SAS® Demand Forecasting for Retail 4.2-SP3

User's Guide

The Power to Know.

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2009. *SAS® Demand Forecasting for Retail 4.2: User's Guide***.** Cary, NC: SAS Institute Inc.

SAS® Detail Forecasting for Retail 4.2-SP3: User's Guide

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

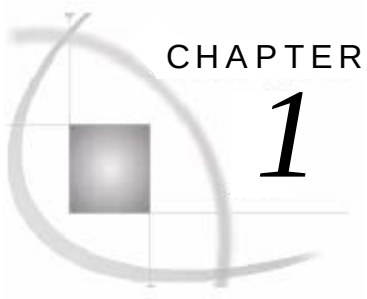
May 2009

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

Contents

Figures

Tables



CHAPTER

1 Document Introduction

1.1 Overview of the User's Guide

SAS Demand Forecasting for Retail enables retailers to build, deploy, and monitor forecasting models to optimize their merchandise planning process.

This document provides the following information to the user:

- Business benefits of demand forecasting in planning process.
- Solution architecture in connection with merchandise planning application and complete workflow summary.
- A step-by-step approach in creating and managing forecasting requests. The details for the administrators and forecast analysts on various elements of system parameters and their usage to leverage solution benefits.
- Detailed program explanation for coders about specifications and parameters pertaining to specific programs to help them customize the solution to suit customer-specific needs.

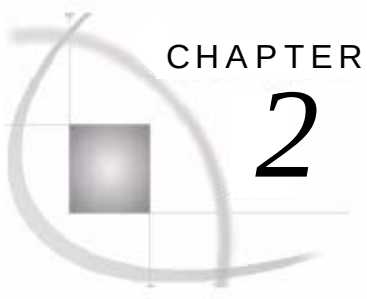
1.2 Audience

This document should be read by business domain experts, business consultants, data architects, analytical consultants, data management specialists, application developers, instructors, testers, and solution specialists in the service provider's project team, as well as IT analysts, system administrators, application developers, and database administrators in the customer's project team.

It can also be read by anyone who needs to understand the analytical data mart or requires guidance to populate the data mart for customer-specific situations.

1.3 Purpose of the Document

This document describes business, technical, and analytical usage of the solution and can be used by anyone who is engaged in implementation of the solution directly or indirectly. This document can be referred for implementation and post-implementation activities of the solution. It can also be referred for the specifics of a component, its usage, parameters, input-output interface, its impact on other programs, and connections and importance of each component in the complete system.



CHAPTER 2

Introduction to the SAS Demand Forecasting for Retail

2.1 Overview of the Solution

Forecasting accurate demand is critical for advanced retail management. Retail organizations offer a multitude of merchandise across extensive selling locations. Forecasting the demand for all Stock Keeping Units (SKUs), geographies, and channels is a huge challenge. The forecasting need too varies by granularity, point in time, and frequency depending on the process used for forecasting. For example, financial planning requires a department or class-level preseason forecast whereas the replenishment process requires a SKU-level daily forecast. Additionally, inaccurate and incomplete data further adds to constraints in forecast accuracy.

SAS Demand Forecasting for Retail generates the right level of forecast to meet specific operational needs. The solution offers varied statistical models to support business-specific, preseason, and in-season forecasts including the capability to account for the lost sales. It creates, evaluates, and monitors the statistical models to ensure consistent forecast accuracy over a period of time. The user can create automated forecasts as well as exception-based forecasts. The analyst can select demand models, tune the forecasting workflow, and apply operational parameters to create the forecasts that are more responsive to the business model.

SAS Demand Forecasting for Retail provides an in-depth understanding of demand. This understanding is the key to effectively align the financial planning, assortment planning, allocation, and replenishment processes in an organization. The solution provides interfaces to generate forecasts for preseason and in-season financial or assortment plans from SAS Merchandise Planning suite, and through a batch process server. Merchandise and financial plans can refer or be initialized from one of the multiple forecast versions that can be created. The product also provides support for in-season re-forecasting, clustering, and new-product forecasting.

The forecast execution process consists of data preparation, diagnosis, model selection, fit/forecast, automated forecast quality evaluation, and reconciliation. The user can define relevant performance metrics to evaluate the generated forecast. Using SAS analytics, forecasts adjust rapidly to the changing demand and quantity patterns to effectively handle seasonality and difficult-to-predict demand for slow-moving goods. The solution enables retailers to estimate the effect of past promotions and the uplift generated by them. Retailers can use this information to forecast the future demand that is generated by promotions.

SAS Demand Forecasting for Retail uses SAS Forecast Server for forecast generation. The solution integrates with SAS Merchandise Planning and provides specific retail features such as forecast job management, version, new product, and clustering support.

SAS Demand Forecasting for Retail follows an integrated approach with an external planning or a replenishment system to automate the forecast execution process. The solution helps retailers to improve their forecasting process with manageable and controllable forecasting flows that can be easily implemented.

2.1.1 Forecasting for Planning

Accurate forecast of the demand is a key input for financial and merchandise planning. SAS Demand Forecasting for Retail provides a mechanism to deliver forecasts to SAS Merchandise Financial Planning (formerly Marketmax Financial Planning) and SAS Merchandise Assortment Planning (formerly Marketmax Assortment Planning). The time dimension for forecast can be quarter, month, or week.

The forecast execution process can be executed for each distinct level of merchandise and location as well as for each distinct KPI. The process includes data preparation, diagnosis, model selection, parameter fitting forecast generation, automated evaluation, and reconciliation. An analyst can choose to skip one or more of the steps in the above process while working on a forecast or during the batch execution. For example, the analyst can choose to turn off the automated evaluation process option or skip re-extraction of data during a test-run, if the data has already been extracted.

2.1.1.1 Integration with SAS Merchandise Planning

The solution can extract sales data from SAS Merchandise Planning, prepare the data for forecasting, generate forecasts, and write back the results into SAS Merchandise Planning. The solution supports forecasting demand for SAS Merchandise Planning (Editions 6.1.2 and 6.1.3).

The sales data from SAS Merchandise Planning solution is extracted in time series format. Data for promotions, events, price, and weather is attached to the sales data as applicable. The forecast is then generated on the resultant data set with selected parameters as independent variables. Core components of this solution include forecasting macros that are based on High Performance Forecasting procedures of the SAS Forecast Server. Apart from these macros, the solution also contains a data extraction component from SAS Merchandise Planning Analytic framework.

SAS Demand Forecasting for Retail offers forecasting features that are vital for an effective planning process. The solution can perform the following tasks:

- manage multiple forecasts across the multi-dimensional hierarchy
- create, manage, and update different forecast versions
- reconcile forecasts using different options such as middle-out, top-down, and bottom-up across different plans/forecasts based on the need
- provide additional options for 'no reconciliation' and 'lowest level only' forecast generation
- enable manual edits of the forecasts based on external information
- compare different forecasts and creating exception or variance reports
- produce forecasts for new products using like-item data
- enable any number of KPIs to be executed within a forecast job (for example, Sales Units, Sales Dollars, and Inventory Count)
- automate the forecast for any level in merchandise and location hierarchy for weekly time grain
- parameterize and control forecast workflow at individual time series level
- generate forecasts at a level that is easy to import into the Merchandise Planning solution
- support forecast using dynamic time-set definition

- run in SAS Grid Environment
- support external user-defined group key

2.1.1.2 Business Process to Forecast for Merchandise Planning

Analytical worksheets (the forecasting requests) are created using the user interface of SAS Merchandise Planning solution. These forecasting requests are picked up by the SAS Merchandise Planning Process Server and are run to generate the forecasts. The results are written back into the database of the SAS Merchandise Planning solution.

