SAS® Forecast Analyst Workbench 5.1

Data Administration Guide, Second Edition

SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2013. *SAS®Forecast Analyst Workbench 5.1: Data Administration Guide, Second Edition*. Cary, NC: SAS Institute Inc.

SAS® Forecast Analyst Workbench 5.1: Data Administration Guide, Second Edition

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

Printing 2, November 2013

Contents

About This Book

Audience

SAS Forecast Analyst Workbench is designed for the following users:

- Administrators responsible for setting up and maintaining the application environment.
- Data administrators responsible for data management.
- Business users (including planners, forecast analysts, and advanced forecast analysts) responsible for analyzing the forecasted data and making decisions based on that data.

This document focuses on explaining the data management tasks that you can perform for SAS Forecast Analyst Workbench. As a data administrator of SAS Forecast Analyst Workbench, you might also be required to work on SAS Financial Management and SAS Inventory Optimization solutions.

Accessibility

For more information about the accessibility of SAS Forecast Analyst Workbench, see the *SAS Forecast Analyst Workbench: User’s Guide*.

For more information about the accessibility of any of the other products mentioned in this book, see the documentation for that product.

Recommended Reading

- *SAS Financial Management: User's Guide*
- *SAS Solutions Services: Data Administration Guide*
- *SAS Inventory Optimization: User's Guide*
- *SAS Inventory Optimization: Data Administration Guide*
- *SAS Data Integration Studio: User's Guide*
- *SAS Forecast Analyst Workbench: System Administration Guide*
- *SAS Forecast Analyst Workbench: User's Guide*

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

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

Chapter 1

Introduction to SAS Forecast Analyst Workbench

Introduction to SAS Forecast Analyst Workbench

SAS Forecast Analyst Workbench is a component of SAS Demand-Driven Forecasting, the demand planning module of the SAS Demand-Driven Planning and Optimization suite.

SAS Forecast Analyst Workbench helps organizations track, monitor, and predict demand for products and services. SAS Forecast Analyst Workbench enables organizations to gain instant visibility and understanding of the demand for products without needing to rely on personal judgments. SAS Forecast Analyst Workbench is empowered with the capabilities of SAS solutions: analytics, data integration, and business intelligence.

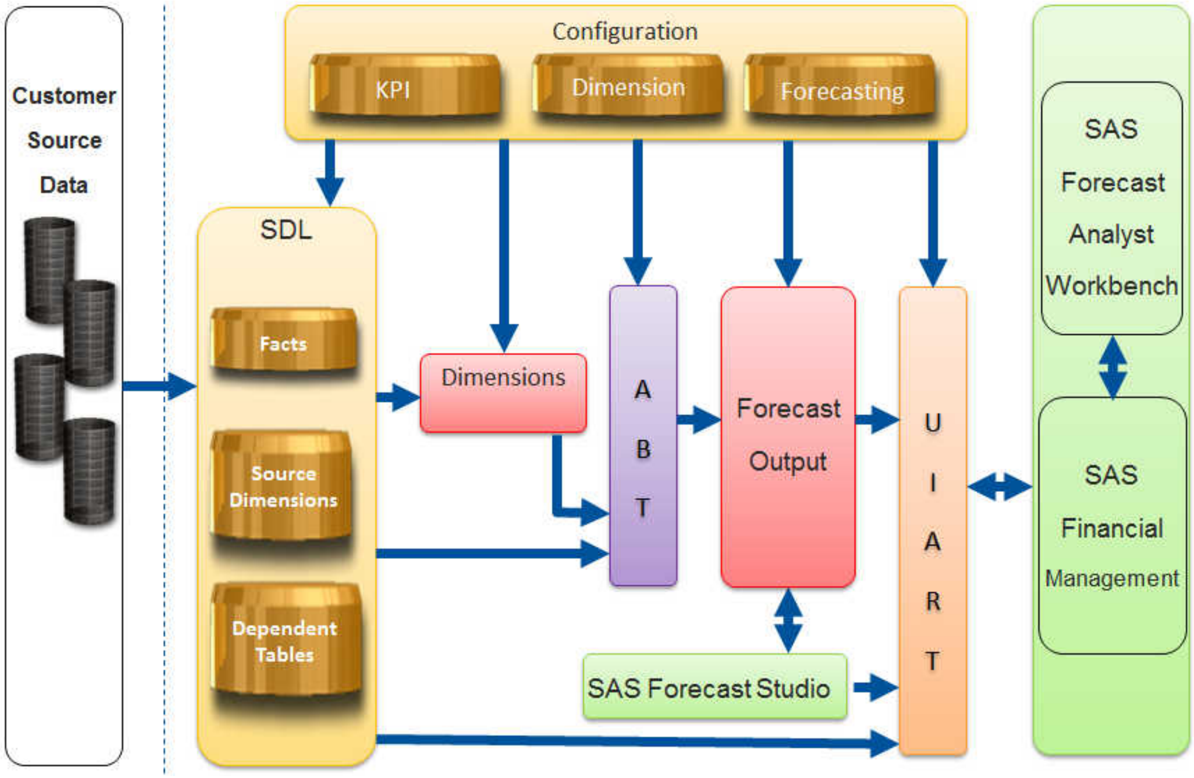
Your main objective as a SAS Forecast Analyst Workbench data administrator is to supply data to the application. As a data administrator, you work with the following kinds of data:

- data that must be supplied initially in order to get the software working
- data that must be supplied periodically so that the software can produce timely output

Loading and Accessing Data

The data moves from its source, through the solution data layer, to the user interface analytical result tables (UIART). This section describes the flow of the data throughout the SAS Forecast Analyst Workbench environment. The following figure shows the data flow.

Figure 1.1 Data Flow through the Environment



The complete path that the data follows consists of the following main steps:

1. Extract the data from your source systems and load it into the solution data layer. In order to load the data in solution data layer, perform the following tasks:

   a. After the System Administrator deploys SAS Forecast Analyst Workbench and enters information in the configuration parameters, run the configuration jobs. The configuration jobs create the configuration tables.

   b. Enter information in the configuration tables to meet your business requirements and data availability.

   c. After the you update the configuration tables, run the initial jobs. The initial jobs use the information that you entered in the configuration tables to create the solution data layer tables. You must map the solution data layer tables with your source system tables and load the data.

      You must update the solution data layer tables whenever the source system is updated or refreshed.

   For more information about running configuration jobs, see "Running Configuration Jobs" on page 8.

   For more information about running initial jobs, see "Running Initial Jobs" on page 9.

   For more information about loading information in solution data layer, see "Loading Data into the Solution Data Layer" on page 14.

2. Run the ongoing ETL jobs to perform the following tasks:

   - load dimension tables with the latest data from the solution data layer

- load user interface analytical result tables (UIART) with the latest data from solution data layer

   Whenever solution data layer is refreshed or updated, update the dimension and UIART tables by running the ongoing jobs. After the dimension tables and UIART are loaded, the user interface for SAS Forecast Analyst Workbench is ready to use.

   For more information about running ongoing jobs, see "Running Ongoing Jobs" on page 17.

3. Use SAS Forecast Analyst Workbench to create plans, diagnose them, and perform the activities related to forecasting and planning. The following list explains how the data flows through the solution when you are working in the user interface of SAS Forecast Analyst Workbench:

   - when you create plans, SAS Forecast Analyst Workbench extracts data from the solution data layer and configuration libraries, and loads it in the analytical base tables (ABTs). SAS Forecast Analyst Workbench also creates plan-specific folders in the path that you specify in the configuration parameters. The data that is related to the plan is loaded in these folders.

     For more information about configuration parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.

   - when you diagnose plans, SAS Forecast Analyst Workbench extracts the data from ABTs to use for high-performance forecasting processes, and then loads the forecasted data into the forecasting tables

   - when you integrate the plans with SAS Forecast Studio, the data is provided to SAS Forecast Studio and the output is loaded in the library that you specified in the configuration parameters

     For more information about defining configuration parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.

   - when you integrate the plans with SAS Financial Management, the plan level data is loaded in SAS Financial Management to use for performing consensus planning.

SAS Forecast Analyst Workbench organizes the files in different libraries according to the subject areas that they belong to. SAS Forecast Analyst Workbench contains the following libraries:

ABT
created in the form that is required for high-performance forecasting-based procedures. The ABTs are created by transforming, filtering, and aggregating the data that is present in the solution data layer tables. The ABTs are used as input to the analytical procedures.

CONFIG
stores tables that contain configuration-related information.

DIM
stores dimension tables that are loaded with the latest data from the solution data layer tables.

SCRATCH
stores intermediate or temporary tables that are used in various processes.

SDL
stores the dimensions, dimension-dependent tables, and facts data. These tables need to be loaded with customer-specific source system data.

UIART
stores application tables. These tables contain information that is required to be displayed in the user interface of SAS Forecast Analyst Workbench. These tables can reside in a MySQL database.

CONTROL
stores tables that contain control-related information that is used to create ABTs.

PLANEXPL
stores the dimension and fact data for each plan. This information is used to display the KPI values for the dimension that the user selected in the Explore Demand view of the application.

DDF_MR
stores user-defined custom models for a catalog. The catalogs are plan-specific.

NPF
stores tables that are specific to forecasting new products.

Chapter 2
ETL Tasks

Overview of ETL Jobs

After you complete the post-installation tasks that are described in the *SAS Forecast Analyst Workbench: System Administration Guide*, you can perform the following tasks in order to prepare SAS Forecast Analyst Workbench data:

1. run configuration jobs
2. update configuration tables
3. run initial jobs
4. load data from the customer source database into solution data layer
5. run ongoing jobs
6. authorize users to access data

For more information about general data flow, see “Loading and Accessing Data” on page 1.

Configuring Parameters

After SAS Forecast Analyst Workbench is installed, the system administrator must configure the parameters that are related to SAS Forecast Analyst Workbench. The system administrator can configure parameters related to ETL, the transactional database, SAS Forecast Studio, and SAS Financial Management. Configure these parameters to meet your business requirements.

For more information about configuration parameters, see the *SAS Forecast Analyst Workbench 5.1: System Administration Guide*.

Preparing SAS Forecast Analyst Workbench Data

Overview of Preparing Data

Data in SAS Forecast Analyst Workbench moves from its source (a customer database), through the solution data layer (SDL), to a destination data mart called user interface analytical result tables (UIART).

The sources of the data can be transactional systems or databases that are outside the SAS Forecast Analyst Workbench environment. The table structures in the SDLs are created based on the information that you provide in the configuration tables. You must load the data into the solution data layer from the source system (the customer system). This process is ongoing. You must run the UIART jobs whenever the source system (customer database) is updated.

The predefined configuration tables are located on the **Folders** tab in SAS Data Integration Studio at the following location:

```
/Products/SAS Forecast Analyst Workbench/Forecast Analyst
5.1/FAW/CONFIGURATION
```

Run Jobs to Load Data

When you are running the ETL jobs, the following prerequisite applies:

- You must be a member of the ETL Users group that you created for SAS Forecast Analyst Workbench.

  For more information about the ETL Users group, see the *SAS Forecast Analyst Workbench: System Administration Guide*.

Perform the following steps to prepare the data for SAS Forecast Analyst Workbench:

1. Run the configuration jobs that are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/CONFIGURATION/CONFIG_JOBS/01_CREATE_CONFIG_TABLES`**

   You must run the configuration jobs only once immediately after you deploy SAS Forecast Analyst Workbench. The configuration tables (for example, DIM_TABLE_LIST, KPI_CONFIG, or CREATE_EVENT) are created when you run the configuration jobs. After the configuration tables are created, enter information that meets your business requirements in the configuration tables.

For more information about running configuration jobs, see "Running Configuration Jobs" on page 8.

2. Run the initial one-time jobs that are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/CONFIGURATION/CONFIG_JOBS/02_INITIAL_ONE_TIME_JOBS`**.

   You must run the initial one-time jobs only once immediately after you deploy SAS Forecast Analyst Workbench and edit the configuration tables. After you run the initial one-time jobs, the following steps are performed:

   a. The table structures in the SDL are created. The SDL tables are created by the initial jobs based on the information that you entered in the configuration tables.

   b. The time period and time dimension data is created based on the information that is specified by the job parameters. You can modify the job parameters.

      For more information about job parameters, see "Job Parameters" on page 83.

   c. The UIART table structures are created and the forecasting information is loaded into the required UIART tables (for example, FORECAST_INTERVAL, INDEP_AGGRE_METHOD).

   For more information about running initial jobs, see "Running Initial Jobs" on page 9.

3. Run the dimension jobs that are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/JOBS/01_DIMENSION_JOBS`**.

   You must run these jobs whenever the source data loads the new data in solution data layer. This step loads dimension information in the dimension tables. While SAS Forecast Analyst Workbench is loading dimension information, it balances the unbalanced hierarchy levels and handles the ragged hierarchy.

   For more information about loading information into the solution data layer, see "Loading Data into the Solution Data Layer" on page 14.

4. Run the UIART jobs that are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/JOBS/02_UIART_JOBS`**

   You must run these jobs whenever the SDL is refreshed or updated. These jobs populate the data that is required to be displayed in the user interface, along with the following data, while a user is creating a plan:

   - list of KPIs
   - associated dimensions for each KPI
   - hierarchy levels, hierarchy values, and attributes that are associated with each dimension
   - independent variables that are associated with each KPI
   - events
   - product succession and substitution information
   - bill of material information

For more information about running ongoing jobs, see "Running Ongoing Jobs" on page 17.

Each of these steps is described in detail in the following sections. After you have performed all these steps, users can open the user interface of SAS Forecast Analyst Workbench to start forecasting the values.

Running Configuration Jobs

The configuration jobs create configuration tables. You must run the configuration jobs that are in SAS Data Integration Studio in the following sequence:

1. 01_CREATE_DIM_TABLE_LIST
2. 02_CREATE_FCST_CONFIG_TABLES

The configuration jobs are in SAS Data Integration Studio at the following location: **/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/CONFIGURATION/CONFIG_JOBS/01_CREATE_CONFIG_TABLES**

After you run the configuration jobs, the configuration tables are created. Before you can run the initial jobs and the ongoing jobs, you must enter the information in the configuration tables to meet your business requirements. The initial jobs and ongoing jobs use the information that you specify in the configuration tables.

The following table describes the configuration jobs in detail.

Table 2.1 *Configuration Jobs*

Job Name	Job Description	Required Inputs
01_CREATE_DIM_TABLE_LIST	Creates the DIM_TABLE_LIST configuration table. You must enter information in the configuration table to create the dynamic dimensions in the solution data layer library. This job loads the default values in the DIM_TABLE_LIST table. You can update these values. For more information about configuration tables, see "SAS Forecast Analyst Workbench Configuration Tables" on page 21.	None

Job Name	Job Description	Required Inputs
02_CREATE_FCST_CONFIG_TABLES	Creates the following configuration tables in the configuration library. You must provide information in these tables to meet your business requirements. • ANALYSIS_VAR • BATCH_INTERVAL_THREADS • CREATE_EVENT • EVENT_REQUIRED • FORECAST_INTERVAL • HPF_PREFERENCES • INDEP_AGGRE_METHOD • INDEP_VAR_TABLE • KPI_CONFIG • NPF_CONFIG • NPF_MODEL_LOOKUP • NPF_REDUCTION_METHODS For more information about configuration tables, see "SAS Forecast Analyst Workbench Configuration Tables" on page 21.	None

Note: Before you run the initial jobs, you must edit the values in the editable columns of the tables that are created by the configuration jobs.

Running Initial Jobs

Immediately after you deploy SAS Forecast Analyst Workbench, you must run the initial jobs only once. The initial jobs create and load the initial tables. These tables are required before you can load data into the solution.

You can access the initial jobs at the following location in SAS Data Integration Studio: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/CONFIGURATION/CONFIG_JOBS/02_INITIAL_ONE_TIME_JOBS`**

The following table explains the initial jobs in detail. You must run the initial jobs in the sequence in which they appear in the table.

Table 2.2 *Initial Jobs*

Job Name	Job Description	Required Inputs
01_CREATE_DIM_VAR_DISP_LIST	Creates the DIM_VAR_DISP_LIST configuration table. You must enter information in this configuration table to link the column names with the display name for the hierarchy and attribute columns. The display names are shown on the user interface. After you run this job, the default values are loaded into the table. You need to update the default values. For more information about this table, see "DIM_VAR_DISP_LIST" on page 24. *Note:* You must edit the values in the editable column before you run the ongoing jobs.	CONFIG.DIM_TABLE_LIST
02_CREATE_SDL_TABLES	Creates table structures for the tables in the SDL. After you run this job, valid dimension structures are created by obtaining information from the DIM_TABLE_LIST table. The number of attributes and hierarchy level columns are created in the same way as for the configuration table. *Note:* The solution data layer must be populated with customer data before you run the ongoing jobs. For more information about loading data into these tables, see "Loading Data into the Solution Data Layer" on page 14.	CONFIG.DIM_TABLE_LIST
03_CREATE_FCST_TABLES	Creates table structures for the KPI and independent variables. The library name and the names that are used to create the table structures are obtained from the CONFIG.KPI_CONFIG table. This job also populates the following control tables: • HPF_PREFERENCE_OPTIONS_CODE • PREV_DIM_HIERARCHY	CONFIG.KPI_CONFIG CONFIG.DIM_TABLE_LIST CONFIG.INDEP_VAR_TABLE
04_CREATE_TIME_PERIOD_DATA	Generates the data for the following tables in the solution data layer: • PERIOD_TYPE • TIME_PERIOD • TIME_PERIOD_ASSOC • TIME_PERIOD_ASSOC_TYPE The job also creates hierarchies by using the associations between the various time period entities.	You can specify parameters such as the start date and the number of years for which data is to be generated as the values of the parameters for the job. For more information about specifying parameters related to this job, see "Job Parameters" on page 83.

