the configuration directory of the framework database server of the target computer, paste the copied SPODB.FDB file.
5. In the computer where you are migrating the solution, start **SAS Framework Data Server** and **JBoss** services.

# Troubleshooting Instructions

### *The Client Fails to Open*

You provided correct logon credentials, but the client application fails to launch.

| Possible Cause | Possible Solution |
|---|---|
| The client application could be incorrectly installed. | Re-install the client application. |
| The client could be pointing to an incorrect server. | Open the **spartoptclnt.ini** file and ensure that the **sasenvironment.xml** file is fetched from the correct server. |

## The Client Cannot Connect to the Server

The client cannot connect to the server.

***Table 4.5*** *Possible Causes and Solutions for Connection Problems*

| Possible Cause | Possible Solution |
|---|---|
| The sas-environment.xml file contains an incorrect URL in the <service-registry> tag. | Provide the correct URL in the sas-environment.xml file. |
| The SAS Application Server is not started at the specified computer and port. | Start the SAS Application Server at the correct computer and port. |
| The SAS Application Server started but with errors. | Open the log files to determine the root cause of the problem and take appropriate action. For more information about some of the possible causes for the errors and their corresponding solutions, see "The SAS Application Server Starts with Errors" on page 46. |
| Network connection between the client and server is broken. | Check the network connection between the client and the server and take corrective actions, if needed. |
| The SAS environment is incorrectly configured. | In SAS Management Console, manually specify the correct URL and port for the server in the SAS environment. |

## The SAS Application Server Starts with Errors

The SAS Application Server starts but with errors.

***Table 4.6*** *Possible Causes and Solutions for Server-Related Errors*

| Possible Cause | Possible Solution |
|---|---|
| The server is incorrectly installed. | Re-install the server. |

| Possible Cause | Possible Solution |
| --- | --- |
| The server could not connect to the database. | Start the SAS database servers by using the correct database computer and port details. Then, verify that the network connection between the server and the database servers is working. |

*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.<br>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.<br>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 |
| | GL_TIME_CAL_ASSOC_CD1 | Specifies the association code for standard time calendar hierarchy: Day → Week | CL1 |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| Extract, transform, and load (ETL) process | GL_TIME_CAL_ASSOC_CD2 | Specifies the association code for standard time calendar hierarchy: Day → Month → Quarter → Year | CL2 |
| | GL_TIME_LTF_ASSOC_CD | Specifies the association code for long-term forecasting time calendar hierarchy: Day → Month → Quarter → Year | CL3 |
| | GL_DATA_STORAGE_PATH | Specifies the path in the environment where all the hierarchical forecasting data sets are created. The data sets include actual demand and prediction results along with model repositories and events data.<br>You must set this parameter value to the path where you want to store the forecasting data sets. | For a Windows operating environment: **`C:\`**<br>For a UNIX operating environment: **`Home/`** |
| | GL_DB_DSN | Specifies the data source name for connecting to transactional database server. | spodb |
| | GL_FCST_STF_THRESHOLD | Specifies minimum days of history required to qualify for forecasting. | 175 |
| | GL_MAX_BUDGET | Specifies a high value for the inventory cost budget to be used for customer-facing facility analysis in the scenarios workspace. | 1E15 |
| | GL_UNSPECIFIED_CATEGORY | Specifies a value for the item category and the facility or organization hierarchy that are not specified. | Unspecified |
| | GL_MIXED_HIERARCHY | Specifies mixed hierarchy for forecasting. | MIXED |
| | GL_NOT_MIXED_HIERARCHY | Specifies non-mixed hierarchy for forecasting. | NOT MIXED |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| Extract, transform,and load (ETL) process | GL_ITEM_PROFILE_A | Specifies class A item profile. | A |
| | GL_ITEM_PROFILE_B | Specifies class B item profile. | B |
| | GL_ITEM_PROFILE_C | Specifies class C item profile. | C |
| | GL_DAY_BASE_PERIOD | Specifies a short value to indicate day as the base period. | D |
| | GL_SHORT_INTERNAL | Specifies a short value to indicate that the facility is internal. | I |
| | GL_DB_HOST_NAME | Specifies the host name where the Framework Data Server is installed. | puneisblademis1.apac.sas.com |
| | GL_SAS_META_CFG_PATH | Specifies the path of the fds.parms file in the data folder. By default, the file is saved in the configuration folder (for example, **<SASCONFIG>**/Lev**<N>**/AppData/SASServicePartsOptimization/data).. *Note:* Use front slashes (/) in the path. | For a Windows operating environment: **C:** For a UNIX operating environment: **Home/** |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| Forecasting process | GL_SPO_LOG_DISPLAY_FLAG | Specifies whether you want to create log files for the forecasting batch process or not.<br>If you want to create the log files, then specify the parameter value as Y, else the parameter value must be N. | Y |
| | GL_FCST_TIME_HIER_LVL_NO_WK | Specifies the time hierarchy level number for week. | 2 |
| | GL_FCST_TIME_HIER_LVL_NO_MTH | Specifies the time hierarchy level number for month. | 3 |
| | GL_FCST_TIME_HIER_LVL_NO_QTR | Specifies the time hierarchy level number for quarter. | 4 |
| | GL_SPO_OPTIMIZATION_FLAG | Specifies whether you want to optimize the forecasting batch process.<br>If you want to optimize the process, then specify the parameter value as Y, else the value must be N. | Y |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| Forecasting process | GL_SPO_USE_ID_FS_FLAG | Specifies whether the ID values of the facility and item pairs are appended to the names of the tables that are used in SAS Forecast Studio. The facility and item pairs are identified by their IDs and retained key values.<br>If you want to append the ID values, then you specify the parameter value as Y, else the parameter value must be N.<br>*Note:* Ensure that either this parameter or the GL_SPO_USE_NM_FS_FLAG parameter value is Y. | Y |
| | GL_SPO_USE_NM_FS_FLAG | Specifies whether the name values of the facility and item pairs are appended to the names of the tables that are used in SAS Forecast Studio. The facility and item pairs are identified by their names and retained key values.<br>If you want to append the name values, then you specify the parameter value as Y, else the parameter value must be N.<br>*Note:* Ensure that either this parameter or the GL_SPO_USE_ID_FS_FLAG parameter value is Y. | N |
| | GL_FCST_RUN_QA_MACRO_IND | Indicates whether the quality assurance macro is to be run in the forecasting batch process or not.<br>If you want to run the quality assurance macro, then specify the parameter value as 1, else the value must be 0. | 1 |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| Forecasting process | GL_SPO_USE_SUBGROUP_FLAG | Specifies whether you want to use the statistically created subgroup information in the forecasting batch process.<br>If you want to use the subgroup information, then specify the parameter value as Y, else the value must be N. | N |
| | GL_SPO_TIMESERIES_DIV_NUM | Specifies the number of time series that must be considered when the Diagnose and Forecast Engine procedures run. | 100 |
| | GL_SHORT_DIAGNOSE | Specifies a short value to indicate DIAGNOSE. | D |
| | GL_SHORT_SELECT | Specifies a short value to indicate SELECT. | S |
| | GL_FCST_IO_RUN_INTERVAL | Specifies the time interval between the forecasting batch process and inventory optimization batch process runs. | 2 |
| Forecasting process | GL_FCST_QA_MIN_OBS_WK | Specifies the minimum number of observations that are required to run the quality assurance macro for the week base period. | 5 |
| | GL_FCST_QA_MIN_OBS_MTH | Specifies the minimum number of observations that are required to run the quality assurance macro for the month base period. | 5 |
| | GL_FCST_QA_MIN_OBS_QTR | Specifies the minimum number of observations that are required to run the quality assurance macro for the quarter base period. | 5 |

| 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.<br>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).<br>You must change this parameter value to the date from when your inventory optimization process starts. | date() |
| | 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.<br>You must modify this parameter as per your selected service type. | FR |

| 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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>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.<br>You must modify this parameter as per your currency. | USD |
| | GL_FCST_UI_DISPLAY_VALUES | Specifies the number of periods for which actual demand quantity is to be displayed in the Forecast Management workspace.<br>You must modify this parameter to the number of past periods for which you want to view data on the user interface. | 5 |
| | GL_IO_METRICS_HISTORY_PERIOD | Specifies the number of base periods of historical data to be displayed in the Inventory Analysis workspace.<br>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.<br>You can modify the coefficient value as per your requirement. | 1.96 |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| User interface | GL_IO_ITEM_CAT_LVLNO | Specifies the number of levels for item categories to be displayed in the Inventory Analysis workspace.<br>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.<br>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.<br>You can modify this threshold value as per your requirement. | 0.02 |
| 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.<br>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 |

| Category (Parameter Used In) | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| Independent variables | GL_IND_VAR<N> | Specifies the value for independent variable <N>, where <N> is a number from 1 to 10. There is one parameter for each variable. | Independent variable <N><br>Here N denotes a number from 1 to 10 depending on the number of the independent variable. For example, for GL_IND_VAR1, the value is Independent variable 1. |