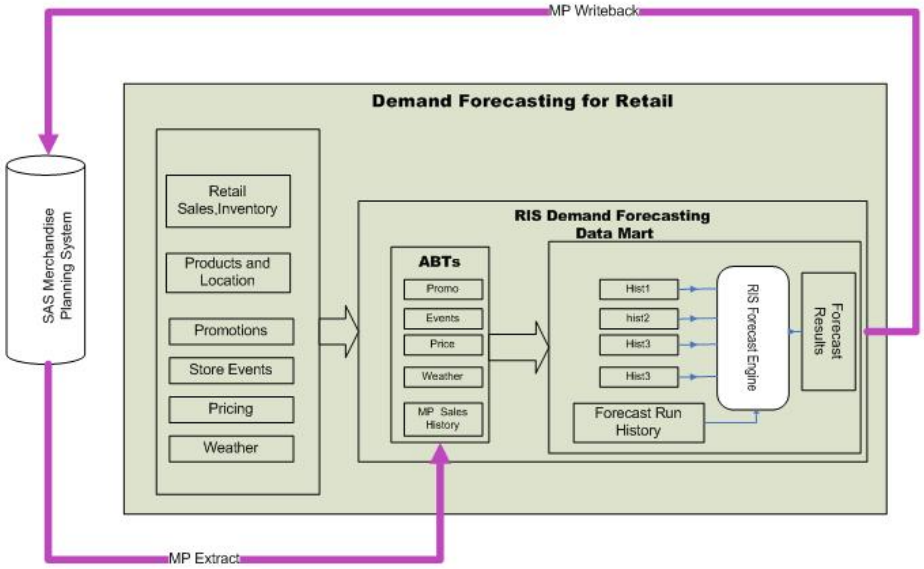


Figure 1. SAS Merchandise Planning Integration for Demand Forecasting

2.1.1.3 Forecasting Product Segments

The success of any retail strategy lies in its planning and execution. Effective product strategy demands comprehensive product planning in terms of assortment, price, promotion, shelf presentation, and adequate product supply.

Product categorization helps the planning process to correctly take the corporate strategy forward to the point of execution. Categorization of products is based on several influencing factors namely, Consumer, Market, Price, Competition, Technology, and so on. Defining distinct plans for each category is vital for efficient category management and eventually provides the consumer with appropriate type and quantity of goods.

SAS Demand Forecasting for Retail supports the concept of ‘Group Key’. Group key helps a retail planner to flag and group individual products as well as product categories of similar nature, together. The various product groups identified can be Premium, High Value, Economy, Traffic Builders, Loss Leaders, and so on. The system allows the planner to assign distinct forecasting and planning strategies for each group. Group key can be assigned to the results of Stock Profiling activity that generates groups (for example, the groups based on similar sales pattern and helps achieve better control over the otherwise massive forecasting process). Each group can have its individual needs for forecasting algorithm. For example, specialty grocery products like ‘Ponche’, ‘Mole sauce’, ‘Tejocotes’, and ‘Hominy’ can be assigned the same group key and planned prominently for store located in area of high Hispanic population. In future if any new brand launches a similar regional-specialty product, then it can follow the existing group key.

The concept of group key helps to implement group-specific product strategies in terms of addressing appropriate market coverage, product retention, and so on. This also enables assortment variation and meaningful additions in the right categories, skillfully aimed at appreciative consumer value and loyalty.

2.2 Functional Modules of SAS Demand Forecasting

The SAS Demand Forecasting for Retail is classified into the following modules:

- Project (Job) Creation
- Database and Integration
- Data Preparation
- New Product Forecast
- Forecasting

The subsequent topics further highlight each of these modules.

2.2.1 Project (Job) Creation

The Project or Job Creation module:

- offers forecast project creation and management through the user interface (UI)
- defines the forecast scope by using retail hierarchies through the UI
- saves UI forecast parameters with a job for easy testing and re-use at a later instance for any purpose
- runs online through plan management or in batch mode through scheduler
- provides exception management through SAS Forecast Studio

2.2.2 Database and Integration

The Database and Integration module has the following features:

- extract and import using SAS Merchandise Planning data store
- version support for multiple stored forecasts
- metadata for flexible configuration of forecast KPIs and forecast storage
- support for Oracle
- auto-archive and data purge

2.2.3 Data Preparation

The Data Preparation module:

- provides new product history assignments
- filters time series based on the availability of history
- provides data integration for secondary parameters such as price, promotion, and events
- provides zero-week fill

2.2.4 New Product Forecast

The new product forecast feature of the solution provides:

- merge history feature to support forecasting for new products and stores
- aggregation for all levels of time, location, and merchandise within the selected worksheet boundaries for options, namely "All History" and "New Product History"

2.2.5 Forecasting

The Forecasting module:

- uses SAS High-Performance Forecasting 2.3 for forecast execution
- supports flexible parameters for workflow, diagnose, and execution control
- facilitates quick testing and batch production by allowing to save parameters with the jobs
- provides custom model file support
- allows parameters to be overridden by group key
- supports user-defined events as well as standard events that can be attributed in forecast generation
- provides a flexible workflow that can be configured as well as controlled through user-defined parameters
- provides top-down, bottom-up, and middle-out methods for reconcile on merchandise or location hierarchy
- provides revisit logic for auto-forecast quality and re-diagnose
- creates, maintains, and generates flexible reports
- supports multi-KPI forecasts
- supports multi-threading forecast execution and SAS Grid Manager for scalable performance
- forecast and reconciliation at week, month, and quarter level

CHAPTER

3 Architecture and Workflow Summary

3.1 Overview of the Forecasting Workflow

The forecasting workflow begins as soon as a forecast request is created. This forecast request initiates all the steps shown in the figure below. These steps are performed automatically without user intervention. The solution uses the SAS Merchandise Planning user interface to set up a forecast project and define scope of the forecast. A forecast project can be defined for all merchandise members and for a single or multiple levels of merchandise within a set of locations for a defined future timeframe.

The solution enables execution of multiple forecast requests on a regular basis in batch mode. Alternatively, the requests can also be executed online. Each forecast is targeted for a subset of merchandise and locations as defined by a step, in a business process such as preseason and in-season assortment or financial planning.

The following figure displays the steps in the forecasting process flow as supported by the SAS Demand Forecast for Retail.