Job Name	Job Description	Required Inputs
05_CALENDAR_HIERARCHY1	Reduces the levels in the calender hierarchy1 (from day to week) and stores the output in the SCRATCH.CALENDAR_FLATTEN_HIER1 table. *Note:* After you run this job, you might receive a warning message about data integration. You can ignore this message.	The tables SDL.TIME_PERIOD and SDL.TIME_PERIOD_ASSOC are required.
06_CALENDAR_HIERARCHY2	Reduces the levels in the calender hierarchy2 (from day to month to quarter to year) and stores the output in the SCRATCH.CALENDAR_FLATTEN_HIER2 table. *Note:* After you run this job, you might receive a warning message about data integration. You can ignore this message.	The tables SDL.TIME_PERIOD and SDL.TIME_PERIOD_ASSOC are required.
07_TIME_DIM	Creates and loads the time dimension table DIM.TIME_DIM.	The tables SCRATCH.CALENDAR_FLATTEN_HIER1 and SCRATCH.CALENDAR_FLATTEN_HIER2 are required.

Job Name	Job Description	Required Inputs
08_CREATE_UIART_INPUT_TABLES	Creates the following tables in the UIART library. These tables are used as input for the user interface of SAS Forecast Analyst Workbench. • DIM_LIST • KPI_LIST • KPI_DIMENSIONS • KPI_INDEP_VAR_LIST • DIM_LVL_METADATA • LEAF_NODE_DATA • DIM_HIERARCHY • ATTRIBUTE_LIST • ATTRIBUTE_VALUES • NODE_ATTRIBUTES • PRODUCT_SUCCESSORS • PRODUCT_SUBSTITUTES • FORECAST_INTERVAL • INDEP_AGGRE_METHOD • PLAN_ANALYSIS_VAR • ANALYSIS_VAR • AUTH_PERMISSION • NPF_MODEL_LOOKUP • NPF_REDUCTION_METHODS • PLAN_SCHEDULING_CONFIG	None

Job Name	Job Description	Required Inputs
09_CREATE_UIART_OUTPUT_TABLES	Creates the following tables in the UIART library. The SAS Forecast Analyst Workbench user interface inserts the data in these tables. • PLAN_DETAILS • PLAN_DIMENSIONS • PLAN_HIERARCHY • PLAN_HIERARCHY_NODE • PLAN_HIER_ATTRIB • PLAN_EVENTS • PLAN_INDEP_VAR • PLAN_HIERARCHY_NODE_SUCCESSOR • PLAN_HIERARCHY_NODE_SUBSTITUTE • PLAN_SET_PARAM • PLAN_TIME_SERIES_SET_PARAM • FS_PROJ_ANALYZE • FS_PROJ_DTLS • FS_PROJ_PROMOTE • FS_PROJ_OPERATION • PLAN_OPERATION • AUTH_IDENTIFY • AUTH_ENTRY • PLAN_RERUN_FORECAST • PLAN_SCENARIO_DETAILS • PLAN_SCENARIO_NODE • SCENARIO_TIMESERIES • PRODUCT_KPI_LIST • PRODUCT_RELATIONS • NPF_PRODUCT_PLAN_MAPPING • NPF_CANDIDATES • NPF_CANDIDATES_ATTRIB For more information about loading data into these tables, see “Running Ongoing Jobs” on page 17.	None

Job Name	Job Description	Required Inputs
10_LOAD_UIART_INPUT_TABLES	Obtains the input from the configuration tables and loads the following UIART tables: • FORECAST_INTERVAL • INDEP_AGGRE_METHOD • KPI_AGGRE_METHOD • ANALYSIS_VAR • AUTH_PERMISSION • NPF_REDUCTION_METHODS • NPF_MODEL_LOOKUP	The following tables are required: • CONFIG.FORECAST_INTERVAL • CONFIG.INDEP_AGGRE_METHOD • CONFIG.ANALYSIS_VAR • CONFIG.NPF_REDUCTION_METHODS • CONFIG.NPF_MODEL_LOOKUP

Note: After you execute the initial one-time jobs, you can load the data into the SDL.

Loading Data into the Solution Data Layer

Overview of the Solution Data Layer

The solution data layer (SDL) is the input data layer for SAS Forecast Analyst Workbench. The table structures for the SDLs are created from the information that you specified in the configuration tables. You specify the following information in the configuration tables:

- the dimensions that you want to use
- the hierarchy levels in each dimension
- dimension attributes
- key performance indicators (KPIs)

SAS Forecast Analyst Workbench contains five core dimensions and two dimensions that you can customize to meet your business requirements. Select **Yes** for the valid flag in the DIM_TABLE_LIST configuration table to use a dimension. You do not need to use all the dimensions. SAS Forecast Analyst Workbench contains the following core dimensions:

- Product
- Store Location
- Customer
- Employee
- Organization

SAS Forecast Analyst Workbench contains the following two dimensions that you can customize to meet your business requirements:

- Custom1
- Custom2

Note: If you select the STORE_LOCATION and PRODUCT dimensions, SAS Forecast Analyst Workbench creates additional columns that are specific to these dimensions. For more information about columns, see "Data Model Diagrams" on page 81.

Note: Based on the dimension that you select, the dimension tables and its dependent tables are created. For example, if you select the PRODUCT dimension, all dependent tables (such as, PRODUCT_SUCCESSION and BOM) are created.

You can specify the number of KPIs that you require and the number of dimensions that will be required for each KPI. For example, you can specify to use two KPIs (such as demand and sales) and the number of dimensions that are dependent on these two KPIs. The table structure is created by obtaining data from the KPI_CONFIG configuration table.

You can map the solution data layer tables to the source data that is in the source system to be in a format that is uniform and complete. The SDL tables ensure that the format of the source data is accurately imported into the tables of SAS Forecast Analyst Workbench.

Structure of the Dimension Tables

The following table provides an example of the columns that will be created for the dimensions in the SDL. The dimension that you select will be used for the table name. For example, if you selected the PRODUCT dimension, the table name will be PRODUCT.

Note: In the following table, <**DIMENSION**> will be replaced by the name of the dimension that you selected. For example, if you select the PRODUCT dimension, the column names are PRODUCT_RK,PRODUCT_ID, and so on.

Table 2.3 *Structure of the Dimension Table*

Column Name	Data Type	Mandatory?	Description
<**DIMENSION**>_RK	Number	Y	The sequence-generated unique key for each business key for that dimension. This name is the value of the leaf hierarchy level.
<**DIMENSION**>_ID	Char	Y	Alphanumeric column where the business key identifier of the dimension is stored. This value must be unique.
<**DIMENSION**>_NM	Char	Y	Value that appears at the leaf-level node of the dimension.
<**DIMENSION**>_DESC	Char	N	Description of the leaf-level node of the dimension.
LANGUAGE_CD	Char	N	Language code that is applicable for the solution. You must use the same language code that you declared in the global parameters. For more information about the global parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.

Column Name	Data Type	Mandatory?	Description
DIMENSION_LVL_RK**n**	Number	Y	Retained key value for the nth level of the dimension.
<**DIMENSION**>_LVL_IDn	Char	Y	Alphanumeric identifier for the nth level of the dimension.
<**DIMENSION**>_LVL_NMn	Char	Y	Name of the nth level of the dimension.
<**DIMENSION**>_LVL_DESCn	Char	N	Description of the nth level of the dimension.
<**DIMENSION**>_ATTRIBn	Char	Y	Different attribute values for the dimension.
VALID_FROM_DTTM	Datetime	Y	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period when the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to **00:00:00:00**.
VALID_TO_DTTM	Datetime	Y	The datetime stamp details for the period until which this record is effective in the warehouse. The time value is set to 1 second before the time of the period that the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to **23:59:59:00**.
PROCESSED_DTTM	Datetime	N	The datetime stamp details for when the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp for the update.

For more information about these columns, see “Data Dictionary” on page 59.

Table Dependency

You must load the solution data layer in the sequence shown in the following table to ensure that all the table dependencies are met.

The following table provides an example that assumes the following scenarios:

- all seven dimensions are used in the implementation
- there is only one fact table, KPI_DEMAND

You can use the loading sequence that is shown in the following table if you have more than one fact table.

Table 2.4 *Table Dependency*

Table Name	Loading Sequence Group
CUSTOM1	1
CUSTOM2	1
CUSTOMER	1
EMPLOYEE	1
ORGANIZATION	1
PRODUCT	1
STORE_LOCATION	1
BOM	2
PRODUCT_SPEC	2
PRODUCT_SUBSTITUTE	2
PRODUCT_SUCCESSION	2
KPI_DEMAND	2

The sequence consists of groups that are specified in the Loading Sequence Group column in the above table. 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 you load tables from the next group.

Running Ongoing Jobs

About Ongoing Jobs

Whenever the customer data is populated and refreshed in the solution data layer library, you must run the ongoing jobs. The ongoing jobs consist of a dimension job and UIART jobs.

Dimension Job

You must run the dimension job that is in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/JOBS/01_DIMENSION_JOBS`**

The following table explains the dimension job.

Table 2.5 Dimension Job

Job Name	Job Description	Required Inputs
01_LOAD_DIMENSION_TABLES	Extracts the valid records from source data layer tables and loads the dimensions in the DIM library. This job also handles the unbalanced hierarchy.	All valid (selected) dimensions from the source data layer are loaded.

UIART Jobs

The UIART jobs are in SAS Data Integration Studio at the following location: `/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FAW/JOBS/02_UIART_JOBS`

The following table describes the UIART jobs.

Table 2.6 UIART Jobs

Job Name	Job Description	Required Tables
01_LOAD_PLAN_INPUT_TABLES	Populates the following tables: • UIART.DIM_LIST • UIART.KPI_LIST • UIART.KPI_DIMENSIONS • UIART.KPI_INDEP_VAR_LIST • UIART.DIM_LVL_METADATA • UIART.PRODUCT_SUCCESSOR • UIART.PRODUCT_SUBSTITUTE	• CONFIG.DIM_TABLE_LIST • CONFIG.DIM_VAR_DISP_LIST • CONFIG.KPI_CONFIG • CONFIG.INDEP_VAR_TABLE • SDL.PRODUCT_SPEC • SDL.PRODUCT_SUCCESSION • SDL.PRODUCT_SUBSTITUTE • DIM.PRODUCT_DIM • UIART.KPI_AGGRE_METHOD
02_LOAD_LEAF_NODE_DATA	Populates the information for the leaf-level node for the PRODUCT dimension in the UIART.LEAF_NODE_DATA table.	• CONFIG.DIM_TABLE_LIST • DIM.PRODUCT_DIM
03_LOAD_DIM_HIERARCHY	Populates hierarchies for each dimension and populates the following tables: • CONTROL.PREV_DIM_HIERARCHY • UIART.DIM_HIERARCHY	• CONFIG.DIM_TABLE_LIST • tables of all valid dimensions
04_LOAD_ATTRIB_LIST	Populates the list of all attributes that are associated with each dimension in the table UIART.ATTRIBUTE_LIST.	• CONFIG.DIM_TABLE_LIST • CONFIG.DIM_VAR_DISP_LIST • SDL.PRODUCT_SPEC

Job Name	Job Description	Required Tables
05_LOAD_ATTRIB_VALUES	Populates the values of attributes for all valid dimensions in the table UIART.ATTRIBUTE_VALUES.	• CONFIG.DIM_TABLE_LIST • SDL.PRODUCT_SPEC • UIART.ATTRIBUTE_LIST
06_LOAD_NODE_ATTRIBUTES	Populates attribute values that are associated with the nodes for all valid dimensions UIART.NODE_ATTRIBUTES.	• CONFIG.DIM_TABLE_LIST • UIART.ATTRIBUTE_LIST
07_LOAD_EVENTS	Populates the PLAN_CREATE_EVENT table in the UIART.	• CONFIG.CREATE_EVENT • CONFIG.EVENT_REQUIRED

Authorizing Users to Access Data

After you run the initial jobs and ongoing jobs, the system administrator must provide users with access to the solution data. Whenever the system administrator adds a new user or whenever you add new data in the solution and then run the ongoing jobs, the system administrator needs to provide users with access to the newly added data.

The system administrator authorizes users to access the data in SAS Management Console. For more information about authorizing users to access the data, see the *SAS Forecast Analyst Workbench: System Administration Guide*.

Chapter 3

SAS Forecast Analyst Workbench Configuration Tables

Overview of the Configuration Tables

This chapter describes the configuration tables that are created after you run the configuration jobs. You must edit the configuration tables to meet your business requirements. The initial one-time jobs use the information that is in these tables to create the solution data layer (SDL).

DIM_TABLE_LIST

The DIM_TABLE_LIST table is the first configuration table that is created after you run the ETL jobs. This table is created with the default values. Keep the following information in mind when you enter information in the columns that are editable:

- the valid dimensions for the configuration
- the dimension display name
- the hierarchy levels that the dimension should contain
- the common attributes that the dimension supports
- the valid language code
- the description of the dimension

This table helps create the dynamic data model for the SDL. The table also helps create SDL tables that contain the mandatory columns with respect to the customer source data.

Table 3.1 *Description of the DIM_TABLE_LIST Table*

Column Name	Description	Data Type	Editable?	Sample Value
DIM_RK	Unique serial number of the dimension.	Number	No	1
DIM_ID	Unique identifier for the dimension.	Char (32)	No	PRODUCT_ID
DIM_NM	Unique name for the dimension.	Char (32)	No	PRODUCT
DIM_ADK	Additional description key column name for the dimension. This column is required in order to integrate SAS Forecast Analyst Workbench with SAS Financial Management.	Char (32)	No	PRODUCT
PRIMARY_RK_NM	Unique name of the retained key column of the dimension.	Char (40)	No	PRODUCT_RK

Column Name	Description	Data Type	Editable?	Sample Value
VALID_DIM_FLG	Flag that indicates whether the dimension is applicable. Valid values are Y or N. These values must be in sync with the value that is specified in the configuration parameter for short names, GL_SHORT_YES or GL_SHORT_NO. For more information about the configuration parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.	Char (3)	Yes	Y
DIM_DESC	Description of the dimension. This column is optional.	Char (255)	Yes	Information about all items that include finished goods, assemblies, subassemblies, and parts.
DIM_DISPLAY_NM	The display name of the dimension. This name is shown on the user interface of SAS Forecast Analyst Workbench.	Char (40)	Yes	ITEM
HIER_LVL_NO	Number of hierarchy levels that are required for the dimension. If the dimension does not have any hierarchy levels, you must enter `1` in order to create a single hierarchy level for that dimension. The value that you specify in this column determines the number of hierarchy levels that are created in the dimension table of the SDL.	Number	Yes	6
COMMON_ ATTRIB_NO	Number of attributes that are required for the dimensions. If attributes are not required, enter `0`. The value that you specify in this column determines the number of attributes that are created in the dimension table of the SDL.	Number	Yes	6

Column Name	Description	Data Type	Editable?	Sample Value
LANGUAGE_CD	Required language codes. If you have purchased the license for performing consensus planning, the value that you enter in this column must match the value that is in the SAS Financial Management Language Code table. Otherwise, you can leave this column blank.	Char (10)	Yes	en

The following table is an example of the DIM_TABLE_LIST table with important columns.

Table 3.2 *Example of the DIM_TABLE_LIST Table*

DIM_RK	DIM_NM	VALID_DIM_FLG	DIM_DISP_NM	HIER_LVL_NO	COMMON_ATTRIB_NO
1	PRODUCT	Y	PRODUCT	6	6
2	STORE_LOCATION	Y	STORE_LOCATION	5	1
3	CUSTOMER	Y	CUSTOMER	1	1
4	EMPLOYEE	Y	EMPLOYEE	4	0
5	ORGANIZATION	Y	ORGANIZATION	4	1
6	CUSTOM1	Y	CUSTOM1	3	0
7	CUSTOM2	Y	CUSTOM2	4	2

DIM_VAR_DISP_LIST

The DIM_VAR_DISP_LIST table stores the information about the relation between the physical column name and the display name. You can use your business terminology for the display names.

SAS Forecast Analyst Workbench displays all the labels that are configured in this table in the user interface.