# 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 |
|---|---|---|---|
| 04_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 |

| Job Name or Directory | Parameter Name | Parameter Description | Default Value |
|---|---|---|---|
| 02_LTF_PROCESS | OUTLIER_START_PERIOD | Specifies the period from which the outlier detection process is to be started. | 1 |
| | MAX_NO_MISSING | Specifies the maximum number of missing values that are acceptable in a time series. | 0 |
| All initial and subsequent forecasting jobs in the **C_WEEK**, **D_MONTH**, **E_QUARTER**, and **F_INTERMITTENT DEMAND** folders in the following basic location: **/Products/SAS Service Parts Optimization/Service Parts Optimization 4.3/SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF** | TOTAL_THREAD_NO | Specifies the total number of threads that need to be spawned for performing parallel processing. | 1 |
| | FCST_LOG_PATH | Specifies the location where the log file is to be stored. | For a Windows operating environment: **C:\log**<br>For a UNIX operating environment: **Home/log**<br>*Note:* Ensure that the log folder exists before you update this parameter. |
| | MEMSIZE | Specifies the memory that is allocated for each thread that is spawned on the server. | 3G |

*Appendix 2*

# ETL Job Details



## Overview

You can access all ETL jobs from SAS Data Integration Studio. The basic location of all ETL jobs for SAS Service Parts Optimization is as follows: **`/Products/SAS Service Parts Optimization/Service Parts Optimization 4.3`**.

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 Service Parts 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 76.