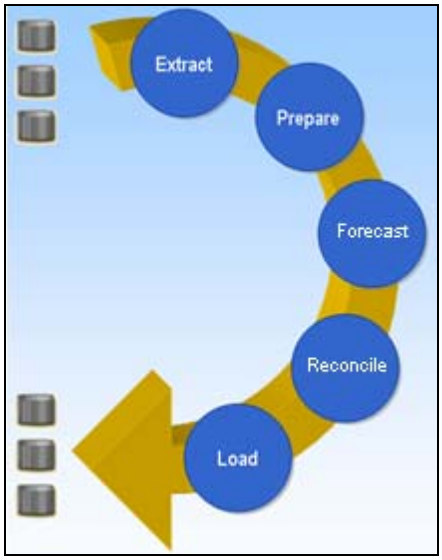


Figure 2. High-Level Flow of Forecasting Process

The following figure shows the execution flow within the SAS Demand Forecasting for Retail software components.

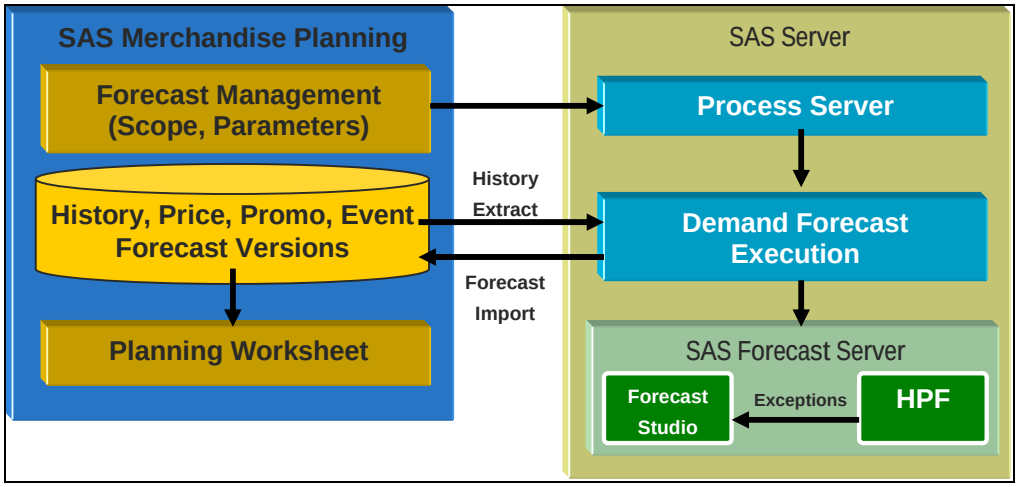


Figure 3. Execution Workflow within the Components of SAS Demand Forecasting for Retail

3.2 Components of the Solution

The following table lists the components of SAS Demand Forecasting for Retail.

Table 1. Solution Components

Component	Description
Forecast Project	Creates forecasting requests (projects) within SAS Merchandise Planning.
Execution Workflow	Provides details to scan an error log to return error codes to a scheduler.
Extract	Extracts merchandise history data into a SAS data set.
Forecast Execution	• Executes a single forecast job using SAS High Performance Forecasting. • Facilitates addition of independent variables for promotions, events, weather, and price. • Provides Model Quality Evaluation (Model Revisit) functionality. • Provides Reconciliation (generated at multiple levels).
Import	• Writes back forecasts into the Merchandise Planning system. • Stores multiple forecast versions. • Provides time hierarchy, synchronization, and aggregation.
SAS Merchandise Planning	Enables viewing or copying the forecast versions.

CHAPTER

4 Usage of Forecasts in Planning

4.1 Planning Scenarios

Preseason financial planning, in-season management, and preseason assortment planning scenarios are supported by SAS Demand Forecasting for Retail. The subsequent topics explain these scenarios in details.

4.1.1 Preseason Financial Planning

In a preseason financial plan, it is possible to reference one or more forecast versions and copy forecast data into working plans. Forecasts are most often generated for the lowest merchandise level (or all merchandise levels) in the financial plans that use the forecast. The time period should be set at ‘week’ or ‘month’ and the location should match the levels within the financial plans.

Forecasts across merchandise levels can be reconciled using a bottom-up (BU), top-down (TD), or middle-out (MO) approach. Multiple forecast versions such as Original (start of season forecast) and Periodic Snapshot versions (quarterly) can also be created.

The following figure displays two forecasts, Original and Current, with Last Year (LY) and Working Plan (WP) versions.

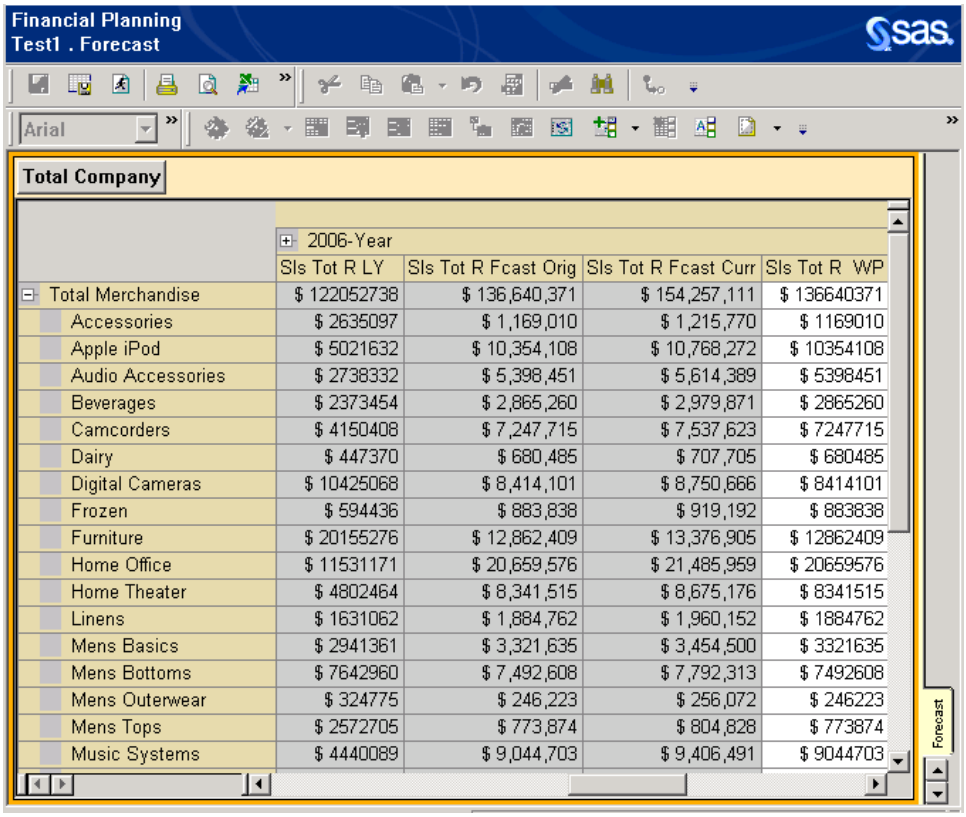


Figure 4. Original and Current Forecast Versions

4.1.2 In-Season Management

In an in-season management plan, forecast data can be grouped in batches and run on weekly basis to reforecast every time the system is loaded with new week's history. This enables the user to view a current forecast version, adjust the projection plan (working plan) in-season, and promptly respond to changes in the marketplace. User can also copy a reforecast to a plan.

The dimensions for an in-season forecast are mostly at the lowest level (or all levels) of merchandise, at the same locations for a weekly forecast.