Table 3.3 *Description of the DIM_VAR_DISP_LIST Table*

Column Name	Description	Data Type	Editable?	Sample Value
DIM_RK	Unique serial number of the dimension.	Number	No	1

Column Name	Description	Data Type	Editable?	Sample Value
DIM_NM	Unique name for the dimension.	Char (40)	No	PRODUCT
DIM_ID	Unique identifier for the dimension.	Char (32)	No	PRODUCT_ID
LVL_ATTRIB_IND	Indicates whether the variable is a hierarchy level or an attribute. Enter `1` to indicate that the variable is a hierarchy level. Enter `0` to indicate that the variable is an attribute.	Number	No	1
HIER_ORDER_NO	Specifies the order of the hierarchy level for the variable of the dimension.	Number	No	1
LEAF_NODE_IND	Indicates whether the variable of the dimension is a leaf-level node for the hierarchy level columns. Enter `1` to indicate that the variable of the dimension is a leaf-level node.	Number	No	0
VARIABLE_NM	Name of the attribute or hierarchy level of the dimensions. This value is the column name of the dimension table.	Char (40)	No	PRODUCT_ LVL_NM_1
VARIABLE_DESC	Description of the variable attributes or of the dimension level. This column is optional.	Char (255)	Yes	Stores the first level of the product hierarchy
DISPLAY_NM	Stores the names of the attributes and hierarchy columns of the dimensions. This name is displayed on the user interface. This column is mandatory. By default, this value is the same as the variable name. You must update the values to meet your business requirements.	Char (40)	Yes	ALL PRODUCTS
ATTRIB_ DATATYPE	Indicates the type of data that is in this column if this variable is an attribute. Enter `0` to indicate that the data type is character, `1` to indicate that the data type is an integer, `2` to indicate that it is a double, and `3` to indicate that the data type is a date.	Number	Yes	1
ATTRIB_UOM_CD	The unit of measurement code for the attribute. This column is optional.	Char (10)	Yes	DZ – Dozen

The following table shows an example of the DIM_VAR_DISP_LIST table with important columns.

***Table 3.4** Example of the DIM_VAR_DISP_LIST Table*

DIM_RK	DIM_NM	LVL_ ATTRIB_IND	HIER_ ORDER_NO	LEAF_ NODE_ID	VARIABLE_ NM	DISPLAY_ NM
1	PRODUCT	1	1	0	PRODUCT_ LVL_NM_1	ALL
1	PRODUCT	1	2	0	PRODUCT_ LVL_NM_2	SEGMENT
1	PRODUCT	1	3	0	PRODUCT_ LVL_NM_3	CLASS
1	PRODUCT	1	4	0	PRODUCT_ LVL_NM_4	CATEGORY
1	PRODUCT	1	5	0	PRODUCT_ LVL_NM_5	SUBCATEG ORY
1	PRODUCT	1	6	1	PRODUCT_ NM	PRODUCT NAME
1	PRODUCT	0			PRODUCT_ ATTRIB1	ASSEMBLY FLAG
1	PRODUCT	0			PRODUCT_ ATTRIB2	MAKE OR BUY
1	PRODUCT	0			PRODUCT_ ATTRIB3	MANUFACT URING SOURCE
1	PRODUCT	0			PRODUCT_ ATTRIB4	PLANNER
1	PRODUCT	0			PRODUCT_ ATTRIB5	CONTRIBUT ION LEVEL
1	PRODUCT	0			PRODUCT_ ATTRIB6	LIFE CYCLE STAGE

PLAN_THREAD

The PLAN_THREAD table is used in the forecasting process when the optimization flag is set to Y. Each row in the table contains a plan ID that is specific to each plan. Each row also contains the corresponding number of sessions to be spawned by the SAS Stored Process Server for parallel processing. This table is initially populated with default values. The default THREAD_NUM value for each plan is 1.

Table 3.5 *Description of the PLAN_THREAD Table*

Column Name	Description	Data Type	Editable?	Sample Value
PLAN_ID	Unique identifier for the plan.	Number	No	2
THREAD_NUM	Number of parallel threads to be spawned.	Number	Yes	1

ANALYSIS_VAR

The ANALYSIS_VAR table contains a list of all the analysis variables that are used to measure the KPI values. After you deploy SAS Forecast Analyst Workbench, this table is populated with the following default analysis variables:

- ACTUAL
- PREDICTED
- OVERRIDE
- CONSENSUS
- LAST_CONSENSUS
- FS_OVERRIDE

  Note: The variable FS_OVERRIDE is used in scenario analysis. This variable is loaded into SAS Financial Management only when you select **Configure Analysis Variables** while you are performing consensus planning.

You can also add custom analysis variables in order to perform consensus planning. You can add a row to this table in order to add additional custom analysis variables. You must seed the custom analysis variable to use it in the consensus planning process. For more information about configuring and seeding analysis variables, see Appendix 4, “Seeding in SAS Forecast Analyst Workbench,” on page 85.

Table 3.6 *Description of the ANALYSIS_VAR Table*

Column Name	Description	Data Type	Editable?	Sample Value
ANALYSIS_VAR_RK	Unique number of the analysis variable. This column is mandatory.	Number	No	1
ANALYSIS_VAR_ADK	Unique code that identifies the new dimension in SAS Financial Management. This code is used to integrate SAS Forecast Analyst Workbench with SAS Financial Management. This column is mandatory when you are using consensus planning.	Char (32)	No	ANALYSIS

Column Name	Description	Data Type	Editable?	Sample Value
ANALYSIS_VAR_ ID	Unique identifier for the analysis variable. This value represents the column name in the fact table. This column is mandatory. This column value must be a valid SAS name. For more information about SAS naming conventions, see *SAS Language Reference: Concepts*.	Char (32)	No	LAST_ CONSENSUS
ANALYSIS_VAR_ DESC	Short description of the analysis variable.	Char (255)	Yes	This variable contains the last consensus values
LANGUAGE_CD	A code that identifies the language or locale for names and descriptions. The code must be defined in the CODE_LANGUAGE configuration table. This column is mandatory when you are using consensus planning.	Char (3)	Yes	en
ANALYSIS_VAR_ NM	Name of the analysis variable. This column is mandatory when you are using consensus planning.	Char (40)	No	LAST_ CONSENSUS
ANALYSIS_VAR_ DISPLAY_NM	Name of the analysis variable that is displayed in the user interface. This column is mandatory.	Char (40)	Yes	LAST FINALS

BATCH_INTERVAL_THREADS

The BATCH_INTERVAL_THREADS table is used during the batch process when the optimization flag is set to Y. Each row in the table contains a different periodicity that SAS Forecast Analyst Workbench supports and the corresponding number of sessions to be spawned by the SAS Stored Process Server for parallel processing.

Table 3.7 *Description of the BATCH_INTERVAL_THREADS Table*

Column Name	Description	Data Type	Editable?	Sample Value
PERIODICITY	The time grain of the forecast period. This value must be same as the configuration parameter value that is specified for different base periods (for example, GL_DAY_BASE_PERIOD). For more information about the configuration parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.	Char (255)	No	D
THREAD_NUM	Number of parallel threads to be spawned.	Number	Yes	1

CREATE_EVENT

The CREATE_EVENT table stores information related to events. You can create events in any of the following ways:

- create events in SAS Forecast Studio, and then promote them in SAS Forecast Analyst Workbench using the promote functionality. To create and promote the events:

  1. Create a forecast plan.
  2. Create a SAS Forecast Studio project for that plan.
  3. Open the SAS Forecast Studio project in SAS Forecast Studio.
  4. Create events in the SAS Forecast Studio project.
  5. Refresh the SAS Forecast Studio project in the Forecast Studio Projects category view.
  6. Promote the SAS Forecast Studio project in the Forecast Studio Projects category view.

  These types of events are automatically appended to this table.

- create events by using the HPFEVENTS procedure, and then manually append them to the table.
- create events in a .csv file and import that file into SAS Forecast Analyst Workbench. For details about importing an event by using a .csv file, see the *SAS Forecast Analyst Workbench: User's Guide*.

Note: You should not delete any events from the table because they might have been used in forecast plans.

Before you add events to the CREATE_EVENTS configuration table, see the *SAS High-Performance Forecasting: User's Guide*.

EVENT_REQUIRED

The EVENT_REQUIRED table is related to the CREATE_EVENT table in the configuration library. For each event in the CREATE_EVENT table, a row is present in this table. The row contains the name of the event and an indicator of how the event is included in the model. For example, you can force an event to be used, or you can use the event when it is significant to the model.

You can create the EVENT_REQUIRED table in one of the following ways:

- create events in SAS Forecast Studio, and then promote them in SAS Forecast Analyst Workbench
- create events by using the HPFEVENTS procedure, and then manually add the event names and the indicator of how the event is used in the model in EVENT_REQUIRED table
- create events in a .csv file and import that file into SAS Forecast Analyst Workbench.

For more information, see the *SAS Forecast Analyst Workbench: User's Guide*.

Table 3.8 *Description of the EVENT_REQUIRED Table*

Column Name	Description	Data Type	Editable?	Sample Value
NAME	Name of the event.	Char (40)	Yes If a new event is created in the CREATE_ EVENT table, you should add that event name to this table.	Event1

Column Name	Description	Data Type	Editable?	Sample Value
REQUIRED	Indicates how the event is included in the model. The following values are permitted: • **YES**: Specifies that the events be included in the model as long as the model does not fail diagnosis. • **MAYBE**: Specifies that the events be included in the model as long as the parameters of the events are significant. • **NO**: Specifies that the events be included in the model as long as the parameters of the events are significant and the increment of the value of criterion exceeds a threshold. The default value is **NO**.	Char (32)	Yes If a new event is created in the CREATE_ EVENT table, you should add an indication of how to use that event in the model in this table.	MAYBE

FORECAST_INTERVAL

The FORECAST_INTERVAL table stores the default values of parameters that are related to forecast intervals (such as **DAY**, **WEEK**, **MONTH**, and **QUARTER**). These intervals are displayed in the user interface and are used in the forecasting process.

Table 3.9 *Description of the FORECAST_INTERVAL Table*

Column Name	Description	Data Type	Editable?	Sample Value
FORECAST_ INTERVAL_NO	Forecast interval number.	Number	No	1
FORECAST_ INTERVAL_CD	Forecast interval code. This value must be same as the configuration parameter value that is specified for different base periods (for example, GL_DAY_BASE_PERIOD).	Char (32)	Yes	D
FORECAST_ INTERVAL_DESC	Forecast interval description that is displayed on the user interface.	Char (40)	Yes	DAY
HORIZON	Number of periods to be forecasted.	Number	Yes	12

Column Name	Description	Data Type	Editable?	Sample Value
HISTORY_PERIOD	Number of historical periods to be used in the forecasting process. If a time series contains more periods than this number, the earlier periods are ignored.	Number	Yes	1095
LOW_HISTORY_ THRESHOLD	Least number of periods of historical data that the forecasting process requires in order to generate a statistical forecast.	Number	Yes	175
MINOBS_SEASON	Minimum number of observations that are required in order to perform seasonal adjustments to the demand forecast for the new product. When you edit this column, the values that you provide should be greater than the default values.	Number	Yes	14

HPF_PREFERENCES

The HPF_PREFERENCES table contains a list of options for high performance forecasting procedures, with default values. You can reset the default values by specifying the values in the HPF_PREFERENCES configuration table. If you change a parameter in the HPF_PREFERENCES configuration table, all plans are created with the updated parameter values.

When you create a plan by using the user interface, SAS Forecast Analyst Workbench uses the HPF_PREFERENCES table to create a HPF_PREFERENCES<**`plan_ID`**> table in the configuration library. If you change any parameter in the HPF_PREFERENCES<**`plan_ID`**> table, the values are applied only to the plan that you have specified. The plan_ID value is the unique identifier of the plan and appears in the UIART.PLAN_DETAILS table.

The following table describes the HPF_PREFERENCES table and its columns.

Table 3.10 *Description of the HPF_PREFERENCES Table*

Column Name	Description	Data Type	Editable?
OPTION	Unique keyword for each HPF option.	Char (40)	No
VALUE	Value of the HPF option.	Char (40)	Yes

The following table contains the parameters that are in the HPF_PREFERENCES table. You can edit the parameters for the table where the value of the column EDIT_FROM is

BACK. The parameters that you can edit from the user interface are not shown. You can change the parameters that are related to the user interface from the user interface only.

The following table describes the parameters that can be used in the Value column of the HPF_PREFERENCES table.

Table 3.11 *Parameters for the Value Column*

Parameter Name	Description	Default Value
BASENAME	Prefixes any generated XML specification filename if there are no other contextual base name options.	Diag
DIAG_DELAY_LAG_EVENT	Indicates whether the delay lag for the event is present.	0
DIAG_DELAY_LAG_INPUT	Indicates whether the delay lag for input is present.	0
DIAG_DELAYEVENT	Specifies the delay lag for the events. If this option is not specified, the delay lag for the events is set to `0` by default.	0
DIAG_DELAYINPUT	Specifies the delay lag for the inputs. If this option is not specified, the procedure chooses the delay lag for the inputs.	0
DIAG_ENTRYPCT	Specifies a threshold at which to check the percentage increment of the criterion between two candidate models. The value should be in the range (0,100).	0.1
DIAG_ERROR_SEVERITY	Specifies the severity level of error to be printed. You can specify one of the following values: LOW, MEDIUM, HIGH, ALL, or NONE.	ALL
DIAG_ERROR_STAGE	Specifies the stage of the procedure at which an error message needs to be printed. You can specify one of the following values: PROCEDURELEVEL, DATAPREP, DIAGNOSE, or ALL.	All
DIAG_INCLUDE_IDM	Indicates whether the IDM statement needs to be included.	YES
DIAG_INPUT_ SETMISSSING	Specifies how missing values (actual or accumulated) are assigned in the accumulated time series for variables that are listed in the INPUT statement. You can specify one of the following values: MISSING, AVERAGE | AVG, MINIMUM | MIN, MEDIAN | MED, MAXIMUM | MAX, FIRST, LAST, PREVIOUS | PREP, or NEXT.	MISSING
DIAG_INPUT_TRIMMISS	Specifies how missing values (actual or accumulated) are trimmed from the accumulated time series for variables that are listed in the INPUT statement. You can specify one of the following values: NONE, LEFT, RIGHT, or BOTH.	NONE
DIAG_INPUT_ZEROMISS	Specifies how beginning and ending zero values (actual or accumulated) are interpreted in the accumulated time series for variables that are listed in the INPUT statement. You can specify one of the following values: NONE, LEFT, RIGHT, or BOTH.	NONE

Parameter Name	Description	Default Value
DIAG_INPUTMISSING_PCT	Specifies the size of the missing observation as a percentage of the length of the input time series. If INPUTMISSINGPCT=50, an input time series that has more than 50% missing data is ignored in the model.	10
DIAG_TREND_P	Specifies the autoregressive order for the augmented unit root tests and a seasonality test.	min(5,[T/10])
DIAG_TREND_SIGLEVEL	Specifies the significance level to use as a cutoff value for deciding whether the series needs differencing.	0.01
HPF_ALPHA	Specifies the confidence level size to use in computing the confidence limits in the model selection list files.	0.05
HPF_BACK	Specifies the number of observations that are included before the end of the data.	0
HPF_DIAGNOSE_ INTERMITTENT	Specifies a number greater than 1 that is used to determine whether a time series is intermittent	2
HPF_FCST_ACCUMULATE	Specifies how the data set observations are accumulated within each time period for the variables that are listed in the FORECAST statement. You can specify one of the following values: NONE, TOTAL, AVERAGE | AVG, MINIMUM | MIN, MEDIAN | MED, MAXIMUM | MAX, N, NMISS, NOBS, FIRST, LAST, STDDEV | STD, CSS, or USS.	NONE
HPF_FCST_SETMISSING	Specifies how missing values (actual or accumulated) are assigned in the accumulated time series for variables that are listed in the FORECAST statement. You can specify one of the following values: MISSING, AVERAGE | AVG, MINIMUM | MIN, MEDIAN | MED, MAXIMUM | MAX, FIRST, LAST, PREVIOUS | PREP, or NEXT.	MISSING
HPF_FCST_TRIMMISS	Specifies how missing values (actual or accumulated) are trimmed from the accumulated time series for variables that are listed in the FORECAST statement. You can specify one of the following values: NONE, LEFT, RIGHT, or BOTH.	NONE
HPF_FCST_ZEROMISS	Specifies how beginning zero values, ending zero values, or both (actual or accumulated) are interpreted in the accumulated time series for variables that are listed in the FORECAST statement. You can specify one of the following values: NONE, LEFT, RIGHT, or BOTH.	NONE
HPF_ID_ACCUMULATE	Specifies how the data set observations are accumulated within each time period. You can specify one of the following values: NONE, TOTAL, AVERAGE | AVG, MINIMUM | MIN, MEDIAN | MED, MAXIMUM | MAX, N, NMISS, NOBS, FIRST, LAST, STDDEV | STD, CSS, or USS.	NONE

Parameter Name	Description	Default Value
HPF_ID_ALIGN	Controls the alignment of SAS dates that are used to identify output observations. You can specify one of the following values: BEGINNING | BEG | B, MIDDLE | MID | M, or ENDING | END | E.	BEGINNING
HPF_ID_SETMISSING	Specifies how missing values (actual or accumulated) are assigned in the accumulated time series. You can specify one of the following values: MISSING, AVERAGE | AVG, MINIMUM | MIN, MEDIAN | MED, MAXIMUM | MAX, FIRST, LAST, PREVIOUS | PREP, or NEXT.	MISSING
HPF_ID_TRIMMISS	Specifies how missing values (actual or accumulated) are trimmed from the accumulated time series. You can specify one of the following values: NONE, LEFT, RIGHT, or BOTH.	NONE
HPF_ID_ZEROMISS	Specifies how beginning zero values, ending zero values, or both (actual or accumulated) are interpreted in the accumulated time series. You can specify one of the following values: NONE, LEFT, RIGHT, or BOTH.	NONE
DIAG_INPUT_REQUIRED	Specifies how the input variables be included in the model. You can specify one of the following values: YES, NO, or MAYBE.	NO
DIAG_FORCED_CUSTOM_MODEL	Indicates whether the custom model is required to be forced.	0
DIAG_ARIMAX_METHOD	Specifies the method for choosing the tentative ARMA orders. You can specify one of the following values: ESACF, MINIC, or SCAN.	MINIC
HPFENGINE_INPUT_REQUIRED	Enables a check of inputs to models. The options are YES and NO. If you set this option to **`YES`**, these checks are not performed and no inputs are dropped from a model. If you set this option to **`NO`**, inputs are checked and those with errors or those judged to be collinear are dropped from the model for the current series and task only.	YES
HPFENGINE_INPUTTRIMMISS	Specifies how missing values (either actual or accumulated) are trimmed from the accumulated time series for variables listed in the INPUT statement.	RIGHT
HPF_EXCEPTIONS	Specifies how arithmetic exceptions are handled during the run. The options are CATCH and IGNORE. The option IGNORE states that SAS Forecast Analyst Workbench stops on an arithmetic exception. No recovery is attempted. The option CATCH specifies that SAS Forecast Analyst Workbench skips the generation of diagnostic output for the variable that produces the exception in the current BY group.	CATCH

Parameter Name	Description	Default Value
DIAG_ARIMAX_OUTLIER_FILTER	Chooses a model for outlier detection. The options are FULL and SUBSET. If you set the option to **FULL**, use a full model. If you set the option to **SUBSET**, use a subset model that includes non-seasonal AR and MA filters only.	SUBSET

In the HPF_PREFERENCES<**plan_id**> table, in addition to the parameters that are listed in the previous table, the following parameters are created for each independent variable:

- DIAG_INPUT_REQUIRED<N>
- DIAG_INPUT_SETMISSING<N>
- DIAG_INPUT_TRIMMISS<N>
- DIAG_INPPUT_ZEROMISS<N>

In these parameters, <N> represents the sequence in which the independent variable is selected in the plan.