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 | | |
| **`SPO_JOBS/01_DIMENSION_JOBS`** | | | |
| 1 | FACILITY_DIM | None | None |
| 2 | ITEM_DIM | None | None |
| | Forecasting Jobs | | |
| **`SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ A_ABT_POPULATION_JOBS/1_ABT_DATA_PREPARATION`** | | | |
| 3 | 1_NPF_DATA_PREPARATION_INT | 1,2 | Run these jobs only in the first run of the forecasting process. |
| 4 | 2_STF_DATA_PREPARATION_INT | | |
| **`SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ A_ABT_POPULATION_JOBS/2_TIME_GRAIN/A_WEEK`** | | | |
| 5 | 1_CREATE_WEEK_NPF_ABT_INT | 3 | Run these jobs only in the first run of the forecasting process. |
| 6 | 2_CREATE_WEEK_STF_ABT_INT | 4 | |
| **`SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ A_ABT_POPULATION_JOBS/2_TIME_GRAIN/B_MONTH`** | | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 7 | 1_CREATE_MONTH_NPF_ABT_INT | 3 | Run these jobs only in the first run of the forecasting process. |
| 8 | 2_CREATE_MONTH_STF_ABT_INT | 4 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ A_ABT_POPULATION_JOBS/2_TIME_GRAIN/C_QUARTER** | | | |
| 9 | 1_CREATE_QUARTER_NPF_ABT_INT | 3 | Run these jobs only in the first run of the forecasting process. |
| 10 | 2_CREATE_QUARTER_STF_ABT_INT | 4 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ A_ABT_POPULATION_JOBS/3_NO_HISTORY_ABT** | | | |
| 11 | 1_NPF_DEMAND_ABT_LESS_HISTORY_INT | 3–10 | Run this job only in the first run of the forecasting process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/ 1_FCST_GP_FAC_ITEM_CONTROL_DATA_PREPARARATION** | | | |
| 12 | 1_CONTROL_DATA_PREPARARATION_INT | 11 | Run this job only in the first run of the forecasting process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/2_WEEK** | | | |
| 13 | 1_CREATE_WK_FCST_GP_FACILITY_ITEM_TABLE_INT | 12 | Run this job only in the first run of the forecasting process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/3_MONTH** | | | |
| 14 | 1_CREATE_MTH_FCST_GP_FACILITY_ITEM_TABLE_INT | 12 | Run this job only in the first run of the forecasting process. |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/4_QUARTER** | | | |
| 15 | 1_CREATE_QTR_FCST_GP_FACILITY_ITEM_TABLE_INT | 12 | Run this job only in the first run of the forecasting process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/C_WEEK** | | | |
| 16 | 1_WK_DATA_PREPARATION_JOB_INT | 1,2,5–15 | Run these jobs only in the first run of the forecasting process. |
| 17 | 2_WK_HIERARCHY_CREATION_INT | 16 | |
| 18 | 3_WK_FORECAST_INT | 17 | |
| 19 | 4_WK_RECONCILIATION_INT | 18 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/D_MONTH** | | | |
| 20 | 1_MTH_DATA_PREPARATION_JOB_INT | 1,2,5–15 | Run these jobs only in the first run of the forecasting process. |
| 21 | 2_MTH_HIERARCHY_CREATION_INT | 20 | |
| 22 | 3_MTH_FORECAST_INT | 21 | |
| 23 | 4_MTH_RECONCILIATION_INT | 22 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/E_QUARTER** | | | |
| 24 | 1_QTR_DATA_PREPARATION_JOB_INT | 1,2,5–15 | Run these jobs only in the first run of the forecasting process. |
| 25 | 2_QTR_HIERARCHY_CREATION_INT | 24 | |
| 26 | 3_QTR_FORECAST_INT | 25 | |
| 27 | 4_QTR_RECONCILIATION_INT | 26 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/INITIAL_FORECASTING_JOBS/ F_INTERMITTENT_DEMAND** | | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 28 | GET_INTERMITTENT_DEMAND_INT | 19,23,27 | Run this job only in the first run of the forecasting process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ A_INCREMENTAL_ABT_POPULATION_JOBS/1_TIME_GRAIN/A_WEEK** | | | |
| 29 | 1_WEEK_NPF_INCREMENTAL_DATA_PREPARATION | 3–11 | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 30 | 2_CREATE_INCREMENTAL_WEEK_NPF_ABT | 29 | |
| 31 | 3_WEEK_STF_INCREMENTAL_DATA_PREPARATION | 3–11 | |
| 32 | 4_CREATE_INCREMENTAL_WEEK_STF_ABT | 31 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ A_INCREMENTAL_ABT_POPULATION_JOBS/1_TIME_GRAIN/B_MONTH** | | | |
| 33 | 1_MONTH_NPF_INCREMENTAL_DATA_PREPARATION | 3–11 | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 34 | 2_CREATE_INCREMENTAL_MONTH_NPF_ABT | 33 | |
| 35 | 3_MONTH_STF_INCREMENTAL_DATA_PREPARATION | 3–11 | |
| 36 | 4_CREATE_INCREMENTAL_MONTH_STF_ABT | 35 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ A_INCREMENTAL_ABT_POPULATION_JOBS/1_TIME_GRAIN/C_QUARTER** | | | |
| 37 | 1_QUARTER_NPF_INCREMENTAL_DATA_PREPARATION | 3–11 | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 38 | 2_CREATE_INCREMENTAL_QUARTER_NPF_ABT | 37 | |
| 39 | 3_QUARTER_STF_INCREMENTAL_DATA_PREPARATION | 3–11 | |
| 40 | 4_CREATE_INCREMENTAL_QUARTER_STF_ABT | 39 | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ A_INCREMENTAL_ABT_POPULATION_JOBS/2_NO_HISTORY_ABT** | | | |
| 41 | 1_NPF_DEMAND_ABT_LESS_HISTORY | 29–40 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ A_INCREMENTAL_ABT_POPULATION_JOBS/3_EDIT_PARAMETERS** | | | |
| 42 | 1_APPEND_CHANGED_HPF_PARAM_TO_CONTROL_TABLE | None | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 43 | 2_CREATE_HPF_PREFERENCES_UI_TABLE | 42 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/ 1_FCST_GP_FAC_ITEM_CONTROL_DATA_PREPARARATION** | | | |
| 44 | 1_CONTROL_DATA_PREPARARTION | 41 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/2_WEEK** | | | |
| 45 | 1_CREATE_WK_FCST_GP_FACILITY_ITEM_TABLE | 44 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/ A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/ B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/3_MONTH** | | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 46 | 1_CREATE_MTH_FCST_GP_FACILITY_ITEM_TABLE | 44 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/B_CREATE_FCST_GP_FAC_ITEM_CONTROL_TABLE/4_QUARTER** | | | |
| 47 | 1_CREATE_QTR_FCST_GP_FACILITY_ITEM_TABLE | 44 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/C_WEEK** | | | |
| 48 | 1_WK_ADDITIONAL_FG_CONSIDERATION | 29–47 | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 49 | 2_WK_HIERARCHY_CREATION | 48 | |
| 50 | 3_WK_FORECAST | 49 | |
| 51 | 4_WK_RECONCILIATION | 50 | |
| 52 | 5_WK_QUALITY_ANALYSIS | 51 | |
| 53 | 6_WK_NORMAL_FORECASTS | 52 | |
| 54 | 7_WK_LOW_ACCURACY_FORECASTS | 52 | |
| 55 | 8_WK_FORECAST_WITH_WARNINGS | 52 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/D_MONTH** | | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 56 | 1_MTH_ADDITIONAL_FG_CONSIDERATION | 29–47 | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 57 | 2_MTH_HIERARCHY_CREATION | 56 | |
| 58 | 3_MTH_FORECAST | 57 | |
| 59 | 4_MTH_RECONCILIATION | 58 | |
| 60 | 5_MTH_QUALITY_ANALYSIS | 59 | |
| 61 | 6_MTH_NORMAL_FORECASTS | 60 | |
| 62 | 7_MTH_LOW_ACCURACY_FORECASTS | 60 | |
| 63 | 8_MTH_FORECAST_WITH_WARNINGS | 60 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/E_QUARTER** | | | |
| 64 | 1_QTR_ADDITIONAL_FG_CONSIDERATION | 29–47 | Do not run these jobs in the first run of the forecasting process. Run the jobs from the second run of the process. |
| 65 | 2_QTR_HIERARCHY_CREATION | 65 | |
| 66 | 3_QTR_FORECAST | 65 | |
| 67 | 4_QTR_RECONCILIATION | 66 | |
| 68 | 5_QTR_QUALITY_ANALYSIS | 67 | |
| 69 | 6_QTR_NORMAL_FORECASTS | 68 | |
| 70 | 7_QTR_LOW_ACCURACY_FORECASTS | 68 | |
| 71 | 8_QTR_FORECAST_WITH_WARNINGS | 68 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/F_INTERMITTENT_DEMAND** | | | |
| 72 | GET_INTERMITTENT_DEMAND | 55,63,71 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/A_ABT_FORECASTING_JOBS/HPF/SUBSEQUENT_FORECASTING/G_TRUNCATE_INCREMENTAL_ABT | | | |
| 73 | TRUNCATE_INCREMENTAL_ABT | 29–40 | Do not run this job in the first run of the forecasting process. Run the job from the second run of the process. |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/B_LTF/A_CLUSTERING** | | | |
| 74 | 01_SELECT_FACILITY_ITEM_FOR_LTF_CLUSTER | None | None |
| 75 | 02_CALCULATE_YEAR_DEMAND_HISTORY_LTF_CLUSTER | 74 | |
| 76 | 03_LTF_GET_CLUSTER_PARAM | None | |
| 77 | 04_LTF_CLUSTERING_PROCESS | 74, 75 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/B_LTF/B_BATCH_PROCESS** | | | |
| 78 | 01_LTF DEMAND_DATA_AGGREGATION | 74–77 | None |
| 79 | 02_LTF_PROCESS | 78 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/C_UI_TABLES/PRE-UIART JOBS/01_WK_PRE_UIART_JOBS** | | | |
| 80 | 01_WK_CREATE_FCST_CONTROL_DATA | 16–19 for the first run of the forecasting process and 48–55 for the subsequent runs of the forecasting process. | None |
| 81 | 02_WK_CREATE_IDM_ABT | | |
| 82 | 03_WK_CREATE_HPF_FORECAST_OUTFOR_ART | | |
| SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/C_UI_TABLES/PRE-UIART JOBS/02_MTH_PRE_UIART_JOBS | | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 83 | 01_MTH_CREATE_FCST_CONTROL_DATA | 20–23 for the first run of the forecasting process and 56–63 for the subsequent runs of the forecasting process. | None |
| 84 | 02_MTH_CREATE_IDM_ABT | | |
| 85 | 03_MTH_CREATE_HPF_FORECAST_OUTFOR_ART | | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/C_UI_TABLES/PRE-UIART JOBS/03_QTR_PRE_UIART_JOBS** | | | |
| 86 | 01_QTR_CREATE_FCST_CONTROL_DATA | 24–27 for the first run of the forecasting process and 64–71 for the subsequent runs of the forecasting process. | None |
| 87 | 02_QTR_CREATE_IDM_ABT | | |
| 88 | 03_QTR_CREATE_HPF_FORECAST_OUTFOR_ART | | |
| **SPO_JOBS/02_ANALYTICS_JOBS/FORECASTING/C_UI_TABLES/UIART JOBS** | | | |
| 89 | 01_CREATE_UI_TABLES | 80-88 and 11 for the first run of the forecasting process and 41 for the subsequent runs of the forecasting process. | None |
| 90 | 02_CREATE_DEMAND_PROJECTION_DETAIL | 80,82,83,85,86, 88 | |
| 91 | 03_NO_HISTORY_DP_DETAIL | 11, 41 | |
| 92 | 04_ONE_TIME_BUY | 74-79 | |
| 93 | 05_CREATE_TIMESERIES_DETAIL | 80,82,83,85,86, 88 | |
| 94 | 06_NO_HISTORY_TIMESERIES_DETAIL | 11, 41 | |
| 95 | 07_CREATE_IDM_DETAIL | 81,84,87,82,85, 88 | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 96 | 08_CREATE_IDM_TIMESERIES_DETAIL | 81,84,87,82,85,88 | None |
| 97 | 09_FILTER_ATTRIBUTE_RANGE | 90, 95 | |
| 98 | 10_NO_HIST_FILTER_ATTRIBUTE_RANGE | 91, 94 | |
| 99 | 11_CREATE_SUBGROUP_PROPERTY_TABLE | None | |
| 100 | 12_CREATE_IMPORT_FORECAST_LEAD_PERIOD | None | |
| 101 | 13_DELETE_IDM_FCST_CNTRL_DATA | None | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/A_FORECAST_IO_INTEGRATION** | | | |
| 102 | 01_WK_CREATE_FORECAST_OUTFOR_ART | 16–19 for the first run of the forecasting process and 48–55 for every subsequent run. | None |
| 103 | 02_MTH_CREATE_FORECAST_OUTFOR_ART | 20–23 for the first run of the forecasting process and 56–63 for every subsequent run. | |
| 104 | 03_QTR_CREATE_FORECAST_OUTFOR_ART | 24–27 for the first run of the forecasting process and 64–71 for every subsequent run. | |
| 105 | 04_AGGREGATE_FCST_ART | 102–104 | |
| 106 | 05_AGGREGATE_FCST_ART_FROM_NPF | 102–104 | |
| Inventory Optimization Jobs | | | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/B_CREATE_IO_ABT** | | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 107 | 01_POPULATE_NODE_DATA | 28 for first run of the forecasting process and 72 for every subsequent run. | None |
| 108 | 02_POPULATE_ARC_DATA | None | |
| 109 | 03_POPULATE_DEMAND_DATA | 105, 106, 28 for the first run of the forecasting process and 72 for every subsequent run. | |
| 110 | 04_POPULATE_INVENTORY_DATA | None | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/C_IO_PROCESS** | | | |
| 111 | 01_POLICY_ORDER_KPI_1 | 107-110 | None |
| 112 | 02_TRANSSHIPMENT | 111 | |
| 113 | 03_POLICY_ORDER_KPI_2 | 112 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/01_ATTRIBUTE_JOBS** | | | |
| 114 | FACILITY_ATTR | None | None |
| 115 | ITEM_ATTR | | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/02_METRICS_CALC** | | | |
| 116 | 01_GET_LEAD_TIME | None | None |
| 117 | 02_GET_SL_DOWNSTREAM | | |
| 118 | 03_GET_SL_UPSTREAM | | |
| 119 | 04_GET_DEMAND_ACT | | |
| 120 | 05_GET_DEMAND_TGT | | |
| 121 | 06_GET_COST_ACT | | |
| 122 | 07_GET_COST_TGT | | |
| 123 | 08_GET_KPI_PREDICT_MIN_MAX | | |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/ 03_METRICS_BASE_PERIOD** | | | |
| 124 | 01_IO_METRICS_PERIODIC_ART_ HIST | None | None |
| 125 | 02_IO_METRICS_PERIODIC_ART_ LOAD | | |
| 126 | 03_IO_METRICS_PERIODIC_ART_ REPLACE | | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/04_METRICS_DAILY** | | | |
| 127 | 01_IO_METRICS_ALERT_SETTIN GS_LOAD | None | None |
| 128 | 02_IO_METRICS_ALERT_SETTIN GS_REPLACE | | |
| 129 | 03_IO_METRICS_ART_LOAD | | |
| 130 | 04_IO_METRICS_ART_VENDOR_ DETAILS | | |
| 131 | 05_IO_METRICS_ART_REPLACE | | |
| 132 | 06_IO_METRICS_ITEM_FACITY_N ETWORK | | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/ 05_SCENARIO_BASE_PERIOD** | | | |
| 133 | SCENARIO_UI_TABLES | None | None |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/ 06_ORDERS_BASE_PERIOD** | | | |
| 134 | 01_CREATE_LOCK_PLAN_SETTIN G | None | Run this job only once, after you populate all tables of the SDL library. |