4.1.3 Preseason Assortment Planning

SAS Demand Forecasting for Retail offers the following features to support the needs of assortment planning:

- cluster support
- new product forecasts
- direct run of the forecast for a plan

An assortment plan that uses a cluster-set requires a special plan-focused forecast. The batch forecast for all merchandise members in the assortment do not give correct results because clusters are different for different assortment plans. Additionally, the clusters might need addition or removal of merchandise members from the plan.

The solution supports clusters by pre-aggregating the store-level history to cluster-level and then forecasting at the cluster-level. This approach is more accurate than just forecasting at cluster-level and then aggregating the forecast from store to cluster because the store-level history can often be incomplete or inconsistent. It is best to aggregate the history up to the levels being planned before forecasting.

Decision to forecast at a particular level of merchandise is supported by various approaches. If the plan is for hard goods with a few new products then a bottom-up (BU) category to SKU forecast is

good. If it is a soft goods plan with many new products then a top-down (TD) category (the top-most level) to style is often the right approach.

For more information about creating new product forecasts, see chapter New Products in this guide.

In a new product forecast, the new products must have like-item products assigned to them before forecasts can be run. This enables the system to assign history (from similar selling products) to the new products. Additionally, the forecast must be top-down, so that the category-level forecast can be spread over a style-level forecast. The category-level forecasts are more accurate in terms of the total number of items per week, but the style-level forecasts maintain their relative sales patterns.

4.2 Referencing Forecast Data

Here is a sample financial plan for the category ‘Apparel’ with the working plan (in white cells), Last Year (LY) historical data, and two forecast version KPIs including original forecast units (Sls Tot R FC Orig) and current forecast units (Sls Tot R FC Curr). History plans and two reference forecast versions can be compared side-by-side. It is also possible to copy the forecast to plan, if required. See Copying Forecast Data for details.

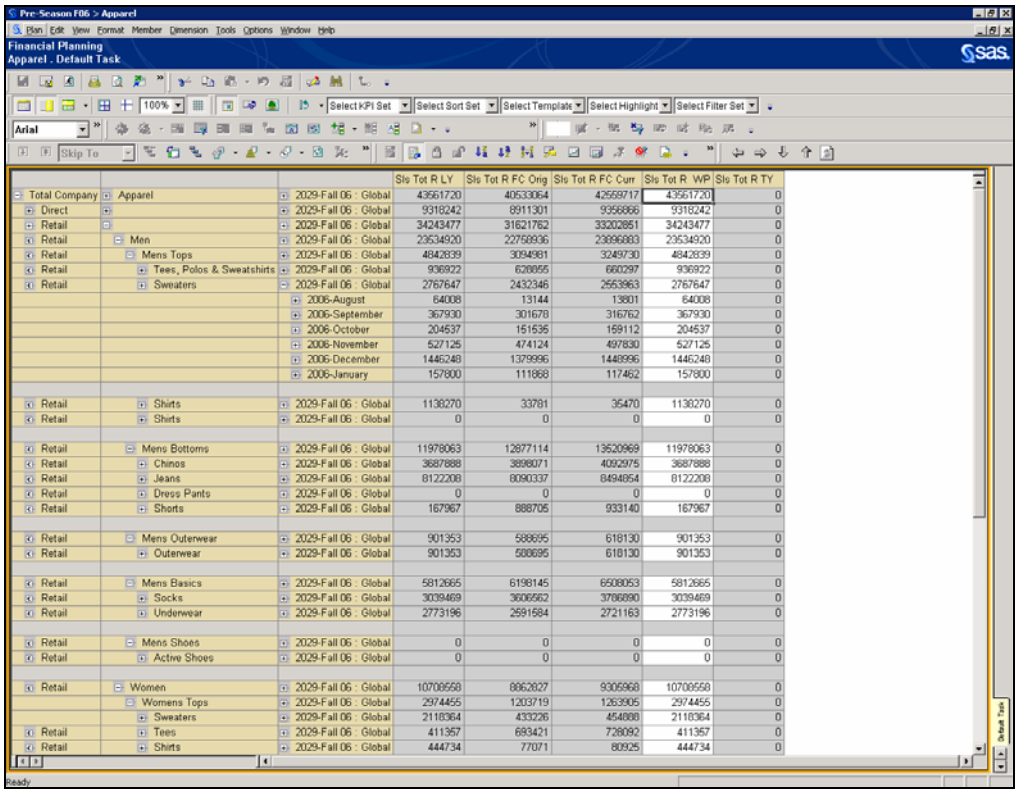


Figure 5. Sample of a Financial Plan for Apparel

4.3 Adding Forecast KPIs to KPI Sets and Process Templates

To add KPIs (key performance indicators) to an existing or a new KPI set so that forecast data can be referenced in the task pane:

1. Select one or more forecast versions.
   a. From the **KPI Set Editor**, select **Data Versions** tab.
   b. Select **Forecast** from the **Data Type** list and the appropriate version from the **Forecast Versions** list.

c. Select the appropriate time version from the **Time Versions** list. TY is selected to indicate 'This Year' or current forecast. TY means that no time-shift occurs in the data.

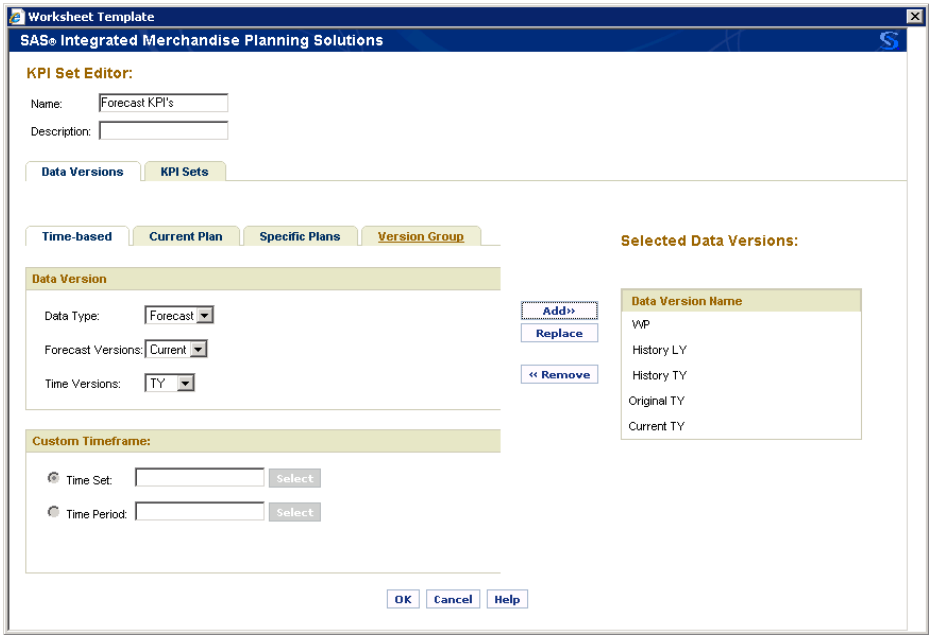


Figure 6. Selecting the Forecast Version

2. Select the forecast KPIs.

a. Select the **KPI Sets** tab and then a forecast data version from the **Data Versions** list.

b. Select the KPIs from the **Available KPIs** list.

c. Click **Add** to add the selected KPI to the list of KPIs appearing in the worksheet view.