INDEP_AGGRE_METHOD

The INDEP_AGGRE_METHOD table contains a list of all aggregation methods that are used to aggregate the independent variables. The list contains all the methods that are described in the *SAS High-Performance Forecasting: User's Guide*. Unless a new method is added or changes are made to the existing name of the aggregation method in SAS High-Performance Forecasting, you do not need to modify this table.

Table 3.12 *Description of the INDEP_AGGRE_METHOD Table*

Column Name	Description	Data Type	Editable?	Sample Value
METHOD_RK	Unique number of the aggregation method.	Number	No	2
METHOD_NM	Name of the method.	Number	No	TOTAL

INDEP_VAR_TABLE

The INDEP_VAR_TABLE table is a configuration table that stores information that is related to the independent variables. After you run the ETL jobs, the table structure is created. You must enter independent variable information in this table. Each row of the table belongs to an independent variable.

Note: After you enter the independent variables in this table and those independent variables are used in the plan, you must not edit or change their values except for entries in the INDEP_VAR_DISPLAY_NM column. If you edit the values of the independent variables after they are used in a plan, a mismatch in the existing data might occur.

The names in the INDEP_VAR_NM column are used as the column names in the independent variable that is specific to key performance indicators (KPIs).

Table 3.13 *Description of the INDEP_VAR_TABLE Table*

Column Name	Description	Data Type	Editable?	Sample value
INDEP_VAR_RK	Unique number of the independent variable. This column is mandatory.	Number	No	1
INDEP_VAR_NM	Name of the independent variable. This name must be valid SAS name because it is used as a column name in an independent table. This column is mandatory.	Char (32)	No	Product_Discount
INDEP_VAR_DESC	Short description of the independent variable.	Char (255)	Yes	Discount that our organization wants to give to customers
INDEP_VAR_ DISPLAY_NM	Name of the independent variable, which is displayed in the user interface This column is mandatory.	Char (40)	Yes	Product Discount

KPI_CONFIG

The KPI_CONFIG table stores information about the dependent variables. The dependent variables are the key performance indicators (KPIs) to be forecasted by SAS Forecast Analyst Workbench. After you run the ETL jobs, the KPI_CONFIG table structure is created. Each row in the table belongs to a KPI. The table also stores information about the independent variables that are associated with the KPI.

Note: After you insert the rows in this table and these values are used in the plan, do not edit or change the values except for the values in the KPI_AGGRE_METHOD column. If you edit or change the values, a mismatch in the existing data might occur.

Note: You do not need to specify values in the LIB_INDEP,INDEP_TABLE_NM, and INDEP_VAR_LIST columns if none of the independent variables are associated with the configured KPI.

Table 3.14 *Description of the KPI_CONFIG Table*

Column Name	Description	Data Type	Editable?	Sample Value
KPI_RK	Unique serial number that is associated with the KPI. This column is mandatory.	Number	No	1

Column Name	Description	Data Type	Editable?	Sample Value
KPI_NM	Name of the KPI. This column is mandatory.	Char (40)	No	DEMAND_KPI
KPI_TYPE	Type of the KPI. This column is mandatory.	Char (32)	No	Operational
LIB_NM	Name of the library in which the KPI table that is specified in the KPI_TABLE_NM column is stored. The library name must be a valid SAS library name. This column is mandatory.	Char (32)	No	SDL
KPI_TABLE_NM	Name of the table in which fact data that is associated with the KPI is stored. The table name must be a valid SAS data set name. This column is mandatory.	Char (40)	No	DEMAND_KPI
KPI_VAR_NM	Variable name that is associated with the KPI. The variable name must be a valid SAS variable name. For more information about SAS naming conventions, see *SAS Language Reference: Concepts*. This column is mandatory.	Char (40)	No	Demand_QTY
KPI_AGGRE_ METHOD	Aggregation and accumulation method to be used for the KPI. This method must be a valid method as described in the *SAS High Performance Forecasting: User's Guide*. You can change this method for different plans. This column is mandatory.	Char (32)	Yes	Total
DECIMAL_ ROUND_OFF_VAL	Number of digits after the decimal point to be displayed in the user interface. This column is mandatory.	Number	No	2
DIM1_IND	Indicates whether the dimension with a DIM_RK value equal to 1 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	1

Column Name	Description	Data Type	Editable?	Sample Value
DIM2_IND	Indicates whether the dimension with a DIM_RK value equal to 2 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	1
DIM3_IND	Indicates whether the dimension with a DIM_RK value equal to 3 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	1
DIM4_IND	Indicates whether the dimension with a DIM_RK value equal to 4 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	0
DIM5_IND	Indicates whether the dimension with a DIM_RK value equal to 5 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	0
DIM6_IND	Indicates whether the dimension with a DIM_RK value equal to 6 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	0
DIM7_IND	Indicates whether the dimension with a DIM_RK value equal to 7 in the DIM_TABLE_LIST configuration table is applicable for this KPI. This column is mandatory.	Number	No	0
LIB_INDEP	Name of the library in which independent variable fact data that is associated with the KPI is stored. The library name must be a valid SAS library name.	Char (32)	No	SDL
INDEP_TABLE_NM	Name of the table in which independent variables fact data is stored. The table name must be a valid SAS data set name.	Char (40)	No	Demand_kpi_ ind_vars

Column Name	Description	Data Type	Editable?	Sample Value
INDEP_VAR_LIST	Specifies the list of the retained keys of the independent variable, separated by commas. This list is displayed in the user interface for the KPI. For more information about the INDEP_VAR_RK column of the INDEP_VAR_TABLE configuration table for the retained key value of the independent variable, see "INDEP_VAR_TABLE" on page 36.	Char (400)	No	1,2,5
LANGUAGE_CD	Required language code. If you have purchased the license for performing consensus planning, the value that you enter in this column must match the value that is in the SAS Financial Management Language Code table. Otherwise, you can leave this column blank.	Char (10)	Yes	en
KPI_ID	Unique identifier for the KPI.	Char (32)	No	DEMAND_ID
KPI_ADK	Additional description key column name for the KPI. This column is required in order to integrate SAS Forecast Analyst Workbench with SAS Financial Management.	Char (32)	No	DEMAND
KPI_DESC	Description of the KPI. This column is optional.	Char (255)	Yes	Daily demand.

NPF_CONFIG

The NPF_CONFIG table stores parameters that are used for forecasting the demand for a new product. These parameters are used in the system coding. After you run the ETL jobs, the default values are populated. You must not edit these values except for the value of the NPF_MODE option. In SAS Forecast Analyst Workbench, you can change the default value of NPF_MODE from MULT to ADD, if all the candidate time series contain zero values for a given time point.

Table 3.15 *Description of the NPF_CONFIG Table*

Column Name	Description	Data Type	Editable?	Sample Value
PRODUCT_RK	Unique number of a product for which the NPF_OPTION column is applicable. By default, the options are applicable for all products. Therefore, the value of this column should be 0.	Number	No	0
KPI_RK	Serial number that is associated with the KPI. The default value is 0. The value 0 indicates that the record is applicable for all KPIs.	Number	No	0
NPF_OPTION	Option name that is used to forecast the demand for the new product.	Char (40)	No	For more information about the NPF options, see Table 3.16 on page 41.
NPF_VALUE	The option value.	Char (40)	For more information, see Table 3.16 on page 41.	For more information, see Table 3.16 on page 41.

The options for the NPF_OPTION column are described in the following table.

Table 3.16 *Details of the Options for the NPF_OPTION Column*

Option Name	Description	Default Value	Editable?
NPF_MODEL_ALPHA	Significance value for confidence, and prediction intervals and tests.	0.05	No
NPF_DISTANCE_ METHOD	Distance method that is used in the clustering of the time series for the candidate product.	EUCLID	No
NPF_CLUSTER_ METHOD	Clustering method that is used in the clustering of the time series for the candidate product.	WARD	No
NPF_CLUSTER_ SORTLIST	Sorting method that is used in the clustering of the time series for the candidate product.	CLUSTER	No
NPF_QUERY_ ACCUMULATE	Accumulation method that is used in the **Select Candidate Products** step to create the profile series and panel series.	TOTAL	No

Option Name	Description	Default Value	Editable?
NPF_QUERY_ TRANSFORM	The method of transformation to be applied to the input series before the interpolating function is fit. SAS Forecast Analyst Workbench uses this method of transformation while it is using EXPAND procedure during the **Select Candidate Products** step to create profile series and panel series.	CUSUM	No
NPF_ESM_MODEL	Forecasting model to be used in the exponential smoothing reduction method.	SIMPLE	No
NPF_NEURAL_HIDDEN	Number of hidden units in the neural network model.	1	No
NPF_MODE	Model to be used while adjusting the seasonality. You can change the default value to ADD.	MULT	Yes

NPF_MODEL_LOOPUP

The NPF_MODEL_LOOPUP table contains a list of models that are used to forecast the demand for a new product. SAS Forecast Analyst Workbench supports the Naive and Neural models by default. You can add custom models by specifying the path of the non-compiled model code in the MODEL_PATH column. For naive and neural models, you do not need to include the path in the MODEL_PATH column.

Table 3.17 *Description of the NPF_MODEL_LOOPUP Table*

Column Name	Description	Data Type	Editable?	Sample Value
MODEL_ID	Unique identifier for the model.	Number	No	
MODEL_NM	Name of the model.	Char (40)	No	NAIVE_REG
MODEL_DISPLAY_ NM	Name of the model, which is displayed in the user interface.	Char (40)	Yes	NAIVE MODEL

Column Name	Description	Data Type	Editable?	Sample Value
MODEL_PATH	Path where the SAS macro of the custom model is stored. The model path must contain two backslashes, as shown in the Sample Value column. SAS Forecast Analyst Workbench contains two default models for forecasting the new product: naive and neural. In additional to these two models, you can use a custom model by specifying its path in this column. For more information about a custom model, see "Using a Custom Model" on page 93.	Char (255)	Yes	`c:\\custom_model.sas`

NPF_REDUCTION_METHODS

The NPF_REDUCTION_METHODS table stores the methods that are used in the dimension reduction analysis. The dimension reduction analysis removes the time dependency. The reduced data does not depend on the time or cycle index. The reduction methods are used in the **Select Cluster** step in the New Products workspace in the user interface.Currently, the seasonal decomposition method and exponential smoothing methods are supported.

Table 3.18 Description of the NPF_REDUCTION_METHODS Table

Column Name	Description	Data Type	Editable?	Sample Value
REDUCTION_METHOD_RK	Unique number of the reduction method. Enter `1` to indicate the seasonal decomposition method and enter `2` to indicate the exponential smoothing method.	Number	No	1
REDUCTION_METHOD_NM	Reduction method name. You can edit this value in order to set the value according to your locale.	Char (32)	No	Decomposition

Chapter 4

Integrating SAS Forecast Analyst Workbench with SAS Financial Management

Preparing Data to Integrate with SAS Financial Management

SAS Forecast Analyst Workbench is thoroughly integrated with SAS Financial Management for the consensus planning process. When you are using the consensus planning process, the data flows from SAS Forecast Analyst Workbench to the SAS Financial Management Staging area, and then to the SAS Financial Management Data Mart (which is a MySQL database).

To use SAS Financial Management to carry out the consensus planning process:

1. Run the configuration job that is in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/CONFIGURATION/CONFIG_JOBS/01_CREATE_CONFIG_TABLES`**.

   You must run the configuration job only once immediately after you deploy SAS Forecast Analyst Workbench. When you run the configuration job, the configuration tables (for example, CODE_LANGUAGE_REF, COUNTRY_REF, and CURRENCY) are created.

2. Run the initial one time jobs that are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/CONFIGURATION/CONFIG_JOBS/02_INITIAL_ONE_TIME_JOBS`**.

You must run the initial one-time job only once immediately after you deploy SAS Forecast Analyst Workbench and edit the configuration tables. When you run the initial one-time jobs, the following steps are performed:

a. The UIART tables (for example, PLAN_CONSENSUS_DETAILS and PLAN_FORM_DETAILS) are created.

b. The COUNTRY, CODE_LANGUAGE, and all SAS Forecast Analyst Workbench related dimension tables are created in the SAS Financial Management Staging area.

   These jobs update the SAS Financial Management Stage fact table (GL_TRANSACTION_SUM) with SAS Forecast Analyst Workbench dimension IDs.

c. The data locale (code language), dimension types, and dimensions into the SAS Financial Management Data Mart are loaded.

3. Run the jobs that load dimension members and the user authorizations for dimension members from SAS Forecast Analyst Workbench to the SAS Financial Management Staging area.

   These jobs are in SAS Data Integration Studio at the following location: **/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/JOBS/01_LOAD_FM_STG_TABLES**

4. Run the jobs that load dimension members, users, groups, and user_x_groups from the SAS Financial Management Staging area to the SAS Financial Management Data Mart.

   These jobs are in SAS Data Integration Studio at the following location: **/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/JOBS/02_LOAD_FM_SDM_TABLES**

Integration Jobs for SAS Financial Management

Overview of Integration Jobs

If the plan file includes the license for SAS Financial Management, you must run the jobs that are related to SAS Financial Management integration. The integration jobs are categorized as configuration jobs, initial jobs, and ongoing jobs.

Configuration Job

SAS Forecast Analyst Workbench data is integrated with SAS Financial Management in order to perform consensus planning. You can run the configuration job that is in SAS Data Integration Studio at the following location: **/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/CONFIGURATION/CONFIG_JOBS/01_CREATE_CONFIG_TABLES**

After you run the configuration job, the following tables are created in the configuration library:

- CODE_LANGUAGE_REF
- COUNTRY_REF

- CURRENCY
- SEEDING_DETAILS

The following tables are not used in this release:

- CURRENCY_EXCH_RATE_TYPE
- CURRENCY_EXCH_RATE_SRC
- CURRENCY_EXCH_RATE_SET
- CURRENCY_EXCH_RATE
- CURRENCY_COMPLEX_EXCH_RATE

Initial Jobs

The initial jobs are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/CONFIGURATION/CONFIG_JOBS/02_INITIAL_ONE_TIME_JOBS`**

The following table provides detailed information about the initial jobs.

Table 4.1 *Initial Jobs*

Job Name	Job Description	Required Input
01_LOAD_FM_STG_CONFIG_TABLES	Populates the following tables in the Staging area of SAS Financial Management. • APP_DIMENSION • APP_PROPERTY • APP_MEMBER_PROPERTY • CODE_LANGUAGE • COUNTRY • DIMENSION_TYPE To populate these tables, data must be populated into the required tables, which are listed in the Required Input column. These tables in the SAS Financial Management Staging area are mandatory, including the default country code and language code values. This job performs the following functions: • creates empty structures for all valid dimension tables with respect to SAS Forecast Analyst Workbench in the SAS Financial Management Staging area (for example, PRODUCT, PRODUCT_ASSOC_TYPE, PRODUCT_ASSOC, PRODUCT_NLS • creates empty structures for the STAKEHOLDER dimension • loads all the required valid dimension types, dimensions, and attributes of all dimensions with respect to SAS Forecast Analyst Workbench into the SAS Financial Management Staging area • updates the SAS Financial Management fact table • incorporates all the dimension keys into the SAS Financial Management fact GL_TRANSACTION_SUM table with respect to SAS Forecast Analyst Workbench.	The following tables must be populated with business-specific data: • CODE_LANGUAGE_REF • COUNTRY_REF • DIM_TABLE_LIST • SAS_DIMENSION_TYPE • DIM_VAR_DISP_LIST

Job Name	Job Description	Required Input
02_LOAD_FM_SDM_DATA_LOCALE	Populates the data locale (code language) from the SAS Financial Management Staging area into the SAS Financial Management Data Mart.	Before you run this job, you must have executed the 01_LOAD_FM_STG_CONFIG_TABLES job and the following tables must be populated with data that is specific to SAS Forecast Analyst Workbench: • COUNTRY • CODE_LANGUAGE
03_CREATE_FM_SDM_DIMENSION_TYPES	Populates all the required dimension types from the SAS Financial Management Staging area into the SAS Financial Management Data Mart.	Before you run this job, you must have executed the 01_LOAD_FM_STG_CONFIG_TABLES job and ensured that the data is populated in the DIMENSION_TYPE table in the SAS Financial Management Staging area.
04_CREATE_FM_SDM_DIMENSIONS	Populates all required dimensions from the SAS Financial Management Staging area into the SAS Financial Management Data Mart.	Before you run this job, you must have executed the 01_LOAD_FM_STG_CONFIG_TABLES job and ensured that the data is populated in the following tables in the SAS Financial Management Staging area: • APP_DIMENSION • APP_PROPERTY • APP_MEMBER_PROPERTY_MAP • DIMENSION_TYPE
05_CREATE_FM_UIART_TABLES	Creates the following table structures in the UIART library: • PLAN_CONSENSUS_DETAILS maintains the consensus status of a plan • PLAN_FORM_DETAILS displays the current status of the Stakeholder inputs for the available consensus forms These tables are required for integrating SAS Forecast Analyst Workbench with SAS Financial Management.	