| Job Number | Job Name | Predecessors | Comments |
|---|---|---|---|
| 135 | 02_CREATE_SKU_BUCKETS | 113, 115,128 | None |
| 136 | 03_POPULATE_DRP_TABLE | 135 | |
| 137 | 04_POPULATE_TRANSFER_COST_DETAILS | 136 | |
| 138 | 05_POPULATE_SUBSTITUTE_ITEM | 137 | |
| 139 | 06_POPULATE_ORDER_RESULT | 138 | |
| 140 | 07_POPULATE_REPL_BY_PERIOD_LOOKUP | 139 | None |
| 141 | 08_POPULATE_ORDER_DETAIL | 140 | |
| 142 | 09_POPULATE_REPL_PLAN_DETAIL | 141 | |
| 143 | 10_POPULATE_REPL_PLAN_METRICS | 142 | |
| 144 | 11_CREATE_REPL_PLAN_SUMMARY | 143 | |
| **SPO_JOBS/02_ANALYTICS_JOBS/IO/D_UI_ART_JOBS/07_ORDERS_OPTIONAL** | | | |
| 145 | LOAD_ORDER_DETAILS | 143 | 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 |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 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_FCST_ABT | This job creates a structure for all the analytical base tables (ABTs) that are required by the forecasting process. | None |
| 04_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.<br>• General Calendar: Day → Week<br>• 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. |
| 05_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. |
| 06_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. |
| 07_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. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 08_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. |
| 09_CREATE_USER_TABLES | This job creates the following initial tables that are required for populating the user interface:<br>• UIART.User_Mapping<br>• TSDB.User_Preferences<br>• TSDB.IO_Metrics_Alertsettings<br>• TSDB.Scenarios<br>• TSDB.Scenario_Params<br>• TSDB.Evaluate_Forecast_Parameter<br>• UIART.Evaluated_Timeseries_Result<br>• TSDB.Forecast_Data_Param | None |
| 10_POPULATE_HPF_PREFERENCES | This job populates the tables Control.HPF_Preferences and Control.Forecast_GR_Seasonality. The Control.Forecast_GR_Seasonality table divides forecast groups on weekly, monthly, or quarterly basis. | None |
| 11_POPULATE_HPF_PREFERENCE_OPTIONS_VALUE | This job stores the codes for all combinations of the HPF preferences in the Control.HPF_Preferences_Options_Value table. | The Control.HPF_Preferences table is required. |
| 12_POPULATE_HPF_PREFERENCE_FG_TABLE | This job populates the table that contains the parameters at forecast group level that are displayed on the user interface. Also, the job creates empty tables for the forecast subgroup level and time series level. | The Control.HPF_Preferences and Control.HPF_Preferences_Options_Value tables are required. |
| 13_CREATE_EVENTS_TEMPLATE | This job creates tables that contain events that are used by the forecasting batch process. | None |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 14_CREATE_TIME_GRAIN_FG | This job creates a table for every base period. These tables contain a list of forecast groups for the corresponding base period. | The SDL.Forecast_Group table is required. |

# 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. |

# Forecasting Jobs

## *Initial Forecasting Jobs*

The following table provides an overview of the initial forecasting jobs. These jobs must be run only once, after you install the solution.

*Note:* Some job names and table names include the text **<BASE_PERIOD>** that can be week, month, or quarter depending on the selected base period.

***Table A2.4*** *Initial Forecasting Jobs*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 1_NPF_DATA_PREPARATION_INT | This job populates the Scratch.NPF_FCST_DMD_Temp table to include time series with demand history less than the number of days that you have predefined in a global parameter. These time series include successor items that contain the history of the predecessor items. | The following tables are required:<br>• SDL.Network_Model<br>• SDL.Facility_Item_X_Network<br>• SDL.Facility_Item_Demand<br>• SDL.Item_Succession<br>• SDL.Facility_X_Item<br>• DIM.Time_Dim<br>• SDL.Facility_X_Item<br>• SDL.Facility_Item_Demand |
| 2_STF_DATA_PREPARATION_INT | This job populates the Scratch.STF_FCST_DMD_Temp table to include time series with demand history greater than the number of days that you have predefined in a global parameter. | The following tables are required:<br>• SDL.Network_Model<br>• SDL.Facility_Item_X_Network<br>• SDL.Facility_Item_Demand<br>• SDL.Item_Succession<br>• SDL.Facility_X_Item<br>• DIM.Time_Dim<br>• SDL.Facility_X_Item<br>• SDL.Facility_Item_Demand |
| 1_CREATE_**<BASE_PERIOD>**_NPF_ABT_INT | These jobs create a table in the ABT library for every forecast group for the chosen base period. The tables contain time series with demand history less than the number of days that you have predefined in a global parameter. These time series include successor items that contain the history of the predecessor items. | The following tables are required:<br>• SDL.Forecast_Group_Item_Detail<br>• Scratch.NPF_Fcst_Dmd_Temp<br>• SDL.Forecast_Group<br>• DIM.Item_Dim<br>• DIM.Facility_Dim<br>• DIM.Facility_Item_Ind_Variable |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 2_CREATE_**<BASE_PERIOD>**_STF_ABT_INT | These jobs create a table in the ABT library for every forecast group for the chosen base period. The tables contain time series with demand history greater than the number of days that you have predefined in a global parameter. If the tables already exist, then these jobs append new rows to the existing tables. | The following tables are required:<br>• SDL.Forecast_Group_Item_Detail<br>• Scratch.STF_Fcst_Dmd_Temp<br>• SDL.Forecast_Group<br>• DIM.Item_Dim<br>• DIM.Facility_Dim<br>• DIM.Facility_Item_Ind_Variable |
| 1_NPF_DEMAND_ABT_LESS_HISTORY_INT | This job creates the ABT.NPF_Demand_ABT_Less_History table that contains the time series with demand history less than the number of days that you have predefined in a global parameter. These time series do not qualify for the forecasting process due to less history. | The Scratch.NPF_Demand_ABT_Less_History, DIM.Item_Dim, and DIM.Facility_Dim tables are required. |
| 1_CONTROL_DATA_PREPARARATION_INT | This job creates the Scratch.Forecast_GR_Fac_Item_FCST_GR table that contains all time series with customer-facing facilities. This table is further split into multiple tables depending on the base period. | The following tables are required:<br>• ABT.NPF_Demand_Abt_Less_Hist<br>• SDL.Forecast_Group_Item_Detail<br>• SDL.Facility_X_Item<br>• DIM_Item_Dim<br>• DIM.Facility_Dim |
| 1_CREATE_**<BASE_PERIOD>**_FCST_GP_FACILITY_ITEM_TABLE_INT | These jobs create a table in the Control library for every forecast group for the chosen base period. | The SDL.Forecast_Group and Scratch.Forecast_Gr_Fac_Item_Fcst_Gr tables are required. |
| 1_**<BASE_PERIOD>**_DATA_PREPARATION_JOB_INT | These jobs remove the unused hierarchies of item and facility per base period. By default, ten item hierarchies and five facility hierarchies are available. | None |
| 2_**<BASE_PERIOD>**_HIERARCHY_CREATION_INT | These jobs create hierarchical tables from the analytical base tables. Data for each hierarchy is placed in separate folders. | None |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 3_**<BASE_PERIOD>**_FORECAST_INT | These jobs perform forecasting that includes selecting models and parameters and saving the forecast results. | None |
| 4_**<BASE_PERIOD>**_RECONCILIATION_INT | These jobs reconcile forecast results across all the hierarchies. | None |
| GET_INTERMITTENT_DEMAND_INT | This job identifies the intermittent demand series and retrieves the parameters that are required for the inventory optimization process. The job populates the table ABT.IO_**<BASE_PERIOD>**_IDM_ABT. | The ABT.Forecast_Demand_Data_ABT and ABT.IDM_Facility_Item tables are required. |