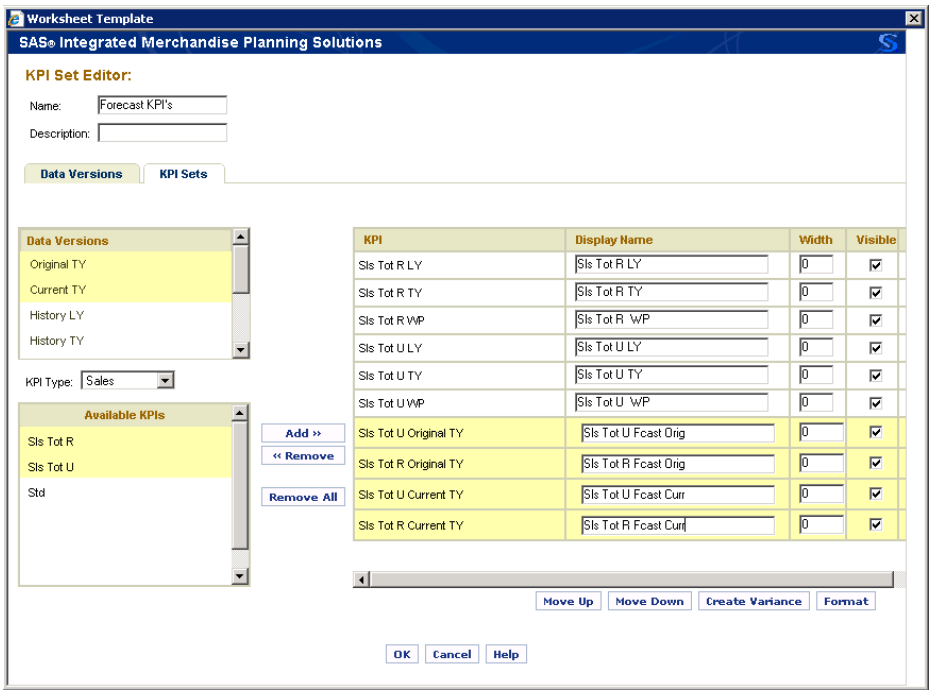


Figure 7. Selecting KPIs

4.4 Copying Forecast Data

The **Copy** function is used to copy a set of cells of the referenced forecast data to the working plan data in a plan worksheet.

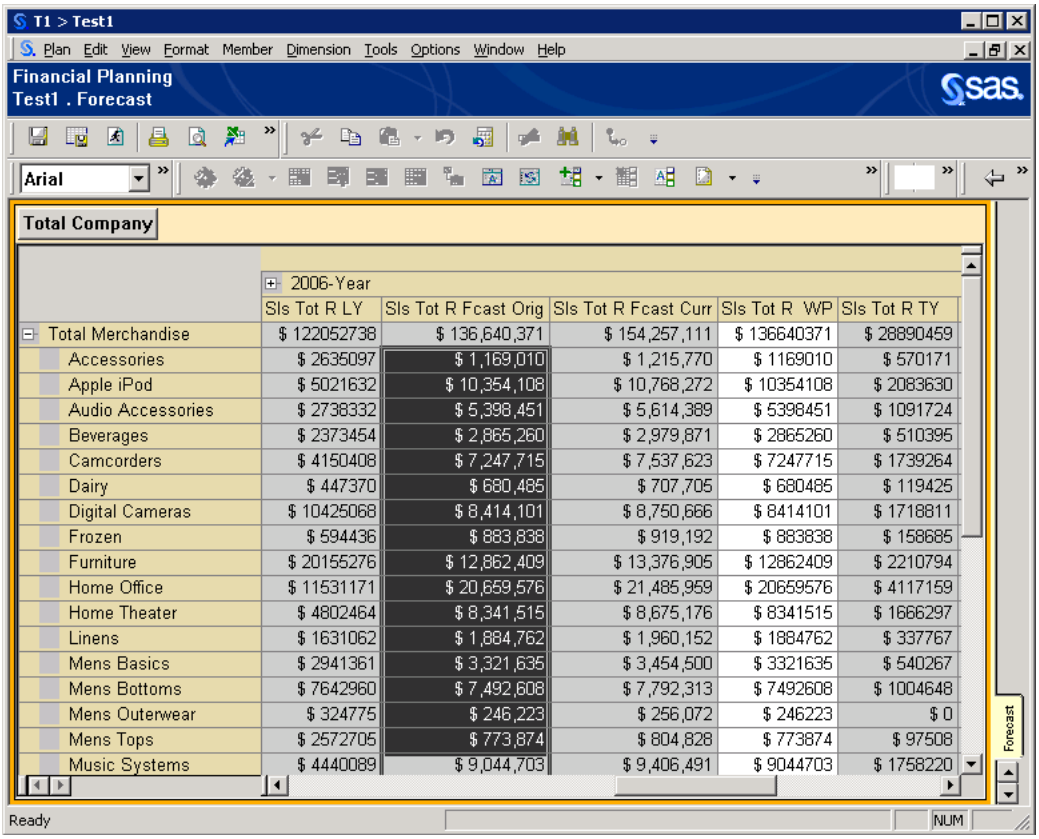


Figure 8. Copying a Set of Cells in a Plan Worksheet

The **Paste Special** function can be used to copy forecast data across all dimensions into a working plan for KPIs.

4.5 Calculating with Forecast Data

Working plan calculations can refer to, and include forecast data fields within them. The basic approach is to reference the forecast using a KPI Data Version ID in the calculation.

For more information about setting up these calculations, see the *SAS Merchandise Planning Metadata Guide*.

4.6 Seeding from a Forecast

To seed Plan worksheets with forecast KPIs:

1. Select **Forecast** from the list for the seeding rule **Data Type**.
2. Select a forecast version and time version from the **Forecast Version** and **Time Version** list. A second seeding rule such as History LY can also be added because forecasting provides only one or two KPIs. The second seeding rule can be used to fill in the other KPIs for a plan.

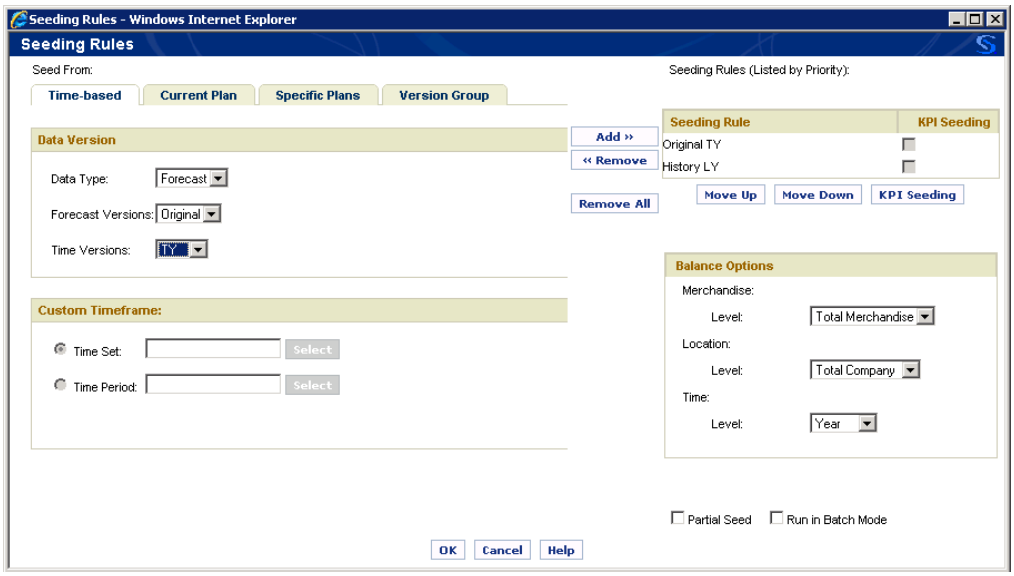


Figure 9. Setting-up Seeding Rules

To seed from a forecast, the metadata seeding rules must be set for the forecast table fields that are to be used for seeding.