Ongoing Jobs

After you run the initial jobs, you must run the ongoing jobs. The ongoing jobs are categorized as the jobs that load data into the SAS Financial Management Staging area and from the Staging area into the SAS Financial Management Data Mart.

The following jobs load the SAS Forecast Analyst Workbench dimension data into the SAS Financial Management Staging area:

1. 01_LOAD_FM_STG_DIMENSION_MEMBERS
2. 02_LOAD_FM_STG_USER_X_MEMBERS

These jobs are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/JOBS/01_LOAD_FM_STG_TABLES`**

The following table provides detailed information about the ongoing jobs.

Table 4.2 Ongoing Jobs

Job Name	Job Description	Required Input
01_LOAD_FM_STG_DIMENSION_MEMBERS	Populates the dimension data for the following dimensions into the SAS Financial Management Staging area: • TIME_PERIOD • ACCOUNT • CURRENCY • INTORG (Internal Organization) • ANALYSIS • STAKEHOLDER • All valid SAS Forecast Analyst Workbench dimensions (for example, PRODUCT, STORE_LOCATION) This job also populates the default hierarchy records for the INTORG (internal organization), CURRENCY, and STAKEHOLDER dimensions.	The following tables of the CONFIG library must be populated with business-specific data: • KPI_CONFIG • ANALYSIS_VAR The following tables of the solution data layer must be populated with business-specific data: • TIME_PERIOD • TIME_PERIOD_ASSOC_TYPE • TIME_PERIOD_ASSOC • STAKEHOLDER • STAKEHOLDER_ASSOC • All valid SAS Forecast Analyst Workbench dimensions (for example, PRODUCT, STORE_LOCATION)
02_LOAD_FM_STG_USER_X_MEMBERS	Populates the USER_X_MEMBERS table for dimension data into the SAS Financial Management Staging area. This table provides the user association to a dimension member for workflow purposes (which is only valid if those dimension members are used in the target hierarchy of the formset).	The AUTH_ENTRY table of the UIART library must be populated with business-specific data.

The following jobs load the data from SAS Financial Management Staging area into the Data Mart of SAS Financial Management:

1. 01_LOAD_FM_SDM_USERS
2. 02_LOAD_FM_SDM_GROUPS
3. 03_LOAD_FM_SDM_USERS_X_GROUPS
4. 04_LOAD_FM_SDM_DIMENSION_MEMBERS

These jobs are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/FM_INTEGRATION/JOBS/02_LOAD_FM_SDM_TABLES`**

Table 4.3 *Ongoing Jobs*

Job Name	Job Description	Required Input
01_LOAD_FM_SDM_USERS	Populates valid SAS Forecast Analyst Workbench users into the SAS Financial Management Data Mart.	
02_LOAD_FM_SDM_GROUPS	Populates valid SAS Forecast Analyst Workbench user groups into the SAS Financial Management Data Mart.	
03_LOAD_FM_SDM_USERS_X_GROUPS	Populates valid SAS Forecast Analyst Workbench USER_X_GROUPS into the SAS Financial Management Data Mart.	
04_LOAD_FM_SDM_DIMENSION_MEMBERS	Populates all dimension members along with user authorizations for all valid dimension members from the following tables in the SAS Financial Management Staging area into the SAS Financial Management Data Mart: • TIME_PERIOD • ACCOUNT • CURRENCY • INTORG (Internal Organization) • ANALYSIS • STAKEHOLDER • All valid SAS Forecast Analyst Workbench dimensions (for example, PRODUCT, STORE_LOCATION) This job also populates the default hierarchy records for the INTORG (internal organization), CURRENCY, and STAKEHOLDER dimensions from the SAS Financial Management Staging area into the SAS Financial Management Data Mart.	You must execute the following jobs before you execute this job: • 01_LOAD_FM_STG_DIMENSION_MEMBERS • 02_LOAD_FM_STG_USER_X_MEMBERS The following tables in the SAS Financial Management Staging area must be populated with business-specific data: • TIME_PERIOD • GL_ACCOUNT • CURRENCY • INTORG (Internal Organization) • ANALYSIS • STAKEHOLDER • All valid SAS Forecast Analyst Workbench dimensions (for example, PRODUCT, STORE_LOCATION)

SAS Financial Management Configuration Tables

CODE_LANGUAGE_REF

The CODE_LANGUAGE_REF table is required when you are using SAS Financial Management to perform consensus planning. This table contains a row for each language or locale that is used in the solution. This table is required.

Table 4.4 *Description of the CODE_LANGUAGE_REF Table*

Column Name	Description	Data Type	Editable?	Sample Value
LANGUAGE_CD	A code that identifies a language or locale (for example, en).	Char (3)	Yes	en
LANGUAGE_DESC	Description of the character code that identifies the languages.	Char (255)	Yes	ENGLISH
DEFAULT_ LANGUAGE_FLG	Flag that indicates the default language code that is used in SAS Financial Management. The language or locale is used for the names and descriptions in the tables that do not use LANGUAGE_CD as part of the key.	Char (1)	Yes	Y
LOCALE_ LANGUAGE_CD	The 2-character language code that is associated with the locale.	Char (2)	Yes	en
LOCALE_ VARIANT_CD	This column is left blank by default. This column can be used as the third parameter in the aggregate locale string that is used by the Data Mart.	Char (32)	Yes	
LOCALE_ COUNTRY_CD	The 2-character standard code for a country (for example, AF for Afghanistan and AL for Albania).	Char (2)	Yes	US
PROCESSED_ DTTM	The time at which a particular row of data was created or most recently modified by the job.	Number	Yes	04MAY2011:15:47:00

COUNTRY_REF

The COUNTRY_REF table contains 2-character country codes that conform to the ISO 3166 standard. This table is required.

Table 4.5 *Description of the COUNTRY_REF Table*

Column Name	Description	Data Type	Editable?	Sample Value
COUNTRY_CD	A 2-character code for a country (for example, AF for Afghanistan, AL for Albania).	Char (3)	Yes	US
COUNTRY_DESC	Country name or description for the unique 2-character code.	Char (255)	Yes	United States
COUNTRY_ REGION_DESC	Regional location of the country (for example, North America, Eastern Europe, and Asia).	Char (255)	Yes	US
LANGUAGE_CD	A code that identifies the language or locale that is used for names and descriptions. The code must be defined in the CODE_LANGUAGE table (for example, en for English).	Char (3)	Yes	en
PROCESSED_ DTTM	The time at which a row of data was created or most recently modified by the job.	Number	Yes	04MAY2011:09: 55:00

CURRENCY

The CURRENCY table contains the list of standard 3-character ISO 4217 codes. These codes are used to identify currency codes that are required by SAS Financial Management. This table is required.

Table 4.6 *Description of the CURRENCY Table*

Column Name	Description	Data Type	Editable?	Sample Value
CURRENCY_CD	Standard 3-character code used to identify currency (for example, USD for US dollar).	Char (3)	No	USD
CURRENCY_DESC	Currency name or description for the unique 3-character code.	Char (255)	No	US Dollar
LANGUAGE_CD	A code that identifies the language or locale that is used for names and descriptions. The code must be defined in the CODE_LANGUAGE table (for example, en for English).	Char (3)	No	en
CONVERTED_TO_ EURO_FLG	Indicates whether the currency was converted to the Euro.	Char (1)	No	N
PROCESSED_ DTTM	The time at which a row of data was created or most recently modified by the job.	Number	Yes	04MAY2011:09: 55:00

SEEDING_DETAILS

The SEEDING_DETAILS table enables seeding functionality for SAS Forecast Analyst Workbench. The solution provides sample templates for the seeding formulas. Consultants can add more templates to meet business requirements. The sample templates are at the following location: **`<SASCONFIG>\Lev<N>\AppData\SASForecastAnalystWorkbench\templates`**

Table 4.7 Description of the SEEDING_DETAILS Table

Column Name	Description	Data Type	Editable?	Sample Value
FORMULA_RK	Unique identifier for the seeding formula.	Number	No	1
FORMUALA_NAME	Name of the seeding formula (for example, Seeding Target Growth or Seeding Predicted).	Char (40)	No	Seeding Target Growth
FORMULA_DESC	Description of the seeding formula.	Char (255)	No	Seeding Target Growth
FORMUALA_ANALYSIS_VAR_IDS	List of analysis variables, separated by commas. The name of each analysis variables must match the name in the analysis columns (for example, OVERRIDE, CONSENSUS).	Char (255)	No	OVERRIDE,CONSENSUS
FORMULA_MACRO_NM	Name of the seeding template or macro that is used. SAS Forecast Analyst Workbench provides sample templates for the seeding formulas.	Char (255)	No	ddcf_fm_seeding_sample2.sas
FORMULA_ACTIVE_IND	Indicates whether the seeding formula is active. Only one seeding formula remains active for plans at a time. The values must be either 0 or 1.	Number	No	0
PROCESSED_DTTM	The time at which a particular row of data was created or most recently modified by the job	Number	No	12Oct11:03:24:57

Chapter 5

Integrating SAS Forecast Analyst Workbench with SAS Inventory Optimization

Introduction to SAS Forecast Analyst Workbench and SAS Inventory Optimization Integration

In the sales and operations planning process for your organization, you might want to use the forecasted demand as an input to SAS Inventory Optimization to perform further analysis. To carry out such a scenario, you can use the forecasted output of a plan that is created in SAS Forecast Analyst Workbench as input for SAS Inventory Optimization. This chapter also explains the SAS Inventory Optimization integration jobs, the guidelines for loading data, and the guidelines for creating a plan.

Guidelines for Loading Data

Before you load the data, ensure that SAS Forecast Analyst Workbench and SAS Inventory Optimization are installed on the same machine and that both are using MySQL as the transactional database. The MySQL users for both solutions must be unique.

Follow these guidelines when you load the data into SAS Forecast Analyst Workbench and SAS Inventory Optimization 5.1 for their integration:

- SAS Forecast Analyst Workbench and SAS Inventory Optimization 5.1 must use their own SDLs. The data in both these SDLs is loaded separately from the same customer source data.

  When you load the data from the same customer source, the hierarchies and relationships that are defined in the data at the customer source will be reflected in the solution data layers (SDLs) of these solutions.

- After you load the data into the SDLs for SAS Inventory Optimization 5.1 and SAS Forecast Analyst Workbench, the dimension retain key (rk) of the same customer data can be different. But the relationship of the data will be the same.
- The PRODUCT and STORE_LOCATION dimension of SAS Forecast Analyst Workbench are called the Item and Facility dimension respectively in SAS Inventory Optimization 5.1.
- The PRODUCT and STORE_LOCATION combinations that are valid in SAS Forecast Analyst Workbench must be marked as valid in SAS Inventory Optimization 5.1. In order to mark these combinations as valid in the SDL for SAS Inventory Optimization 5.1, you must specify the FACILITY_ITEM_VALID_FLG flag in the FACILITY_X_ITEM table as **Y**.
- In SAS Inventory Optimization 5.1 and SAS Forecast Analyst Workbench, the same date can have different TIME_PERIOD_RK values in each of the TIME_DIM jobs. In other words, the transactional tables can have different TIME_PERIOD_RK values for the same date in both the solutions.
- If the Item or Facility dimension of SAS Inventory Optimization contains an unbalanced hierarchy, you must populate the last available hierarchy level at the missing hierarchy level when you load the data into the SDL of SAS Inventory Optimization. For example, the Facility dimension in SAS Forecast Analyst Workbench contains the following hierarchy: Region > State > City, but the State node is not present in the Facility dimension of SAS Inventory Optimization. When you are loading the data into SAS Inventory Optimization, you must populate the following hierarchy Region > Region > City in the SDL in order to balance the hierarchies.

After you load the data, you must ensure that the value of the GL_FORECAST_DATE parameter in SAS Forecast Analyst Workbench is same as that of the GL_IO_BATCH_CURR_DATE parameter in SAS Inventory Optimization.

Guidelines for Creating a Plan

When you are integrating SAS Forecast Analyst Workbench output with SAS Inventory Optimization, you must make the following selections for the plan. For the detailed information about how to create a plan, see the *SAS Forecast Analyst Workbench: User's Guide*:

- select Demand as the value of KPI for the plan
- select the Product and Store Location dimensions
- select the leaf level of the product and the store location dimension as hierarchy levels
- select all hierarchical data for the Product dimension and for the Store location dimension.
- set the periodicity of the plan to be the period at which you want to run the SAS Inventory Optimization process

  In SAS Inventory Optimization, you can view and set the periodicity in the GL_Base_Period parameter of the Control.global_parameter_list table.
- set the number of periods in the planning horizon to be equal to the planning horizon for SAS Inventory Optimization

- if you are using succession in SAS Forecast Analyst Workbench, use succession in the inventory data population in SAS Inventory Optimization

After you create the plan, diagnose it. You can also edit the parameters and re-diagnose the plan until you are satisfied with the forecasted values.

Integration Jobs for SAS Inventory Optimization

In order to load the forecasted output of the plan into SAS Inventory Optimization, you must run the integration jobs for SAS Inventory Optimization. Before you can run these integration jobs, you must run the following jobs:

- the initial jobs of SAS Inventory Optimization
- the FACILITY_DIM and ITEM_DIM jobs of SAS Inventory Optimization (these jobs are in SAS Data Integration Studio at **`/Products/SAS Inventory Optimization/SAS Inventory Optimization 5.1/IO_JOBS/01_DIMENSION_JOBS`**)

The integration jobs are in SAS Data Integration Studio at the following location: **`/Products/SAS Forecast Analyst Workbench/Forecast Analyst 5.1/IO_INTEGRATION`**.

The following table describes the integration jobs.

Table 5.1 *Integration Jobs for SAS Forecast Analyst Workbench and SAS Inventory Optimization 5.1*

Job Name	Description	Required Input
01_CREATE_IO_FCST_TABLES	Populates the forecasting output of SAS Forecast Analyst Workbench to the location specified in the LIB_PATH job parameter. *Note:* You must execute this job on the SAS Forecast Analyst Workbench application server.	You must specify values for the following parameters: PLAN_NM name of the plan that you want to integrate with SAS Inventory Optimization 5.1 LIB_PATH directory location where you want to store the forecast output data sets of SAS Forecast Analyst Workbench. *Note:* You must grant full permissions for this directory to the SAS Server Users group. For more information about job parameters, see “Job Parameters” on page 83.

Job Name	Description	Required Input
02_LOAD_IO_FCST_TABLES	Uses the forecasted output of a SAS Forecast Analyst Workbench plan to populate the SDL.FORECASTED_DEMAND and SDL.INTERMITTENT_DEMAND tables of SAS Inventory Optimization 5.1. *Note:* You must execute this job on the SAS Inventory Optimization 5.1 application server.	You must specify values for the following parameter: LIB_PATH The directory location where the plan forecast output is stored. This output is created by the 01_CREATE_IO_FCST_TABLES job. For more information about job parameters, see "Job Parameters" on page 83.

To optimize the inventory with the latest forecasted data for the plan, you must run the ongoing jobs of SAS Inventory Optimization 5.1 after you execute the SAS Inventory Optimization integration jobs. These ongoing jobs are in the SAS Data Integration Studio at the following location:

`Products/SAS Inventory Optimization/SAS Inventory Optimization 5.1/IO_JOBS/02_ANALYTICS_JOBS/IO`

In order to use the forecasted demand data of the plan for the next inventory optimization batch run:

1. Schedule the plan to run in batch mode. Use a forecast interval that is same as the base period of the inventory optimization.
2. Run the integration jobs for SAS Inventory Optimization.
3. Run the ongoing jobs for SAS Inventory Optimization.

Appendix 1
Data Dictionary

BOM

The BOM table stores the bill of material (BOM) information, which consists of the details about child products that are related to particular parent products. Information is stored only for child products that can be sold separately. Here is an example of such information:

- Parent product: motor bike
- Child products: engine, pistons, tires

The BOM information is stored in the form of a relationship between the parent and child products.