## *Subsequent Forecasting Jobs*

The following table provides an overview of the subsequent forecasting jobs. These jobs must be run as per the base period.

***Table A2.5*** *Subsequent Forecasting Jobs*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 1_**<BASE_PERIOD>**_NPF_INCREMENTAL_DATA_PREPARATION | These jobs update tables in the Scratch library for every forecast group for the chosen base period. The tables contain time series with demand history less than the predefined number of intervals. Only incremental records are appended to the tables. | The following tables are required:<br>• SDL.Facility_Item_X_Network<br>• SDL.Network_Model<br>• SDL.Facility_Item_Demand<br>• SDL.Item_Succession<br>• SDL.Facility_X_Item<br>• DIM.Time_Dim |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 2_CREATE_INCREMENTAL_**<BASE_PERIOD>**_NPF_ABT | These jobs create incremental tables in the ABT library for every forecast group for the chosen base period. | The following tables are required:<br>• SDL.Forecast_Group_Item_Detail<br>• Scratch.**BASE_PERIOD>**_Npf_Fcst_Inc_Dmd_Temp<br>• SDL.Forecast_Group<br>• DIM.Item_Dim<br>• DIM.Facility_Dim<br>• DIM.Facility_Item_Ind_Variable |
| 3_**<BASE_PERIOD>**_STF_INCREMENTAL_DATA_PREPARATION | These jobs update tables in the Scratch library for every forecast group for the chosen base period. The tables contain time series with demand history greater than the predefined number of intervals. Only incremental records are appended to the tables. | The following tables are required:<br>• SDL.Facility_Item_X_Network<br>• SDL.Network_Model<br>• SDL.Facility_Item_Demand<br>• SDL.Item_Succession<br>• SDL.Facility_X_Item<br>• DIM.Time_Dim |
| 4_CREATE_INCREMENTAL_**<BASE_PERIOD>**_STF_ABT | These jobs create incremental tables in the ABT library for every forecast group for the chosen base period. | The following tables are required:<br>• SDL.Forecast_Group_Item_Detail<br>• Scratch.**<BASE_PERIOD>**_Stf_Fcst_Inc_Dmd_Temp<br>• SDL.Forecast_Group<br>• DIM.Item_Dim<br>• DIM.Facility_Dim<br>• DIM.Facility_Item_Ind_Variable |
| 1_NPF_DEMAND_ABT_LESS_HISTORY | This job creates the ABT.NPF_Demand_ABT_Less_History table that contains the time series with demand history less than the predefined number of days. These time series do not qualify for the forecasting process due to less history. | The Scratch.NPF_Demand_ABT_Less_History, DIM.Item_Dim, and DIM.Facility_Dim tables are required. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 1_APPEND_CHANGED_HPF_PARAM_TO_CONTROL_TABLE | This job appends the parameters that are modified from the user interface to the Control.HPF_Preferences table. | The following tables are required:<br>• TSDB.HPF_Preferences_UI_Fg<br>• Control.HPF_Preference_Options_Value<br>• Control.HPF_Preferences<br>• TSDB.HPF_Preferences_UI_Sg<br>• TSDB.HPF_Preferences_UI_Ts |
| 2_CREATE_HPF_PREFERENCES_UI_TABLE | This job sets all the parameters that are modified from the user interface as default parameters. | The following tables are required:<br>• TSDB.HPF_Preferences_UI_Fg<br>• TSDB.HPF_Preferences_UI_Sg<br>• TSDB.HPF_Preferences_UI_Ts |
| 1_CONTROL_DATA_PREPARARTION | This job creates the Scratch.Forecast_GR_Fac_Item_FCST_GR table that contains all time series with customer-facing facilities. This table is further split into multiple tables depending on the base period. | None |
| 1_CREATE_**<BASE_PERIOD>**_FCST_GP_FACILITY_ITEM_TABLE | These jobs create a table in the Control library for every forecast group for the chosen base period. | The Scratch.Forecast_GR_Fac_Item_FCST_Gr table is required. |
| 1_**<BASE_PERIOD>**_ADDITIONAL_FG_CONSIDERATION | These jobs remove the unused hierarchy levels from the newly added forecast groups. | All incremental ABTs for the corresponding base period. |
| 2_**<BASE_PERIOD>**_HIERARCHY_CREATION | These jobs create the hierarchical tables from the incremental analytical base tables. Data for each table is stored in separate folders. | All incremental ABTs for the corresponding base period. |
| 3_**<BASE_PERIOD>**_FORECAST | These jobs perform the forecasting process with model parameters that are obtained in the initial run. | The Control.HPF_Preferences table is required. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 4_**<BASE_PERIOD>**_RECONCILIATION | These jobs reconcile forecast results across all the hierarchies. | None |
| 5_**<BASE_PERIOD>**_QUALITY_ANALYSIS | These jobs populate the Control.**<Base_Period>**_Forecast_Control_Data table with results of the quality analysis process. | None |
| 6_**<BASE_PERIOD>**_NORMAL_FORECASTS | These jobs remove the inaccurate forecast results and forecasts with warnings from the analytical result tables. | None |
| 7_**<BASE_PERIOD>**_LOW_ACCURACY_FORECASTS | These jobs rediagnose the time series with less accurate forecast results to fit a better model. | None |
| 8_**<BASE_PERIOD>**_FORECAST_WITH_WARNINGS | These jobs re-evaluate the time series with less accurate forecast results to select better model parameters. | None |
| GET_INTERMITTENT_DEMAND | This job identifies the intermittent demand series and retrieves the parameters that are required for the inventory optimization process. The job populates the table ABT.IO_**<BASE_PERIOD>**_IDM_ABT. | The ABT.Forecast_Demand_Data_ABT and ABT.IDM_Facility_Item tables are required. |
| TRUNCATE_INCREMENTAL_ABT | This job truncates all the incremental analytical base tables to retain only new records that are used in the next incremental run. | All the incremental ABTs for all base periods. |

### *Clustering Jobs for Long-Term Forecasting*

The following table provides an overview of the clustering jobs for long-term forecasting. These jobs must be run when you want to perform clustering.