CHAPTER

5 Forecast Creation

5.1 Defining a Forecast Project

A retail forecast is produced in SAS Demand Forecasting for Retail by defining a forecast project and then running it using the Execution Process Server. Projects can be defined and run online from the SAS Merchandise Planning interface or in batch mode.

A forecast project defines the time-series (using retail merchandise and locations), horizon of the forecast, forecast workflow, and the configuration parameters. Each project is initially defined by using SAS Merchandise Planning management screen. The parameters can be further customized from the job parameter data set.

Forecast projects are built in Financial or Assortment Plan Management system as worksheets. However, they are usually not opened as worksheets as they define just the scope and parameters for a forecast project.

The following figure shows three examples of forecast projects. Each project forecasts 52 weeks of the year 2008 and they are set to ‘Total Company’ and for ‘Total Merchandise’ to ‘Department’, ‘Category’, and ‘Sub-category’.

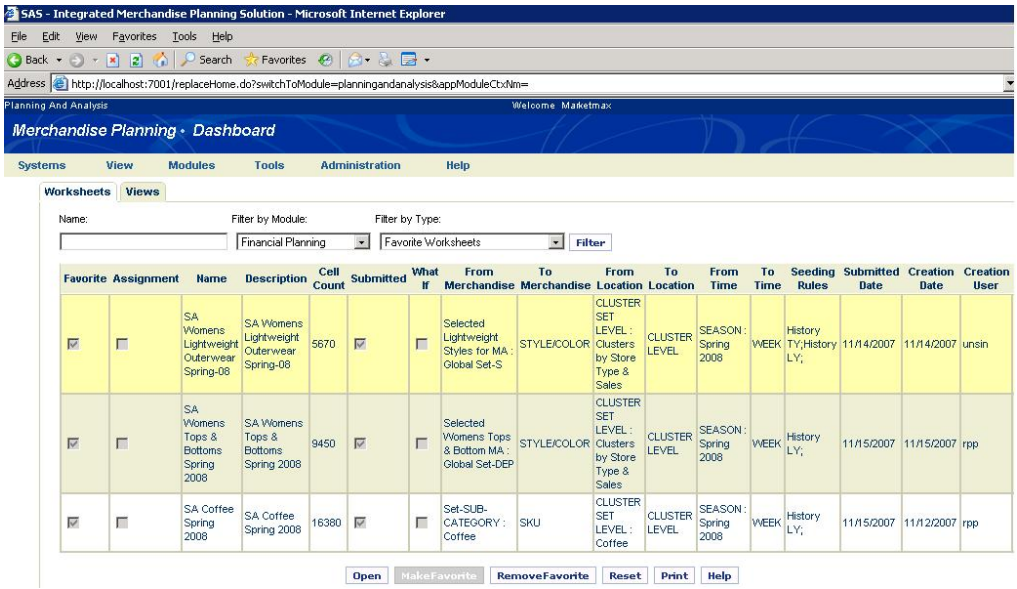


Figure 10. Examples of Forecast Projects

The following figure shows an example of a single forecast project worksheet definition.

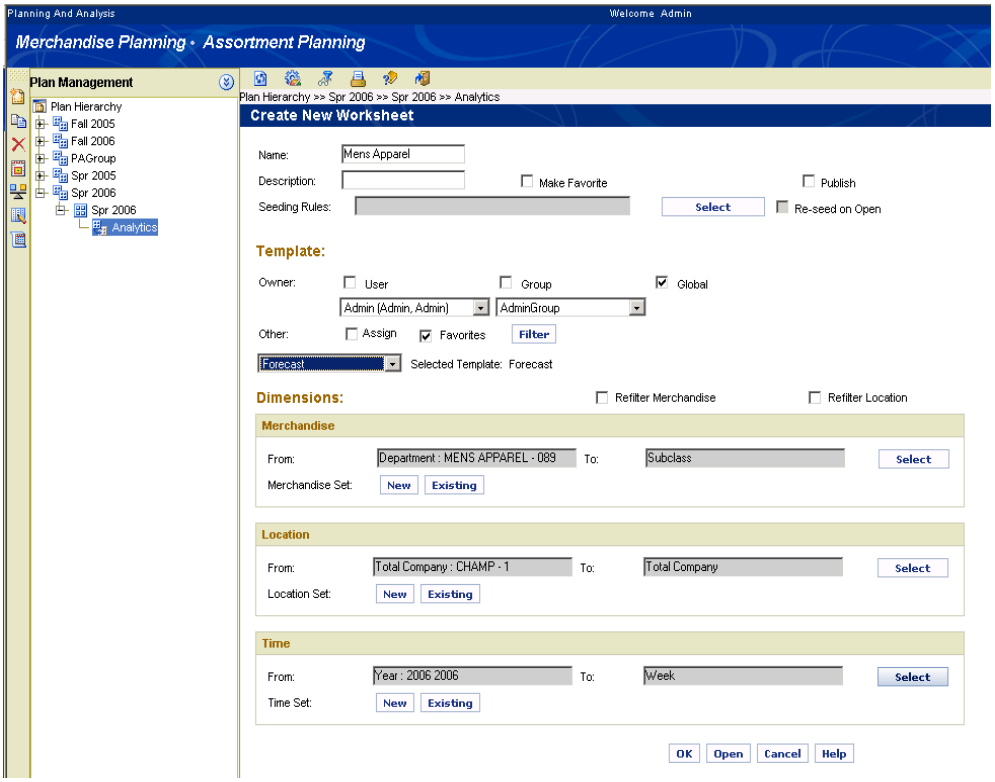


Figure 11. Example of a Forecast Job Definition

The Forecast Request Information that is explained in the following table provides summary of the information that is entered in the **Forecast Request** window.

Note: The forecast request is not intended to be opened as a worksheet.

The Jobparam parameters are changed in the default Jobparam data set and applied to all forecasts. If individual job parameters need to be set and there are no UI parameters, the following approach can be used to set these parameters for each job.

The jobparam_[projid] data set is stored in the data directory (by default this is **`mx_analytics/data`** on the SAS Server). The jobparam_[projid] data set does not appear until after the first run. Therefore, Jobparam data set is copied and is named jobparam_[projid], where projid is the project ID of the forecast job.