Table A1.1 *Description of the BOM Table*

Name	Description	Type	Length	Primary Key
BOM_LEVEL_NO	The BOM level number, which indicates the BOM level that exists in the BOM parent-child hierarchy.	Numeric	8	No
PRODUCT_QTY	The quantity of parent products that are associated with a particular bill of material. By default, the quantity that is stored is 1.	Numeric	8	No
CHILD_PRODUCT_QTY	The quantity of child products that are associated with a bill of material.	Numeric	8	No
QTY_UOM_CD	Unit of measurement used to express the quantity.	Character	10	No
VALID_TO_DTTM	The datetime stamp details for the period up to which this record becomes effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key becomes effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	Yes
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No
PARENT_PRODUCT_RK	Retained surrogate key for the parent product.	Numeric	8	Yes
CHILD_PRODUCT_RK	Retained surrogate key for the child product.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
VALID_FROM_ DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	No

CUSTOM1

The CUSTOM1 table contains information about a custom dimension, including its hierarchy and attribute details.

Table A1.2 *Description of the CUSTOM1 Table*

Name	Description	Type	Length	Primary Key
CUSTOM1_RK	Retained surrogate key for the leaf-level node of this dimension.	Numeric	8	Yes
CUSTOM1_ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the dimension.	Character	32	No
CUSTOM1_ID	Unique business key identifier of the leaf-level node of this dimension.	Character	32	No
CUSTOM1_NM	Name of the leaf-level node of this dimension.	Character	40	No
CUSTOM1_DESC	Description of the leaf-level node of this dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No
CUSTOM1_LVL_ RKn	Retained surrogate key for the nth-level node of this dimension.	Numeric	8	No
CUSTOM1_LVL_ IDn	Unique business key identifier for the nth-level node of this dimension.	Character	32	No

Name	Description	Type	Length	Primary Key
CUSTOM1_LVL_NMn	Name of the nth-level node of this dimension.	Character	40	No
CUSTOM1_LVL_DESCn	Description of the nth-level node of this dimension.	Character	255	No
CUSTOM1_ATTRIBn	Value of the nth common attribute of this dimension.	Character	40	No
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

CUSTOM2

The CUSTOM2 table contains information about a custom dimension, including its hierarchy and attribute details.

Table A1.3 *Description of the CUSTOM2 Table*

Name	Description	Type	Length	Primary Key
CUSTOM2_RK	Retained surrogate key for the leaf-level node of this dimension.	Numeric	8	Yes
CUSTOM2_ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the dimension.	Character	32	No
CUSTOM2_ID	Unique business key identifier of the leaf-level node of this dimension.	Character	32	No
CUSTOM2_NM	Name of the leaf-level node of this dimension.	Character	40	No
CUSTOM2_DESC	Description of the leaf-level node of this dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No
CUSTOM2_LVL_ RKn	Retained surrogate key for the nth-level node of this dimension.	Numeric	8	No
CUSTOM2_LVL_ IDn	Unique business key identifier for the nth-level node of this dimension.	Character	32	No
CUSTOM2_LVL_ NMn	Name of the nth-level node of this dimension.	Character	40	No
CUSTOM2_LVL_ DESCn	Description of the nth-level node of this dimension.	Character	255	No
CUSTOM2_ ATTRIBn	Value of the nth common attribute of this dimension.	Character	40	No
VALID_FROM_ DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_ DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

CUSTOMER

The CUSTOMER table contains information about customers, including their hierarchy and attribute details.

Table A1.4 *Description of the CUSTOMER Table*

Name	Description	Type	Length	Primary Key
CUSTOMER_RK	Retained surrogate key for the leaf-level node of this dimension.	Numeric	8	Yes
CUSTOMER_ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the customer.	Character	32	No
CUSTOMER_ID	Unique business key identifier of the leaf-level node of the CUSTOMER dimension.	Character	32	No
CUSTOMER_NM	Name of the leaf-level node of the CUSTOMER dimension.	Character	40	No
CUSTOMER_DESC	Description of the leaf-level node of the CUSTOMER dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No

Name	Description	Type	Length	Primary Key
CUSTOMER_LVL_RKn	Retained surrogate key for the nth-level node of the CUSTOMER dimension.	Numeric	8	No
CUSTOMER_LVL_IDn	Unique business key identifier for the nth-level node of the CUSTOMER dimension.	Character	32	No
CUSTOMER_LVL_NMn	Name of the nth-level node of the CUSTOMER dimension.	Character	40	No
CUSTOMER_LVL_DESCn	Description of the nth-level node of the CUSTOMER dimension.	Character	255	No
CUSTOMER_ATTRIBn	Value of the nth common attribute of the CUSTOMER dimension.	Character	40	No
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

EMPLOYEE

The EMPLOYEE table contains information about employees, including their hierarchy and attribute details.

Table A1.5 Description of the EMPLOYEE Table

Name	Description	Type	Length	Primary Key
EMPLOYEE_RK	Retained surrogate key for the leaf-level node of the EMPLOYEE dimension.	Numeric	8	Yes
EMPLOYEE_ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the employee.	Character	32	No
EMPLOYEE_ID	Unique business key identifier of the leaf-level node of the EMPLOYEE dimension.	Character	32	No
EMPLOYEE_NM	Name of the leaf-level node of EMPLOYEE dimension.	Character	40	No
EMPLOYEE_DESC	Description of the leaf-level node of the EMPLOYEE dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No
EMPLOYEE_LVL_ RKn	Retained surrogate key for the nth-level node of the EMPLOYEE dimension.	Numeric	8	No
EMPLOYEE_LVL_ IDn	Unique business key identifier for the nth-level node of the EMPLOYEE dimension.	Character	32	No
EMPLOYEE_LVL_ NMn	Name of the nth-level node of the EMPLOYEE dimension.	Character	40	No
EMPLOYEE_LVL_ DESCn	Description of the nth-level node of the EMPLOYEE dimension.	Character	255	No
EMPLOYEE_ ATTRIBn	Value of the nth common attribute of the EMPLOYEE dimension.	Character	40	No

Name	Description	Type	Length	Primary Key
VALID_FROM_ DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Character	8	Yes
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Character	8	No
PROCESSED_ DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Character	8	No

ORGANIZATION

The ORGANIZATION table stores a list of organizations, including their hierarchy and attribute details.

Table A1.6 *Description of the ORGANIZATION Table*

Name	Description	Type	Length	Primary Key
ORGANIZATION_ RK	Retained surrogate key for the leaf-level node of the ORGANIZATION dimension.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
ORGANIZATION_ ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the organization.	Character	32	No
ORGANIZATION_ ID	Unique business key identifier leaf-level node of the ORGANIZATION dimension.	Character	32	No
ORGANIZATION_ NM	Name of the leaf-level node of the ORGANIZATION dimension.	Character	40	No
ORGANIZATION_ DESC	Description of the leaf-level node of the ORGANIZATION dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No
ORGANIZATION_ LVL_RKn	Retained surrogate key for the nth-level node of the ORGANIZATION dimension.	Numeric	8	No
ORGANIZATION_ LVL_IDn	Unique business key identifier for the nth-level node of the ORGANIZATION dimension.	Character	32	No
ORGANIZATION_ LVL_NMn	Name of the nth-level node of the ORGANIZATION dimension.	Character	40	No
ORGANIZATION_ LVL_DESCn	Description of the nth-level node of the ORGANIZATION dimension.	Character	255	No
ORGANIZATION_ ATTRIBn	Value of the nth common attribute of the ORGANIZATION dimension.	Character	40	No
VALID_FROM_ DTTM	the datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_ DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

PRODUCT

The PRODUCT table contains information about products, including their hierarchy and attribute details.

Table A1.7 *Description of the PRODUCT Table*

Name	Description	Type	Length	Primary Key
PRODUCT_RK	Retained surrogate key for the leaf-level node of the PRODUCT dimension.	Numeric	8	Yes
PRODUCT_ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the product.	Character	32	No
PRODUCT_ID	Unique business key identifier of the leaf-level node of the PRODUCT dimension.	Character	32	No
PRODUCT_NM	Name of the leaf-level node of the PRODUCT dimension.	Character	40	No
PRODUCT_DESC	Description of the leaf-level node of the PRODUCT dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No

Name	Description	Type	Length	Primary Key
PRODUCT_LVL_RKn	Retained surrogate key for the nth-level node of the PRODUCT dimension.	Numeric	8	No
PRODUCT_LVL_IDn	Unique business key identifier for the nth-level node of the PRODUCT dimension.	Character	32	No
PRODUCT_LVL_NMn	Name of the nth-level node of the PRODUCT dimension.	Character	40	No
PRODUCT_LVL_DESCn	Description of the nth-level node of the PRODUCT dimension.	Character	255	No
PRODUCT_ATTRIBn	Value of the nth common attribute of the PRODUCT dimension.	Character	40	No
PRODUCT_STATUS_CD	Code that indicates the current status of the product (for example, NEW indicates new, ACT indicates active, OBS indicates obsolete).	Character	10	No
PRODUCT_TYPE_CD	Code that indicates the type of the product. You can use this column as an alternate way of grouping the products (for example, BO indicates bought-out products and MC indicates machine components).	Character	10	No
SALES_INTRODUCTION_DT	The date when the product sale was introduced. If the product sale was never introduced, the value is NULL.	Numeric	8	No
SALES_DISCONTINUED_DT	The date when the product sale was discontinued. If the product sale was never discontinued, the value is NULL.	Numeric	8	No
VALID_FROM_DTTM	the datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_ DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

PRODUCT_SPEC

The PRODUCT_SPEC table contains information about the specifications for leaf-level products.

Table A1.8 *Description of the PRODUCT_SPEC Table*

Name	Description	Type	Length	Primary Key
PRODUCT_RK	Retained surrogate key for the leaf-level node of the PRODUCT_SPEC dimension.	Numeric	8	Yes
PARENT_ PRODUCT_LVL_ RK	The retained key of the parent product for which the specification is applicable.	Numeric	8	Yes
SPEC_NM	Name of the product specification.	Character	40	Yes
SPEC_VALUE	Value of the product specification.	Character	40	No
SPEC_DATATYPE	Type of the specification (for example, 0 indicates an integer, 1 indicates a date, 2 indicates a string).	Numeric	8	No
SPEC_UOM_CD	Code for the unit of measure.	Character	10	No

Name	Description	Type	Length	Primary Key
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

PRODUCT_SUBSTITUTE

The PRODUCT_SUBSTITUTE table stores information about the product that is substituted for a particular product. A product can be substituted only if the substituting product has the same properties as the other product.

Table A1.9 *Description of the PRODUCT_SUBSTITUTE Table*

Name	Description	Type	Length	Primary Key
PRODUCT_SUBSTITUTE_RK	Retained surrogate key for the leaf-level node of the PRODUCT_SUBSTITUTE dimension.	Numeric	8	Yes
PRODUCT_RK	Retained surrogate key for the product.	Numeric	8	Yes
PRIORITY_LEVEL_NO	Priority level number.	Numeric	8	No
EFFECTIVE_TO_DTTM	The date and time up to which the substitute product is effective.	Numeric	8	No
EFFECTIVE_FROM_DTTM	The date and time from which the substitute product is effective.	Numeric	8	No
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_ DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

PRODUCT_SUCCESSION

The PRODUCT_SUCCESSION table contains information about the product that is substituted for a particular product. The properties of the substituting product must contain all the properties of the substituted product.

Table A1.10 *Description of the PRODUCT_SUCCESSION Table*

Name	Description	Type	Length	Primary Key
PRODUCT_RK	Retained surrogate key for leaf-level node of the PRODUCT_SUCCESSION dimension.	Numeric	8	Yes
NEXT_PRODUCT_ RK	Retained surrogate key of the next product that is in the hierarchy.	Numeric	8	No
INVENTORY_ EXISTS_FLG	Flag that indicates whether an inventory exists for the product.	Character	0	No
EFFECTIVE_ FROM_DTTM	The date and time from which the succession product is effective.	Numeric	8	No
EFFECTIVE_TO_ DTTM	The date and time up to which the succession product is effective.	Numeric	8	No

Name	Description	Type	Length	Primary Key
VALID_FROM_ DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is in effect in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_ DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

STAKEHOLDER

The STAKEHOLDER table is required only when SAS Forecast Analyst Workbench is integrated with SAS Financial Management. This table contains a row for each stakeholder. Usually, a stakeholder is a user who is responsible for providing input for consensus planning.

Table A1.11 *Description of the STAKEHOLDER Table*

Name	Description	Type	Length	Primary Key
STAKEHOLDER_ RK	Retained surrogate key for the leaf-level node of the STAKEHOLDER dimension.	Numeric	8	No
STAKEHOLDER_ ID	Unique business key identifier of the leaf-level node of the STAKEHOLDER dimension.	Character	32	No

Name	Description	Type	Length	Primary Key
STAKEHOLDER_NM	Name of the member.	Character	40	No
STAKEHOLDER_DESC	Description of the member.	Character	255	No
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	No
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

STAKEHOLDER_ASSOC

The STAKEHOLDER_ASSOC table is required only when SAS Forecast Analyst Workbench is integrated with SAS Financial Management. This table contains the parent-child relationships that make up the STAKEHOLDER hierarchies.

Table A1.12 *Description of the STAKEHOLDER_ASSOC Table*

Name	Description	Type	Length	Primary Key
STAKEHOLDER_ID	Unique business key identifier of the member.	Character	32	No

Name	Description	Type	Length	Primary Key
PARENT_STAKEHOLDER_ID	Unique business key identifier of the parent member.	Character	32	No
ORDER_NO	Order number.	Numeric	8	No
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	No
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

STORE_LOCATION

The STORE_LOCATION table contains information about store locations , including their hierarchy and attribute details.

Table A1.13 *Description of the STORE_LOCATION Table*

Name	Description	Type	Length	Primary Key
STORE_LOCATION_RK	Retained surrogate key for the leaf-level node of the STORE_LOCATION dimension.	Numeric	8	Yes

Name	Description	Type	Length	Primary Key
STORE_ LOCATION_ADK	Additional description key that is used for integrating SAS Forecast Analyst Workbench with SAS Financial Management. The default value is the name of the store location.	Character	32	No
STORE_ LOCATION_ID	Unique business key identifier of the leaf-level node of the STORE_LOCATION dimension.	Character	32	No
STORE_ LOCATION_NM	Name of the leaf-level node of the STORE_LOCATION dimension.	Character	40	No
STORE_ LOCATION_DESC	Description of the leaf-level node of the STORE_LOCATION dimension.	Character	255	No
LANGUAGE_CD	Required language codes.	Character	10	No
STORE_ LOCATION_LVL_ RKn	Retained surrogate key for the nth-level node of the STORE_LOCATION dimension.	Numeric	8	No
STORE_ LOCATION_LVL_ IDn	Unique business key identifier for the nth-level node of the STORE_LOCATION dimension.	Character	32	No
STORE_ LOCATION_LVL_ NMn	Name of the nth-level node of the STORE_LOCATION dimension.	Character	40	No
STORE_ LOCATION_LVL_ DESCn	Description of the nth-level node of the STORE_LOCATION dimension.	Character	255	No
STORE_ LOCATION_ ATTRIBn	Value of the nth common attribute of the STORE_LOCATION dimension.	Character	40	No
STORE_OPEN_ DTTM	The date and time when the store opened.	Numeric	8	No
STORE_CLOSURE_ DTTM	The date and time when the store closed.	Numeric	8	No

Name	Description	Type	Length	Primary Key
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key is in effect. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

TIME_PERIOD

The TIME_PERIOD table contains a list of time periods that are used to represent time in a hierarchy. The table includes information about both child and parent members that are used in the traditional hierarchy or dimension (for example, ALLYEARS, YR2002).