***Table A2.6*** *Clustering Jobs for Long-Term Forecasting*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 01_SELECT_FACILITY_ITEM_FOR_LTF_CLUSTER | This job populates the table Scratch.LTF_Items_TMP with time series details for all forecast groups that are to be used for the clustering process for long-term forecasting. | The tables SDL.Forecast_Group and SDL.Forecast_Group_Item_Detail are required. |
| 02_CALCULATE_YEAR_DEMAND_HISTORY_LTF_CLUSTER | This job generates the demand history for all time series at a yearly interval in the ABT.LTF_Cluster_ABT table. The demand history for the complete planning horizon is then used for dividing facility and item pairs into clusters. | The following tables are required:<br>• SDL.Facility_Item_Demand<br>• Scratch.LTF_Items_TMP<br>• DIM.Time_Dim<br>• DIM.Year_Dim |
| 03_LTF_GET_CLUSTER_PARAM | This job populates the parameters that are used while creating clusters for long-term forecasting in the ABT.LTF_Cluster_Param_ABT table. | The table SDL.Forecast_Group is required. |
| 04_LTF_CLUSTERING_PROCESS | This is the actual clustering job. This job divides facility and item pairs into user-specified clusters that are based on demand in the complete planning horizon. The FASTCLUS procedure is used for clustering. This procedure uses demand data as the basis of clustering and divides the facility and item pairs. Before clustering, outliers (if any) are replaced with moving average values. Also, demand data is normalized to avoid any bias due to large or small values of demand for the facility and item pairs. The pattern of demand over the lifetime of the product is used for clustering and not the actual demand values. The table LTF_Cluster_ART stores the cluster mean values and the table LTF_Cluster_Member_ART stores information about the facility and item pairs that belong to different clusters. | The tables ABT.LTF_Cluster_Param_ABT, ABT.LTF_Cluster_ABT, and Forecast_Clustering (a user-written transform) are required. |

### *Batch Process Jobs for Long-Term Forecasting*

The following table provides an overview of the batch process jobs for long-term forecasting. These jobs must be run on a yearly basis.

***Table A2.7*** *Batch Process Jobs for Long-Term Forecasting*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 01_LTF DEMAND_DATA_AGGREGATION | This job performs aggregation of demand data for the facility and item pairs that need to be forecasted and stores the data in the ABT.LTF_Demand_Data_ABT table. The facility and item pairs are taken from the short-term forecasting table. If demand data for facility and item pairs is more than 40% of the planning horizon (as defined for the forecast group), then the facility and item pairs are considered for long-term or end-of-life forecasting. Demand for these facility and item pairs is aggregated at a yearly level and then used for forecasting in the subsequent job. | The tables ABT.Facility_Item_Forecast, ABT.STF_FCST_Demand_ABT, and SDL.Facility_Item_Demand are required. |
| 02_LTF_PROCESS | This job runs the actual long-term forecasting process. Regression and Bass models are used for forecasting. The demand pattern for facility and item pairs is regressed against the mean values of all the clusters and clusters with minimum RMSE values are considered for further analysis. The forecast values are calculated based on demand pattern of selected clusters. | The tables ABT.LTF_Demand_Data_ABT, ABT.LTF_Cluster_Param_ABT, and ABT.LTF_Cluster_ART are required. |

### *Jobs to Prepare Data for the User Interface Table Creation Jobs*

The following table provides an overview of the jobs to be run before running the user interface table creation jobs. These jobs must be run as per the base period.

*Table A2.8 User Interface Table Creation Jobs*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 01_**<BASE_PERIOD>**_CREATE_FCST_CONTROL_DATA | These jobs combine all the tables for the corresponding base period from the Control library into a single table Control.FCST_CONTROL_DATA. | All tables for the chosen base period from the Control library. |
| 02_**<BASE_PERIOD>**_CREATE_IDM_ABT | These jobs combine all the tables for the corresponding base period from the ABT library into a single table ABT.IDM_Facility_Item. | All tables for the intermittent demand items for the chosen base period from the ABT library. |
| 03_**<BASE_PERIOD>**_CREATE_HPF_FORECAST_OUTFOR_ART | These jobs delete existing records from the ABT.HPF_Outfor_ART table for the corresponding base period and then appends records from the ABT.**<BASE_PERIOD>**_Forecast_Outfor_ART**<count>**tables. | The ABT.HPF_Forecast_Outfor_ART, Scratch.**<BASE_PERIOD>**_Forecast_Groups, and **<BASE_PERIOD>**_Forecast_Outfor_ART tables are required. |

## User Interface Table Creation Jobs

The following table provides an overview of the jobs for creating user interface tables. These jobs must be run as per the base period.

*Table A2.9 User Interface Table Creation Jobs*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 01_CREATE_UI_TABLES | This job populates the values of normal, revisited, and low accuracy forecast values in the TSDB.Forecast_Result table. This job presents the results of the quality analysis (QA) process in the format that is required by the user interface of the solution. The job calculates the number of time series that fall in normal, revisited, and low accuracy categories. The job also calculates the number of successor items and the number of time series with insufficient history. | The tables Control.FCST_Control_Data, SDL.Item_Succession, ABT.NPF_Demand_ABT_Less_History are required. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 02_CREATE_DEMAND_PROJECTION_DETAIL | This job populates forecasted demand and other important statistics for facility and item pairs under consideration, for the date when the forecasting process was last run. The job updates the tables UIART.Demand_Projection_Detail, UIART.DP_Detail_Normal, UIART.DP_Detail_Revisited, UIART.DP_Detail_Low_Accuracy, and UIART.DP_Detail_Successor. | The following tables are required:<br>• SDL.Facility_Item_Demand<br>• Control.FCST_Control_Data<br>• SDL.Facility<br>• SDL.Item<br>• SDL.Facility_X_Item<br>• SDL.Employee |
| 03_NO_HISTORY_DP_DETAIL | This job populates the actual demand quantity and the forecasted quantity that you provide for the date when the forecasting batch process was last run in the UIART.No_History_DP_Detail table. The table contains actual and predicted values for the facility and item pairs with insufficient available historical demand. HPF procedures are unable to produce forecasts correctly for these pairs. | The following tables are required:<br>• SDL.Facility_Item_Demand<br>• SDL.Facility<br>• SDL.Item<br>• SDL.Facility_X_Item<br>• SDL.Employee |
| 04_ONE_TIME_BUY | This job summarizes long-term forecast values for all the facility and item combinations that fulfill long-term forecasting qualification test and populates the UIART.LTF_Forecast_UI_ART table. | The ABT.LTF_Forecast_ART, ABT.LTF_Forecast_Summary_ART tables are required. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 05_CREATE_TIMESERIES_DETAIL | This job populates tables UIART.Timeseries_Detail, UIART.Timeseries_Detail_Normal, UIART.Timeseries_Detail_Revisited, UIART.Timeseries_Detail_Low_Accuracy, and UIART.Timeseries_Detail_Successor.<br><br>The table UIART.Timeseries_Detail contains all the time series for all facility and item combinations. The table UIART.Timeseries_Detail_Normal contains time series details for the facility and item pairs with normal forecast results. Similarly, tables UIART.Timeseries_Detail_Revisited and UIART.Timeseries_Detail_Low_Accuracy contain time series details for the facility and item pairs with revisited and low accuracy forecast results, respectively. The table UIART.Timeseries_Detail_Successor contains time series details for facility and item pairs with items in succession. | The tables ABT.HPF_Forecast_ART and Control.FCST_Control_Data are required. |
| 06_NO_HISTORY_TIMESERIES_DETAIL | This job populates actual and user-predicted demand values for time series for all the facility and item pairs with insufficient demand history in the table TSDB.No_History_Timeseries_Detail. | The following tables are required:<br>• ABT.NPF_Demand_ABT_Less_History<br>• SDL.Facility<br>• SDL.Item<br>• SDL.Facility_X_Item<br>• SDL.Employee |
| 07_CREATE_IDM_DETAIL | This job creates the TSDB.DP_Detail_Intermittent table that contains the latest facility and item pair with intermittent demand. | The ABT.IDM_Facility_Item and ABT.HPF_Forecast_Outfor_ART tables are required. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 08_CREATE_IDM_TIMESERIES_DETAIL | This job creates the UIART.Timeseries_Detail_Intermittent table that contains all the time series of a facility and item pair with intermittent demand. | The ABT.IDM_Facility_Item and ABT.HPF_Forecast_Outfor_ART tables are required. |
| 09_FILTER_ATTRIBUTE_RANGE | This job populates the minimum and maximum values of MAPE for all the forecast result categories (normal, revisited, low accuracy, successor items, and no history) in the table UIART.Filter_Attribute_Range. | The table UIART.Demand_Projection_Detail is required. |
| 10_NO_HIST_FILTER_ATTRIBUTE_RANGE | This job populates the maximum and minimum values of demand quantity, demand error, external demand variance, grouped by forecast groups in the table UIART.No_Hist_Filter_Attribute_Range. | The table UIART.No_History_DP_Detail is required. |
| 11_CREATE_SUBGROUP_PROPERTY_TABLE | This job populates the UIART.Sub_Group_Property table that contains all the used forecast subgroup details. | The SDL.Forecast_Group_Item_Detail and SDL.Forecast_Subgroup tables are required. |
| 12_CREATE_IMPORT_FORECAST_LEAD_PERIOD | This job populates the UIART.Import_Forecast_Lead_Period table. | The Control.HPF_Preferences and SDL.Forecast_Group tables are required. |
| 13_DELETE_IDM_FCST_CNTRL_DATA | This job deletes records from the ABT.IDM and ABT.Fcst_Control_Data tables to ensure that new values are used in the next incremental run. | The ABT.IDM_Facility_Item, Control.Fcst_Control_Data and Scratch.Forecast_Gr_Fac_Item_Fcst_Gr tables are required. |