Table 2. Forecast Request Information

Parameter	Description
Name	Sets the forecast request name.
Description	Provide a description (optional).
Seeding Rule	Not required.
Template	Not required, use default.
Merchandise Dimension	Sets the merchandise boundaries for the forecast. The lowest level of these boundaries is used as the forecast scope. The data set at each level is forecasted if the reconcile dimension is set to be merchandise dimension. Use merchandise set to select individual merchandise members, if required. After a forecast is run, do not modify the parameters. Consider creating a new worksheet.

Parameter	Description
Location Dimension	Sets the location boundaries of the forecast. The lowest level of these boundaries is used as the forecast scope. The data set at each level is forecasted if the reconcile dimension is set to be location dimension. Use a location set to select individual location members, if required. After a forecast is run, do not modify the parameters. Consider creating a new worksheet.
Time Dimension	Sets the time boundaries for the forecast. The lowest level can be set to week, month, or quarter. All child members at the lowest level are forecasted. Use a static time set to define the forecast timeframe, if required. After a forecast is run, do not modify the parameters. Consider creating a new worksheet.

Jobparam data set contains input parameters that are required for the execution of SAS Demand Forecasting for Retail. The values of these input parameters can be set as according to the requirements. This data set contains parameters for all areas like Data Preparation, Data Extract, Splitter Process, HPF, and so on. To override the default values of parameters, modifications are done in the job-specific Jobparam data set. For example, if modifications are required in worksheet number 3000, then changes are done in jobparam_3000.sas7bdat.

Note: The default Jobparam data set is mandatory for running any forecast request.

5.2 Editing the Forecast Request Parameters

To edit forecast request parameters for a forecast project worksheet:

1. Click **Forecast** in the left panel of the **Merchandise Planning - Financial Planning** window to select it, and then click **Business View**.

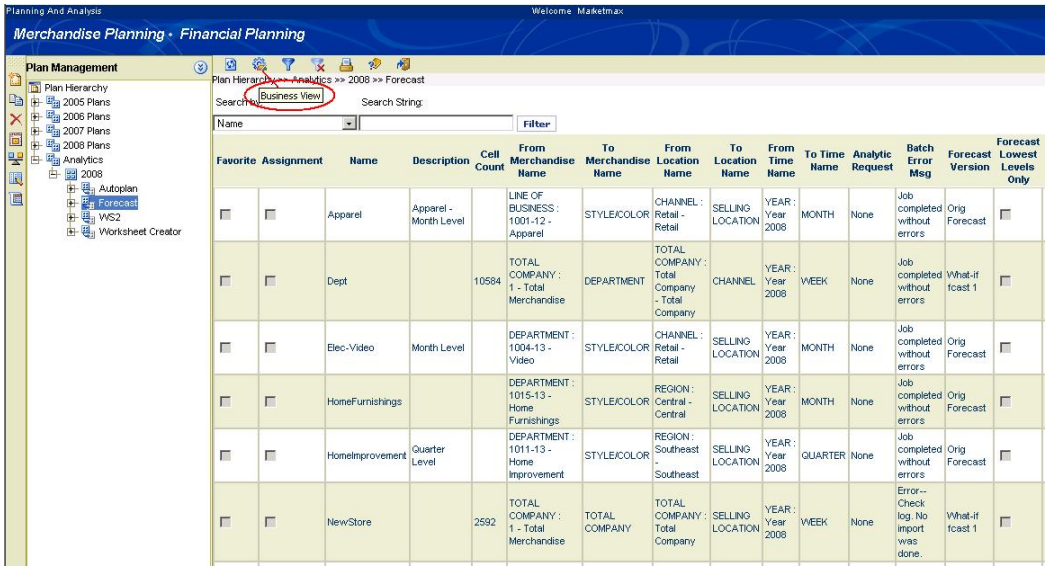


Figure 12. Merchandise Planning - Financial Planning Window

2. The **Business View Manager** window displays a list of the available worksheets. Select the forecast project worksheet from this list, and click **OK**.