Table A1.14 *Description of the TIME_PERIOD Table*

Name	Description	Type	Length	Primary key
TIME_PERIOD_RK	Retained surrogate key for the leaf-level node of the TIME_PERIOD dimension.	Numeric	8	Yes

Name	Description	Type	Length	Primary key
VALID_FROM_DTTM	The datetime stamp details for the period from which this record is effective in the warehouse. The time value is set to the period at which the record is loaded into the warehouse. The time value must be rounded to full seconds. This column is a versioning instrument. Only one record is valid at any given time. If the value for the time is not available, set it to `00:00:00:00`.	Numeric	8	Yes
VALID_TO_DTTM	The datetime stamp details for the period up to which this record is effective in the warehouse. The time value is set to 1 second before the period at which the next record with the same key becomes effective. The time value must be rounded to full seconds. If the value for the time is not available, set it to `23:59:59:00`.	Numeric	8	No
PERIOD_TYPE_CD	Period type (for example, hour of the day, day, week, fiscal year, fiscal quarter, reporting period)	Character	10	No
TIME_PERIOD_ID	Unique business identifier for the time period.	Character	32	No
TIME_PERIOD_DESC	Description of the time period.	Character	255	No
TIME_PERIOD_NM	Name of the leaf-level node of the TIME_PERIOD dimension.	Character	40	No
SOURCE_SYSTEM_CD	Source system code.	Character	10	No
START_DTTM	The date and time when this time period started.	Numeric	8	No
END_DTTM	The date and time when this time period ended.	Numeric	8	No
PROCESSED_DTTM	The datetime stamp details for the time at which the record was loaded in the warehouse. In the case of updates that do not version the row, such as error correction and data patching, the column records the datetime stamp of the update.	Numeric	8	No

Appendix 2

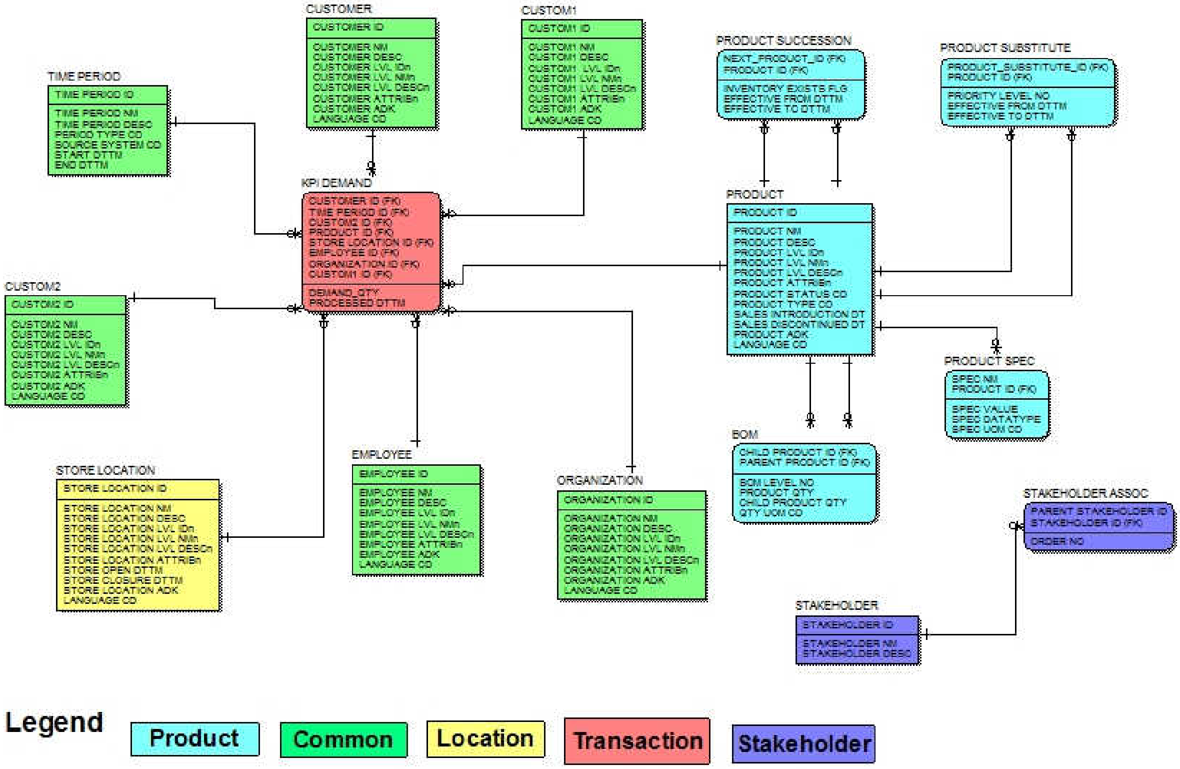
Data Model Diagrams

Logical Data Model

A logical data model (LDM) represents an organization's data, which is organized in terms of entities and relationships. This model is independent of any specific data management technology.

The following figure is the logical data model for SAS Forecast Analyst Workbench:

Figure A2.1 *Logical Data Model*

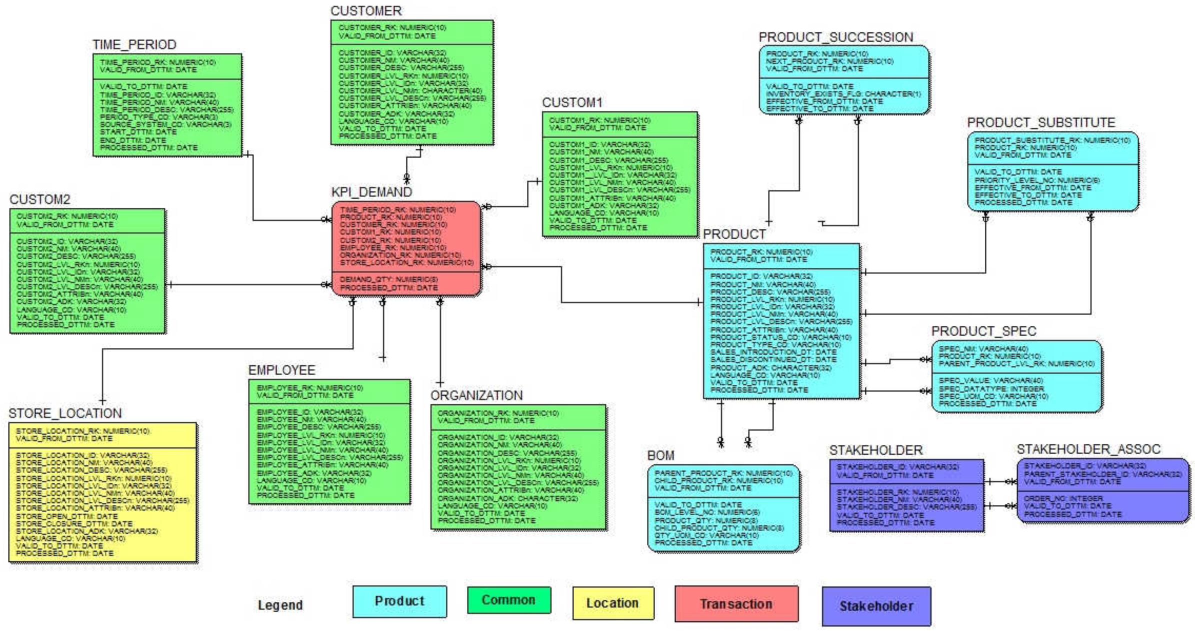


Physical Data Model

A physical data model (PDM) represents a data design, which takes into account the facilities and constraints of a given database management system. This model is typically derived from a logical data model.

The following figure is the physical data model for SAS Forecast Analyst Workbench:

Figure A2.2 *Physical Data Model*


TIME_PERIOD
TIME_PERIOD_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
VALID_TO_DTTM: DATE
TIME_PERIOD_ID: VARCHAR(32)
TIME_PERIOD_NM: VARCHAR(40)
TIME_PERIOD_DESC: VARCHAR(255)
PERIOD_TYPE_CD: VARCHAR(3)
SOURCE_SYSTEM_CD: VARCHAR(3)
START_DTTM: DATE
END_DTTM: DATE
PROCESSED_DTTM: DATE
CUSTOMER
CUSTOMER_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
CUSTOMER_ID: VARCHAR(32)
CUSTOMER_NM: VARCHAR(40)
CUSTOMER_DESC: VARCHAR(255)
CUSTOMER_LVL_RKn: NUMERIC(10)
CUSTOMER_LVL_IDn: VARCHAR(32)
CUSTOMER_LVL_NMn: CHARACTER(40)
CUSTOMER_LVL_DESCn: VARCHAR(255)
CUSTOMER_ATTRIBn: VARCHAR(40)
CUSTOMER_ADK: VARCHAR(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
CUSTOM1
CUSTOM1_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
CUSTOM1_ID: VARCHAR(32)
CUSTOM1_NM: VARCHAR(40)
CUSTOM1_DESC: VARCHAR(255)
CUSTOM1_LVL_RKn: NUMERIC(10)
CUSTOM1__LVL_IDn: VARCHAR(32)
CUSTOM1_LVL_NMn: VARCHAR(40)
CUSTOM1_LVL_DESCn: VARCHAR(255)
CUSTOM1_ATTRIBn: VARCHAR(40)
CUSTOM1_ADK: VARCHAR(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
PRODUCT_SUCCESSION
PRODUCT_RK: NUMERIC(10)
NEXT_PRODUCT_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
VALID_TO_DTTM: DATE
INVENTORY_EXISTS_FLG: CHARACTER(1)
EFFECTIVE_FROM_DTTM: DATE
EFFECTIVE_TO_DTTM: DATE
PRODUCT_SUBSTITUTE
PRODUCT_SUBSTITUTE_RK: NUMERIC(10)
PRODUCT_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
VALID_TO_DTTM: DATE
PRIORITY_LEVEL_NO: NUMERIC(6)
EFFECTIVE_FROM_DTTM: DATE
EFFECTIVE_TO_DTTM: DATE
PROCESSED_DTTM: DATE
CUSTOM2
CUSTOM2_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
CUSTOM2_ID: VARCHAR(32)
CUSTOM2_NM: VARCHAR(40)
CUSTOM2_DESC: VARCHAR(255)
CUSTOM2_LVL_RKn: NUMERIC(10)
CUSTOM2_LVL_IDn: VARCHAR(32)
CUSTOM2_LVL_NMn: VARCHAR(40)
CUSTOM2_LVL_DESCn: VARCHAR(255)
CUSTOM2_ATTRIBn: VARCHAR(40)
CUSTOM2_ADK: VARCHAR(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
KPI_DEMAND
TIME_PERIOD_RK: NUMERIC(10)
PRODUCT_RK: NUMERIC(10)
CUSTOMER_RK: NUMERIC(10)
CUSTOM1_RK: NUMERIC(10)
CUSTOM2_RK: NUMERIC(10)
EMPLOYEE_RK: NUMERIC(10)
ORGANIZATION_RK: NUMERIC(10)
STORE_LOCATION_RK: NUMERIC(10)
DEMAND_QTY: NUMERIC(8)
PROCESSED_DTTM: DATE
PRODUCT
PRODUCT_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
PRODUCT_ID: VARCHAR(32)
PRODUCT_NM: VARCHAR(40)
PRODUCT_DESC: VARCHAR(255)
PRODUCT_LVL_RKn: NUMERIC(10)
PRODUCT_LVL_IDn: VARCHAR(32)
PRODUCT_LVL_NMn: VARCHAR(40)
PRODUCT_LVL_DESCn: VARCHAR(255)
PRODUCT_ATTRIBn: VARCHAR(40)
PRODUCT_STATUS_CD: VARCHAR(10)
PRODUCT_TYPE_CD: VARCHAR(10)
SALES_INTRODUCTION_DT: DATE
SALES_DISCONTINUED_DT: DATE
PRODUCT_ADK: CHARACTER(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
PRODUCT_SPEC
SPEC_NM: VARCHAR(40)
PRODUCT_RK: NUMERIC(10)
PARENT_PRODUCT_LVL_RK: NUMERIC(10)
SPEC_VALUE: VARCHAR(40)
SPEC_DATATYPE: INTEGER
SPEC_UOM_CD: VARCHAR(10)
PROCESSED_DTTM: DATE
EMPLOYEE
EMPLOYEE_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
EMPLOYEE_ID: VARCHAR(32)
EMPLOYEE_NM: VARCHAR(40)
EMPLOYEE_DESC: VARCHAR(255)
EMPLOYEE_LVL_RKn: NUMERIC(10)
EMPLOYEE_LVL_IDn: VARCHAR(32)
EMPLOYEE_LVL_NMn: VARCHAR(40)
EMPLOYEE_LVL_DESCn: VARCHAR(255)
EMPLOYEE_ATTRIBn: VARCHAR(40)
EMPLOYEE_ADK: VARCHAR(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
ORGANIZATION
ORGANIZATION_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
ORGANIZATION_ID: VARCHAR(32)
ORGANIZATION_NM: VARCHAR(40)
ORGANIZATION_DESC: VARCHAR(255)
ORGANIZATION_LVL_RKn: NUMERIC(10)
ORGANIZATION_LVL_IDn: VARCHAR(32)
ORGANIZATION_LVL_NMn: VARCHAR(40)
ORGANIZATION_LVL_DESCn: VARCHAR(255)
ORGANIZATION_ATTRIBn: VARCHAR(40)
ORGANIZATION_ADK: CHARACTER(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
STORE_LOCATION
STORE_LOCATION_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
STORE_LOCATION_ID: VARCHAR(32)
STORE_LOCATION_NM: VARCHAR(40)
STORE_LOCATION_DESC: VARCHAR(255)
STORE_LOCATION_LVL_RKn: NUMERIC(10)
STORE_LOCATION_LVL_IDn: VARCHAR(32)
STORE_LOCATION_LVL_NMn: VARCHAR(40)
STORE_LOCATION_LVL_DESCn: VARCHAR(255)
STORE_LOCATION_ATTRIBn: VARCHAR(40)
STORE_OPEN_DTTM: DATE
STORE_CLOSURE_DTTM: DATE
STORE_LOCATION_ADK: VARCHAR(32)
LANGUAGE_CD: VARCHAR(10)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
BOM
PARENT_PRODUCT_RK: NUMERIC(10)
CHILD_PRODUCT_RK: NUMERIC(10)
VALID_FROM_DTTM: DATE
VALID_TO_DTTM: DATE
BOM_LEVEL_NO: NUMERIC(6)
PRODUCT_QTY: NUMERIC(8)
CHILD_PRODUCT_QTY: NUMERIC(8)
QTY_UOM_CD: VARCHAR(10)
PROCESSED_DTTM: DATE
STAKEHOLDER
STAKEHOLDER_ID: VARCHAR(32)
VALID_FROM_DTTM: DATE
STAKEHOLDER_RK: NUMERIC(10)
STAKEHOLDER_NM: VARCHAR(40)
STAKEHOLDER_DESC: VARCHAR(255)
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
STAKEHOLDER_ASSOC
STAKEHOLDER_ID: VARCHAR(32)
PARENT_STAKEHOLDER_ID: VARCHAR(32)
VALID_FROM_DTTM: DATE
ORDER_NO: INTEGER
VALID_TO_DTTM: DATE
PROCESSED_DTTM: DATE
Legend
Product
Common
Location
Transaction
Stakeholder

Appendix 3
Job Parameters

Overview of Job Parameters

Job parameters help you prepare your data to meet your business requirements. You can modify the parameters of the 04_CREATE_TIME_PERIOD_DATA ETL initial one time job. This chapter describes the procedure for modifying these parameters.

Job Parameter Details

You can modify the parameters for the 04_CREATE_TIME_PERIOD_DATA ETL job. The following list describes the parameters that you can modify for this job.

CAL_YR_START_DT
specifies the date from which to begin generating data. The default value is `'01JAN2000'D`.

CAL_NO_OF_YEARS
specifies the number of years that are included in the calendar time dimension. The default value is `15`.

LOADING_DDTM
specifies the datetime value to be loaded. The default value is `'01JAN2000:00:00:00'DT`.

Note: Ensure that the KPI or fact historical dates are in sync with the time-period data that will be created by the 04_CREATE_TIME_PERIOD_DATA job.

Set or Modify the Job Parameter Values

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

To set or modify the value of a job parameter:

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

Appendix 4
Seeding in SAS Forecast Analyst Workbench

Seeding is the process of using existing data to populate initial values in a worksheet to meet the business requirements. This chapter explains how you can use seeding in the consensus planning process.

The PLAN_FORECAST_FACT_**`<PLAN_ID>`** fact table contains fact columns, dimension key columns, and analysis variables, such as ACTUAL, PREDICTED, CONSENSUS, LAST_CONSENSUS, and OVERRIDE.

The fact table also contains user-defined analysis variables, such as STAKEHOLDER_INP. The ACTUAL is populated with the actual historical data, with respect to the dimensions that you select. The PREDICTED is populated with the forecasted data, with respect to the dimensions that you select.

Default analysis variables (such as CONSENSUS and LAST_CONSENSUS) and user-defined analysis variables (if any) do not contain a default set of values. You can provide default values (by seeding) for these variables, with respect to the dimensions. For example, CONSENSUS = PREDICTED or OVERRIDE = LAST YEAR ACTUALS * 15%

SAS Forecast Analyst Workbench provides sample templates for the seeding at the following location: **`<SASCONFIG>\Lev<N>\AppData\SASForecastAnalystWorkbench\templates`**. You can provide your own templates for seeding by creating an entry for them in the SEEDING_DETAILS table in the configuration library. At any time, only one seeding formula remains active for plans.

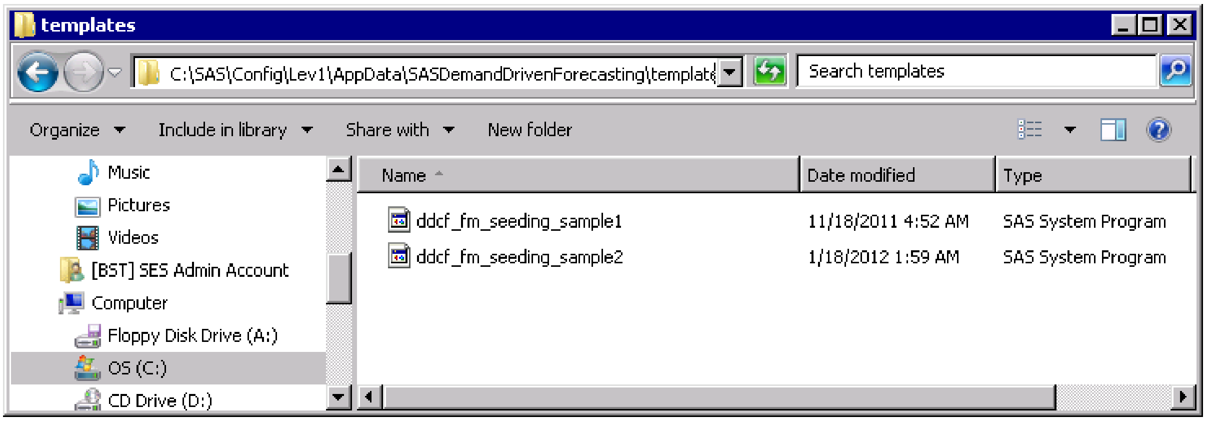
The following table describes the SEEDING_DETAILS table.