## *Forecasting and Inventory Optimization Integration Jobs*

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

***Table A2.10*** *Forecasting and Inventory Optimization Integration Jobs*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 01_**<BASE_PERIOD>**_CREATE_FORECAST_OUTFOR_ART | These jobs combine all the tables for the forecast groups for the corresponding base period from the ABT library into a single table ABT.**<BASE_PERIOD>**_Forecast_Outfor_ART. | The following tables are required:<br>• ABT.**<BASE_PERIOD>**_Forecast_Outfor_ART<br>• SDL.Forecast_Group<br>• ABT.**<BASE_PERIOD>**_Forecast_Outfor_ART**<N>** |
| 01_AGGREGATE_FCST_ART | This job aggregates data depending on the base period and populates the corresponding ABT (ABT.WK_Aggregated_Forecast_ART, ABT.MTH_Aggregated_Forecast_ART, ABT.QTR_Aggregated_Forecast_ART) | The tables ABT.WK_Forecast_Outfor_ART, ABT.MTH_Forecast_Outfor_ART, and ABT.QTR_Forecast_Outfor_ART are required. |
| 02_AGGREGATE_FCST_ART_FROM_NPF | This job aggregates data from the input tables depending on the base period and populates the following tables:<br>• ABT.Day_Aggregated_Forecast_ART<br>• ABT.WK_Aggregated_Forecast_ART<br>• ABT.MTH_Aggregated_Forecast_ART<br>• ABT.QTR_Aggregated_Forecast_ART | The following tables are required:<br>• ABT.NPF_DMD_Less_HIS_AGGR_ABT<br>• TSDB.No_History_Timeseries_Detail<br>• SDL.Facility_Item_Demand<br>• SDL.Facility_X_Item<br>• Control.FCST_Offdays_Peakdays_List |

# 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 A2.11*** *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:<br>• SDL.Network_Model<br>• SDL.Item<br>• SDL.Item_Price<br>• SDL.Facility_X_Item<br>• SDL.Route<br>• SDL.Network_X_Route<br>• SDL.Route_X_Item<br>• SDL.Facility_Item_X_Network<br>• SDL.Item_Succession<br>• SDL.BOM |
| 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:<br>• SDL.Network_Model<br>• SDL.Item<br>• SDL.Facility_X_Item<br>• SDL.Route<br>• SDL.Network_X_Route<br>• SDL.Route_X_Item<br>• SDL.Facility_Item_X_Network<br>• SDL.Item_Succession<br>• SDL.BOM |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 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:<br>• SDL.Network_Model<br>• SDL.Item<br>• SDL.Item_Price<br>• SDL.Facility_X_Item<br>• SDL.Facility_Item_X_Network<br>• SDL.Item_Succession<br>• SDL.Back_Order_Summary<br>• SDL.BOM<br>• ABT.**<BASE_PERIOD>**_Aggregated_Forecast_ART<br>• DIM.Time_Dim |
| 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.<br>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:<br>• SDL.Network_Model<br>• SDL.Item<br>• SDL.Facility_X_Item<br>• SDL.Facility_Item_Inventory<br>• SDL.Pipeline_Inventory<br>• SDL.Item_Succession<br>• SDL.BOM<br>• 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 A2.12*** *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:<br>• ABT.MIRP_**<BASE_PERIOD>**_Node_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_ARC_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_Demand_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_Inventory_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_OPT_Message_ART<br>• ABT.MIRP_**<BASE_PERIOD>**_OUT_Beforetrans | The following tables are required:<br>• ABT.IO_**<BASE_PERIOD>**_Node_Data_ABT<br>• ABT.IO_**<BASE_PERIOD>**_ARC_Data_ABT<br>• ABT.IO_**<BASE_PERIOD>**_Demand_Data_ABT<br>• ABT.IO_**<BASE_PERIOD>**_Inventory_Data_ABT<br>• Control.IO_Batch_Attributes |
| 02_TRANSSHIPMENT | This job runs the transshipment process and populates the following tables:<br>• TRANS_**<BASE_PERIOD>**_Node_Data_ABT<br>• TRANS_**<BASE_PERIOD>**_Nodearc_Data_ABT<br>• TRANS_**<BASE_PERIOD>**_Alertdata_IN_ABT<br>• TRANS_**<BASE_PERIOD>**_Alertdata_Type_ABT<br>• Transshipment_**<BASE_PERIOD>**_ART<br>• Transshipment_Cost_**<BASE_PERIOD>**_ART<br>• Transshipment_**<BASE_PERIOD>**_Summary_ART<br>• TRANS_**<BASE_PERIOD>**_Shipin_Period_Summary<br>• TRANS_**<BASE_PERIOD>**_Shipout_Summary_ART<br>• TRANSSHIP_**<BASE_PERIOD>**_Shipin_Summary | The following tables are required:<br>• ABT.MIRP_**<BASE_PERIOD>**_Node_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_ARC_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_Demand_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_Inventory_Data_ABT<br>• 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:<br>• ABT.MIRP_**<BASE_PERIOD>**_Node_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_ARC_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_Demand_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_Inventory_Data_ABT<br>• ABT.MIRP_**<BASE_PERIOD>**_OUT_Beforetrans<br>• TRANS_**<BASE_PERIOD>**_Shipin_Period_Summary<br>• 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 A2.13*** *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. |
| 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 A2.14*** *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:<br>• ACTUAL_LEAD_TIME<br>• TARGET_LEAD_TIME<br>• LEAD_TIME_MAX<br>• 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:<br>• ACTUAL_SL_DOWNSTREAM<br>• 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:<br>• ACTUAL_SL_UPSTREAM<br>• 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. |
| 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. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 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:<br>• Actual inventory quantity<br>• Actual inventory cost<br>• 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:<br>• Inventory cost<br>• Reorder level<br>• Order-up-to level<br>• 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. |
| 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:<br>• SL_LB<br>• SL_UB<br>• Demand_UL<br>• 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 A2.15*** *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:<br>• LT_ACT<br>• LT_TGT<br>• DEMAND_ACT<br>• DEMAND_FORECAST<br>• SL_UPSTREAM_ACT<br>• SL_UPSTREAM_TGT<br>• SL_DOWNSTREAM_ACT<br>• SL_DOWNSTREAM_TGT<br>• COST_ACT<br>• COST_TGT | The following tables are required:<br>• ABT.IO_<BASE_PERIOD>_NODE_DATA_ABT<br>• UIART.GET_LEAD_TIME_act_tgt<br>• UIART.SL_DOWNSTREAM<br>• UIART.GET_SL_UPSTREAM<br>• UIART.GET_DEMAND_ACT<br>• UIART.GET_DEMAND_TGT<br>• UIART.GET_INVENTORY_COST_ACT<br>• UIART.GET_INVENTORY_COST_TGT<br>• ABT.MIRP_<BASE_PERIOD>__Predict_KPI_Hist_ART |
| 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. |