Table A4.1 *The SEEDING_DETAILS Table*

FORMULA_ RK	FORMULA_ NAME	FORMULA_ DESC	FORMUALA_ ANALYSIS_ VAR_IDS	FORMULA_ MACRO_NM	FORMULA_ ACTIVE_IND	PROCESSED_ DTTM
1	Seeding Target Growth	Seeding Target Growth	OVERRIDE,CONSENSUS	ddcf_fm_ seeding_ sample2.sas	0	12Oct11:03:24:57
2	Seeding Predicted	Seeding Predicted	OVERRIDE,CONSENSUS	ddcf_fm_ seeding_ sample1.sas	1	12Oct11:03:24:57

The following figure shows the sample templates.

Figure A4.1 *Seeding Templates*



Name	Date modified	Type
ddcf_fm_seeding_sample1	11/18/2011 4:52 AM	SAS System Program
ddcf_fm_seeding_sample2	1/18/2012 1:59 AM	SAS System Program

The following examples describe the macros.

Seeding Scenario Sample 1:

The following formulae are applicable in this template.

1. **First period/First Run**

   Analysis Variable = Predicted/Baseline forecast

2. **Second period/Subsequent Run**

   Analysis Variable = Last Period Consensus

Seeding Scenario Sample 2:

The following formulae are applicable in this template.

1. **First period/First Run**

   Analysis Variable = LY Actuals * (+/- TARGET Growth)

2. **Second period/Subsequent Run**

   Analysis Variable = LPConsensus

Implementation consultants customize these macros to meet the business requirements. The format of the macros is predefined. For more information, see the sample templates provided.

The SCRATCH.PLAN_FORECAST_FACT_**<PLAN_ID>** table is used as input for these macros. After the macros are run, the SEEDING_INPUT_**<PLAN_ID>** output table is created.

The columns of the SEEDING_INPUT_**<PLAN_ID>** table must contain DIMENSION KEY COLUMNS of the PLAN_FORECAST_FACT_**<PLAN_ID>** table and seeded ANALYSIS variables.

The following table is the sample SEEDING_INPUT_**<PLAN_ID>** table that illustrates the example.

Table A4.2 *Sample SEEDING_INPUT_<PLAN_ID> Table*

PRODUCT_ LVL_RK3	STORE_ LOCATION_ LVL_RK2	CAL_MTH_SK	FCST_DATE	OVERRIDE	CONSENSUS
1004	1002	121	01Jan10	126497	126497
1004	1002	122	01Feb10	130673.4	130673.4
1004	1002	123	01Mar10	126407.2	126407.2
1004	1002	124	01Apr10	130903	130903
1004	1002	125	01May10	132834.9	132834.9

This table is merged with the PLAN_FORECAST_FACT_**`<PLAN_ID>`** table by using key columns. After the tables are merged, the analysis variables of the fact table are seeded as per your business requirements, and the PLAN_FORECAST_FACT_**`<PLAN_ID>`** table is available in the PLANEXPL library.

Appendix 5
Customizing the Consensus Planning Process

Overview of Customizing the Consensus Planning Process

In the sales and operations planning process in your organization, you might want to customize the consensus planning process to meet your business requirements. SAS Forecast Analyst Workbench enables you to customize the consensus process at the following two points in the process:

- while data is being loaded into SAS Financial Management
- while the consensus data is being written back from SAS Financial Management to SAS Forecast Analyst Workbench

You can modify the following stored processes to customize the consensus planning process:

- **`ddcf_fm_initiate_consensus_stp.sas`**
- **`ddcf_fm_complete_formset_stp.sas`**

These stored processes are in the following location: **`<SASHome>\SASFoundation\9.3\ddcfsrv\sasstp`**

Customizing the ddcf_fm_initiate_consensus_stp.sas Stored Process

The ddcf_fm_initiate_consensus_stp.sas stored process helps you load the plan-specific dimension data, hierarchy data, and fact data into SAS Financial Management.

You can customize a macro that actually loads the data. You must not edit the rest of the code of the stored process.

In the ddcf_fm_initiate_consensus_stp.sas stored process, you can modify the following part:

```
%if (&job_rc<5)%then %do;
%ddcf_fm_stg_load_plan_data (plan_id = &plan)_id );
/* CUSTOMIZATION CAN BE PROVIDED AT THIS PLACE */
%end;
%else %do;
%let consensus_status=7;
%goto exit;
%end;
```

The %ddcf_fm_stg_load_plan_data macro loads the data to the Staging area of SAS Financial Management. SAS consultants and implementation experts can modify the macros and customize how data is loaded to the Staging area of SAS Financial Management. For example, you might want to modify the macro to load currency exchange rates data.

The data that is loaded into the Staging area or solution data mart of SAS Financial Management is used for consensus planning.

Customizing the ddcf_fm_complete_formset_stp.sas Stored Process

The ddcf_fm_complete_formset_stp.sas stored process extracts the consensus data from SAS Financial Management and updates the SAS Forecast Analyst Workbench fact tables.

You can customize a macro that actually loads the data. Do not edit the rest of the code for the stored process.

In the ddcf_fm_complete_formset_stp.sas stored process, you can modify the following part:

```
%let dimension=FAW_ANALYSIS;
%let dsname=scratch.output_plan_&plan_id
 %if (&job_rc < 5) %then %do;
%ddcf_fm_sdm_extract_facts(plan_id= &plan_id,dimension= &dimension, dsname= &dsname);
%end;
%if (&job_rc < 5) %then %do;
/* CUSTOMIZATION CAN BE PROVIDED AT THIS PLACE */
Either suppress the following macro and write new code or after completion of this macro provide an additional macro
%ddcf_fm_update_consensus(config=config,stage=stagefm,uirt=uiart,plan_id= &plan_id,facttable=plan_forecast_fact,dsname= &dsname);
/* CUSTOMIZATION CAN BE PROVIDED AT THIS PLACE */
```

The %ddcf_fm_update_consensus macro loads the data from SAS Financial Management into SAS Forecast Analyst Workbench. You can modify the macro so that the data that is to be loaded into SAS Forecast Analyst Workbench can be modified.

For example, you might want to load the inputs that were given by various functional departments along with the final consensus data.

The data that you load into SAS Forecast Analyst Workbench is displayed in the Explore Demand view and the Model Management view.

Appendix 6
Using Plan Output for Reporting

Overview

In the sales and operations planning process within your organization, you might want to use the output of a plan for additional processing and reporting. You can use SAS reporting products (such as SAS Web Report Studio) to obtain and deliver the information that you need, when you need it, and in the format that you need.

Generating Reports of the Forecasted Data

In order to use the forecast data for reporting, the system administrator must perform the following steps:

1. Ensure that the location where forecasted data should be generated for reporting is specified in the GL_DDF_ARCHIVE_DIR_PATH configuration parameter. For more information about configuration parameters, see *SAS Forecast Analyst Workbench: System Administration Guide*.
2. Run the following macro through SAS Data Integration Studio: %ddcf_generate_forecast_reports(plan_id=).

   The plan_id is the identification number of the plan. This identification number is in the Plan_Details table, which is in the UIART library.

After the macro is run, SAS Forecast Analyst Workbench creates the following tables.

Table A6.1 *Plan-Specific Tables That Contain Forecasted Data*

Table Format	Description
PLAN_FCST_ DETAILS<PLAN_ ID>	Contains the complete plan hierarchy, including the hierarchy name and forecast related information (for example, the values for ACTUAL, PREDICTED, LCL, UCL)
PLAN_STAT_ DETAILS<PLAN_ ID>	Contains the complete plan hierarchy, including the hierarchy name and statistical information (for example, the values for MAPE, RMSE, PREDICTION_APE)

SAS Forecast Analyst Workbench updates these tables with the latest forecasted data. You can use these tables to generate the reports and to archive consensus data.

Archiving Consensus Data and Generating Reports

In order to archive the output of the plan that contains the consensus data for additional processing, your system administrator must perform the following steps:

1. Set the GL_INCLUDE_ARCHIVE_IND configuration parameter to 1. For more information about configuration parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.
2. Ensure that the location where data should be archived is specified in the GL_DDF_ARCHIVE_DIR_PATH configuration parameter.

In order to use the consensus data for reporting, the data administrator must perform the following steps:

1. Ensure that the GL_INCLUDE_ARCHIVE_IND configuration parameter is set to 1. For more information about configuration parameters, see the *SAS Forecast Analyst Workbench: System Administration Guide*.
2. Ensure that the location where data should be generated for reporting is specified in the GL_DDF_ARCHIVE_DIR_PATH configuration parameter.
3. Run the following macro through SAS Data Integration Studio: %ddcf_generate_consensus_reports(plan_id=, consensus_date=, version=).

   In the macro, the plan_id is the identification number of the plan. This identification number is in the Plan_Details table, which is in the UIART library. The consensus_date is the date on which the consensus process was executed, and version is the consensus version number.

SAS Forecast Analyst Workbench generates the reporting data in plan-specific tables in the location that is specified in the GL_DDF_ARCHIVE_DIR_PATH configuration parameter. SAS Forecast Analyst Workbench creates the table in the following format: CONSENSUS_REPORTING<PLAN_ID>_<VERSION>. This plan-specific table contains the complete plan hierarchy (rk and names), and the corresponding analysis variables values.

Appendix 7
Using a Custom Model

Writing Code for a Custom Model

You can write the code for additional models that you want to appear in the **Select model** list of the **Select Model** step in the New Products workspace. The following code describes the sample code of the SAS file that you will write. The code must contain a call to the macro ddcf_npf_spec, as shown in this example.

```
%macro ddcf_profile_naive_mod(
                  /*- input Panel Series data set information  -----*/
dataset=,         /*- I - input data set                           -*/
var=,             /*- I - variable name                            -*/
byvars=,          /*- I - List of BY variable names                -*/
seriesid=_SERIES_,/*- I - series ID variable name                  -*/
cycleid=_CYCLE_,  /*- I - cycle ID variable name                   -*/
                  /*- output model predictions data set information-*/
outset=,          /*- O - output data set                          -*/

                  /*- options -------------------------------------*/
alpha=       /*- I - confidence limit size                    -*/
);
%ddcf_npf_spec;
  /*-----------------------------------------------------------*/
     /*- sort by Cycle Index                                     -*/
     /*-----------------------------------------------------------*/
         PROC SORT DATA=&dataset(KEEP=&byvars&seriesid&cycleid&var)    OUT=temp_prof_naive;
         BY &cycleid
         RUN;
     /*-----------------------------------------------------------*/
     /*- compute the NAIVE model. Run proc reg                    -*/
     /*-----------------------------------------------------------*/
         proc reg data=temp_prof_naive(rename=&var=actual) NOPRINT;
         id &byvars &seriesid
         by &cycleid
         model ACTUAL = / alpha=α
         output out= &outset P=PREDICT LCL=LOWER UCL=UPPER STDP=STD RESIDUAL=ERROR;
         run;
         quit;
         %if &SYSERR^=0 & &SYSERR ^= 4 %then %do;
            %let npf_model_status = 1;
             %goto EXIT;
         %end;
     /*----------------------------------------------------------*/
     /*- sort by BY variable                                    -*/
     /*----------------------------------------------------------*/
         proc sort data=&OUTSET out=&OUTSET
         by &byvars&seriesid&cycleid
         run;
        /* format the output */
         data &outset
         set &outset
         format ACTUAL  percent6.2;
         format PREDICT percent6.2;
         format LOWER   percent6.2;
         format UPPER   percent6.2;
         RUN;
  %EXIT:;
%mend ddcf_profile_naive_mod;
%ddcf_profile_naive_mod( dataset=npf.Query_panel_subset&PRODUCT_KPI_LIST_ID,
                           var=_PROFILE_,
                           byvars=&NPF_BYVARs,
                           alpha=&npf_model_alpha,
                           outset=npf.Mdl_prof_&model_id._&PRODUCT_KPI_LIST_ID
                           );
```

Input Table

The input table for the macro is NPF.QUERY_PANEL_SUBSET&**`PRODUCT_KPI_LIST_ID`**. This table is always present in the NPF library. The value of **`PRODUCT_KPI_LIST_ID`** is the unique identifier of the new product, which can be found in the UIART.PRODUCT_KPI_LIST table. The following table is an example of the input table.

PRODU RK	_SERIES	start_dt	_CUSUM	Demand QTY	cycle_of_npf	_AGG_	_SHARE	_CYCLE_	SUM	_PROFIL	_CUPCT_
5	4	1-Jan-10	1611	1611	1	123522	1.30%	1	32222	5.00%	5.00%
5	4	2-Jan-10	3225	1614	2	123649	1.30%	2	32222	5.00%	10%
5	4	3-Jan-10	4800	1575	3	123891	1.30%	3	32222	4.90%	15%

Output Table

The output table is created from the input table and the macro. The name of the output table should be in the following format: NPF.MDL_PROF_&**MODEL_ID**_&**PRODUCT_KPI_LIST_ID**. In the name, **MODEL_ID** refers to the unique number for the model mentioned in this table and **PRODUCT_KPI_LIST_ID** is the unique identifier of the new product, which can be found in the UIART.PRODUCT_KPI_LIST table.

The following table is an example of the output of custom model table.

PRODUCT_ RK	_SERIES_	_CYCLE_	actual	PREDICT	STD	LOWER	UPPER	ERROR
5	4	1	5.00%	5.00%	0.000534	4.60%	5.40%	3.39E-05
5	4	2	5.00%	5.00%	0.000456	4.70%	5.40%	-0.00026
5	4	3	4.90%	5.00%	0.000593	4.50%	5.50%	-0.00105

The first three columns also appear in the input table. The following list describes the rest of the columns:

- **PREDICT**: Predicted values
- **LOWER**: Lower confidence limit
- **UPPER**: Upper control limit
- **ERROR**: Error term

Code for the ddcf_npf_spec Macro

You can use the following sample code for the ddcf_npf_spec macro.

```
*---------------------------------------------------------------------
* This macro defines the byvariables, dependent variables, interval,
    cycle index, kpi, cluster specifications etc for npf flow.
*--------------------------------------------------------------------;
%macro ddcf_npf_spec;
%Global NPF_BYVARS
        NPF_NUM_BYVARS
        NPF_BYVAR1
        NPF_BYVAR2
        NPF_TIMEID
        npfinterval
        disp_cycle
        calc_cycle
        npf_kpi_var_nm
        NPF_DEPVARS
        NPF_NUM_DEPVARS
        NPF_DEPVAR1
        NPF_CURRENT_DEPVAR
        NPF_NCLUSTERS
        npf_kpi_aggre
        npf_kpi_lib
        npf_kpi_lib
        ;
*----------------------------------------------------------------------
* Specify the Time Series Data Set BY variables names
*---------------------------------------------------------------------;
/*%let NPF_BYVARS      = Product_RK  Product_NM;*/
%let NPF_BYVARS      = Product_RK;
%let NPF_NUM_BYVARS  = 1;
%let NPF_BYVAR1      = Product_RK;
%let NPF_BYVAR2     = Product_NM;
%let NPF_TIMEID      = cycle_of_npf;
*----------------------------------------------------------------------
* Specify the Time Series Data Set Time ID variable name
*---------------------------------------------------------------------;
        data _null_;
            set uiart.product_kpi_list;
            where PRODUCT_KPI_LIST_ID=input("&PRODUCT_KPI_LIST_ID",best12.);
/*            where PRODUCT_RK = input("&PRODUCT_RK",best12.) and KPI_RK = input("&KPI_RK",best12.);*/
            call symputx('npfinterval',PERIODICITY);
            call symputx('disp_cycle',NO_OF_CYCLES_FOR_DISP);
/*            call symputx('calc_cycle',NO_OF_CYCLES_FOR_CALC);*/
            call symputx('NPF_NCLUSTERS',no_of_clusters);
            call symputx('KPI_RK',KPI_RK);
            call symputx('PRODUCT_RK',PRODUCT_RK);
            stop;
        run;
        %let calc_cycle=&DISP_CYCLE
         data _null_;
          set config.kpi_config;
           where kpi_rk = input("*KPI_RK",best12.);
           call symputx('npf_kpi_var_nm',kpi_Var_nm);
           call symputx ('npf_kpi_aggre',KPI_AGGRE_METHOD);
           call symputx('npf_kpi_lib',LIB_NM);
           call symputx('npf_kpi_table',KPI_TABLE_NM);
        run;
*----------------------------------------------------------------------
* Specify the Time Series Data Set Dependent Variable names
*---------------------------------------------------------------------;
%let NPF_DEPVARS     = &npf_kpi_var_nm ;
%let NPF_NUM_DEPVARS = 1;
%let NPF_DEPVAR1     = &npf_kpi_var_nm ;
%let NPF_CURRENT_DEPVAR = &npf_kpi_var_nm ;
**********************************************************************
* SPECIFIY QUERY SPECIFICATION - CLUSTER SPECIFICATION AND MODEL SPECIFICATION
*********************************************************************;
%ddcf_load_global_parameters;
%ddcf_npf_interval_specs;
%ddcf_set_npf_option_val;
%ddcf_npf_prod_kpi_options;
%ddcf_set_default_npf_param;
%mend ddcf_npf_spec;
/*%ddcf_npf_spec;*/
```

Index