| Job Name | Job Description | Input Requirements |
|---|---|---|
| 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:<br>• LT_ACT<br>• LT_TGT<br>• DEMAND_ACT<br>• DEMAND_FORECAST<br>• SL_UPSTREAM_ACT<br>• SL_UPSTREAM_TGT<br>• SL_DOWNSTREAM_ACT<br>• SL_DOWNSTREAM_TGT<br>• COST_ACT<br>• 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 A2.16*** *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:<br>• LT_UP_CNTRL_LIM<br>• LT_LW_CNTRL_LIM<br>• DEMAND_UP_CNTRL_LIM<br>• DEMAND_LW_CNTRL_LIM<br>• SL_UPSTREAM_UP_CNTRL_LIM<br>• SL_UPSTREAM_LW_CNTRL_LIM<br>• SL_DOWNSTREAM_UP_CNTRL_LIM<br>• SL_DOWNSTREAM_LW_CNTRL_LIM | The following tables are required:<br>• UIART.GET_LEAD_TIME_min_max<br>• UIART.GET_KPI_min_max_period<br>• 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 framework data 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<br>`spoid = 0`<br>. | 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:<br>• DIM.Item_Dim<br>• DIM.Facility_Dim<br>• SDL.Lookup_Detail<br>• UIART.IO_Metrics_ART_Temp<br>• TSDB.IO_Metrics_Alertsettings<br>• SDL.Employee<br>• 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:<br>• SDL.Facility_Item_X_Network<br>• SDL.Vendor_Facility_Item<br>• UIART.IO_Metrics_ART_No_Vendor_Details<br>• 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.<br>• %IO_METRICS_ITEM_HIER_TABLES<br>• %IO_METRICS_FACILITY_ALL<br>• IO_METRICS_NETWORK_FACILITY<br>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 A2.17*** *The Job for the Scenario Development Workspace*

| Job Name | Job Description | Input Requirements |
|---|---|---|
| SCENARIO_UI_TABLES | This job loads the following target output tables:<br>• UIART.Facility_Face<br>• UIART.Item_Vendor<br>• UIART.I_X_F_ATTR<br>• UIART.Item_Buyer_List<br>• UIART.Buyers<br>• UIART.Vendors<br>These tables are required by the Scenario Development workspace with the SDL and ABT source tables. | The following tables are required:<br>• SDL.Facility_X_Item<br>• SDL.Facility_Item_X_Network<br>• SDL.Vendor_Facility_Item<br>• SDL.Employee<br>• SDL.Vendor<br>• ABT.IO_<BASE_PERIOD>_Node_Data_ABT |

## 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 A2.18*** *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. |
| 02_CREATE_SKU_BUCKETS | This job classifies the facility and item pairs into the following categories for the Order Suggestions workspace:<br>• Normal<br>• Overstock<br>• Primary<br>• Primary and Alternative<br>• Partial<br>• Incomplete<br>The job populates the following tables from the UIART library:<br>• Order_Bucket_Normal_Stock<br>• Order_Bucket_Over_Stock<br>• Order_Bucket_Low_Stock<br>• Order_Bucket_Primary<br>• Order_Bucket_Primary_Alternate<br>• Order_Bucket_Partial<br>• Order_Bucket_Incomplete<br>• Order_Bucket_Type | The input and output tables that are created by the Inventory Optimization process jobs and the following tables are required:<br>• SDL.Lookup_Detail<br>• SDL.Route_Type_Ref<br>• SDL.Vendor<br>• UIART.Item_ATTR<br>• UIART.I_X_F_ATTR<br>• UIART.Facility_ATTR<br>• UIART.Buyers<br>• 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:<br>• SDL.Lookup_Detail<br>• SDL.Route_Type_Ref<br>• SDL.Vendor<br>• UIART.Item_ATTR<br>• UIART.I_X_F_ATTR<br>• UIART.Facility_ATTR<br>• UIART.Buyers<br>• TSDB.IO_Metrics_Allsettings<br>• 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 on the Order Suggestions workspace. | |
| 06_POPULATE_ORDER_RESULT | This job populates the UIART.Order_Result table that is required for displaying information on 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 on 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 such as projected transfer cost, projected holding cost, projected penalty cost, and so on, for displaying information on the Order Suggestions workspace. | |
| 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 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 A2.19*** *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

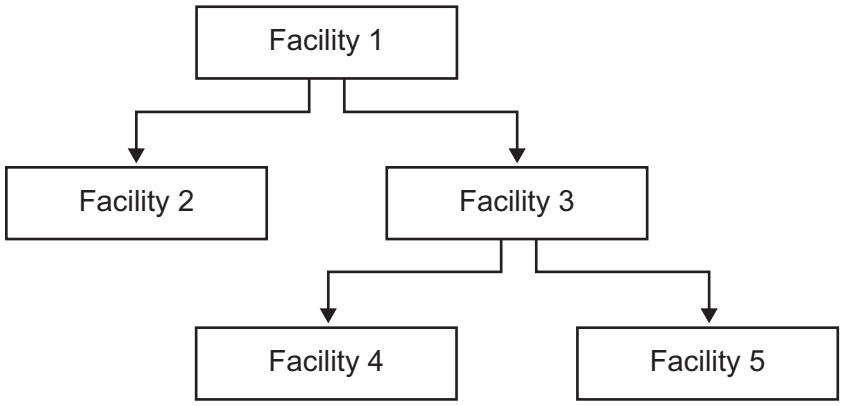
If the lead time actual value is less than the lead time minimum value, then lead time error low = 1, else the value is 0.

## 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

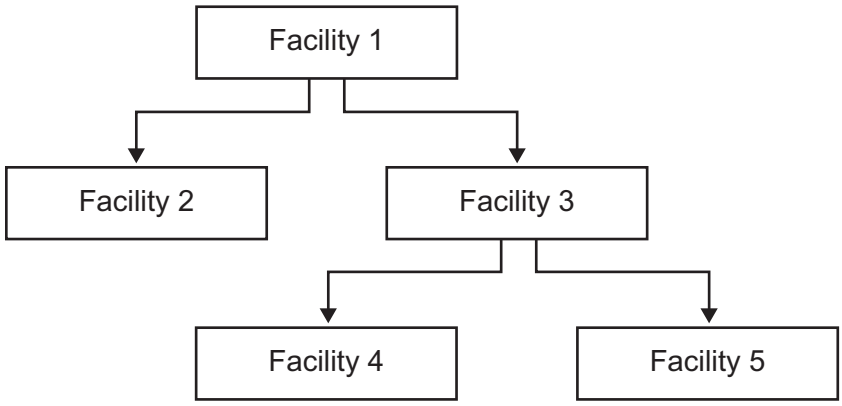
If the downstream service level actual value is less than the downstream service level lower control limit, then the downstream service level error low value is stored as 1, else 0.

## 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 = Fixed ordering cost + Total transfer cost from primary channel and alternate channels

**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<br>• Internal facility service level analysis<br>• 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<br>• Ad hoc analysis |
| On-Hand Holding Cost | On-hand holding cost = On-hand mean * Holding cost amount | • Internal facility service level analysis<br>• 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<br>• Internal facility service level analysis<br>• Ad hoc analysis |

# Glossary

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

**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.

**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.

**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.

**replenishment policy**
a set of guidelines that determine the quantities for orders that are placed to restock inventory.

**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

## W