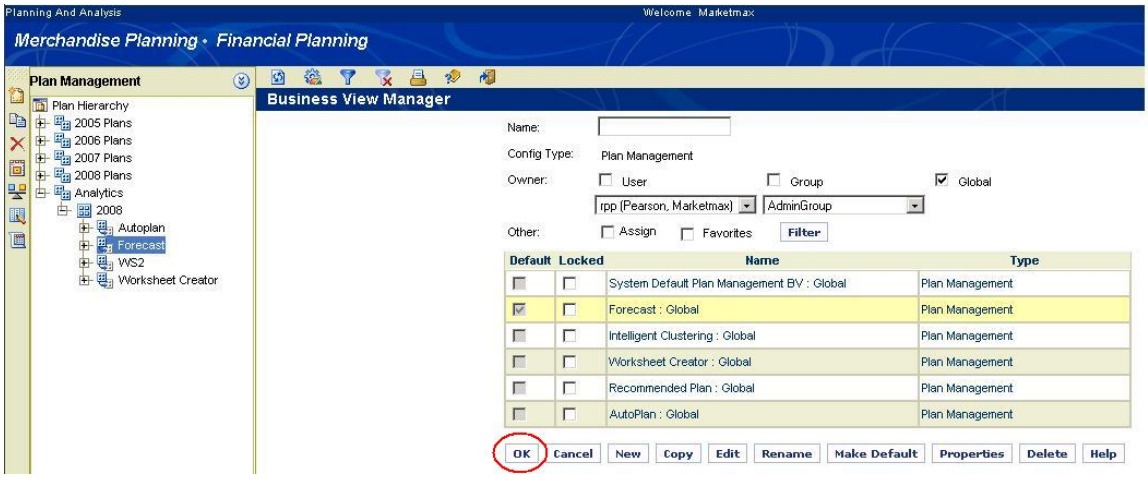


Figure 13. Business View Manager

3. After forecast business view is selected as active view, select the forecast project worksheet and click **Edit**.

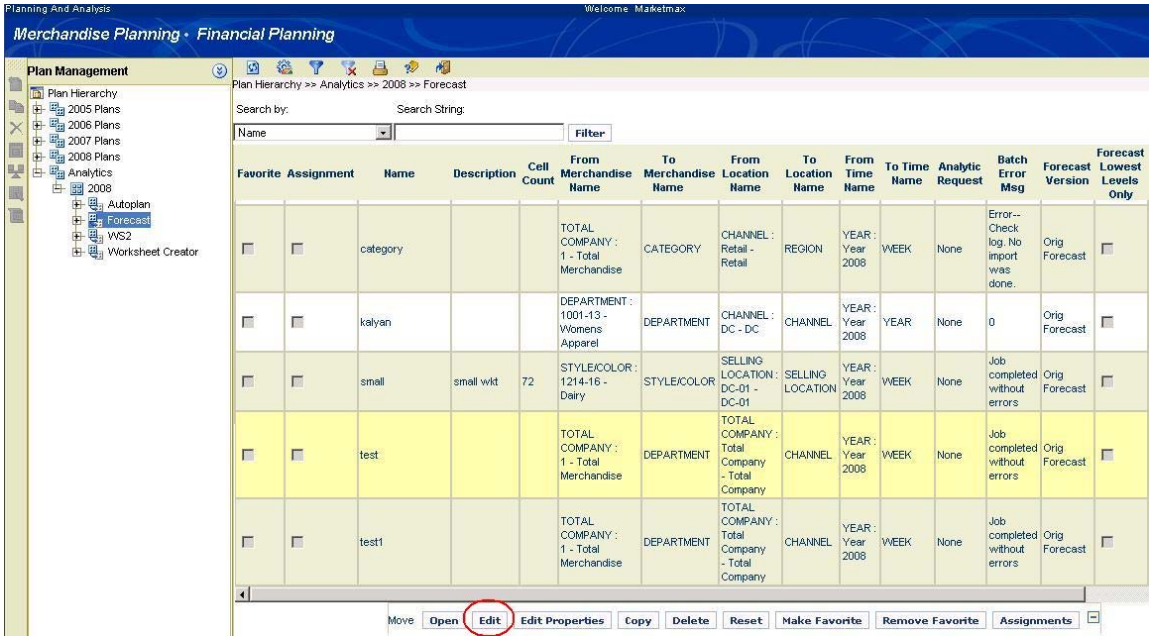


Figure 14. Editing a Forecast Project

4. The **Forecast Request Parameter** window appears.

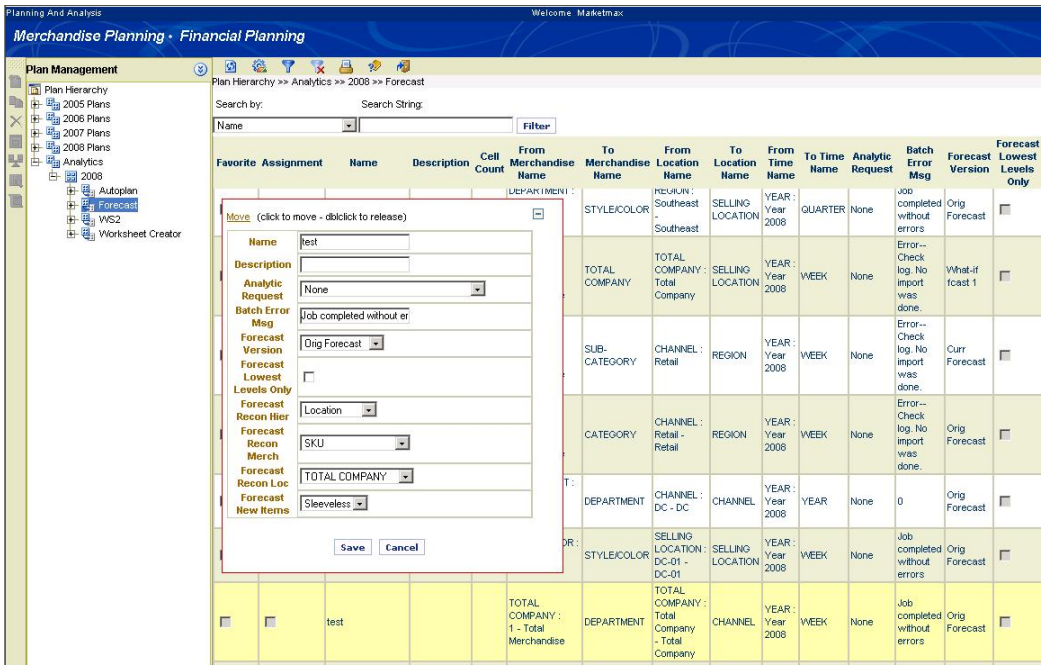


Figure 15. Forecast Request Parameter Window

The table below lists the forecast request parameters and their values.

Table 3. Forecast Request Parameters

Parameter	Value	Description
Analytic Request	Set this to execute on-demand. Set to “Forecast Sales Units” or “Forecast Sales $”.	Runs the forecast request.
Forecast Version	Set through lookup from the table maxdata.fcast_version. For example, FC1, FC2, FC3, and so on. The actual value passed to the request is numeric such as 1, 2, or 3.	Forecast is stored in the forecast version specified.
Reconciliation	Set the parameters for reconciliation. For example, reconcile hierarchy, reconcile merchandise level, reconcile location level, and reconcile lowest level only.	See Reconciliation functionality description in this document.
Effects	Set options like Promotions, Events, Weather, and Price parameters to add forecast effects to the forecast. Promotions can have other effect options such as halo/cannibalization, pantry, and delay.	When included in a forecast request screen these effects are considered while forecasting, thus generating a Value Added Forecast (FVA).
Forecast New Item	Any one of the following four options can be selected: **Merge History before Forecast:** This option creates like-item history, merges it with history for the regular (non-new) items, and then forecast both new and regular items using merged history. **Merge History with Forecast:** This option forecasts regular items and then shifts new-item like-history to the forecast periods. Finally, it	Select to forecast new products that have been defined with ‘acts like’ products. System will substitute history of acts like product for new product(s).

Parameter	Value	Description
	merges with the regular item forecasts before the reconcile process. **All History:** This option merges all history for new and regular items and imports it into the system without statistical forecasting. The timeframe of the imported items is historical and not the worksheet timeframe. This feature generates like-item history and combines this history with normal regular item history that can be referenced or seeded in the planning process. **New Product History:** This is similar to the 'All History' option but returns only like-history for new products.	

5.3 Forecast Versions

Multiple versions or snapshots of a forecast are possible for the same merchandise, location, and time dimensions. Examples of these versions are FP Original, FP Current, and FP Qtr1. These three versions hold an original forecast, the current active forecast, and a snapshot of the forecast, as it appears at the start of first fiscal quarter. Versions might also be needed to store multiple forecasts where parameters of the forecast differ. For example, where one forecast is reconciled and the other is not.

Once created, multiple forecast version data can then be compared within the plan worksheets.

5.4 Reconciliation

Forecast data can be reconciled across merchandise or location levels after being forecasted at merchandise or location level. The product supports the following reconciliation options and their corresponding parameters.

Table 4. Reconciliation Options and their Parameters

Reconcile Option	Description	Parameters
None	All levels of merchandise or location are forecasted but not reconciled.	RECON HIER=Merchandise or Location LOWEST LEVELS ONLY= Off RECON_MERCH=None RECON_LOC=None
Single Level Forecast	Lowest level of merchandise and location are forecasted.	LOWEST LEVELS ONLY=On
Merchandise Bottom-Up	Forecast aggregated up from lowest level to top level of merchandise.	RECON HIER=Merchandise LOWEST LEVELS ONLY=Off RECON_MERCH=lowest level of merchandise in forecast project RECON_LOC= None
Merchandise Top-Down	Forecast disaggregated down from top level to lowest level of merchandise.	RECON HIER= Merchandise LOWEST LEVELS ONLY= Off RECON_MERCH= top level of merchandise in forecast project

Reconcile Option	Description	Parameters
		RECON_LOC= None
Merchandise Middle-Out	Forecast aggregated up and disaggregated down from a merchandise level in the middle.	RECON HIER= Merchandise LOWEST LEVELS ONLY=Off RECON_MERCH= middle level of merchandise in forecast project RECON_LOC=None
Location Bottom-Up	Forecast aggregated up from lowest level to top level of location.	RECON HIER=Location LOWEST LEVELS ONLY=Off RECON_MERCH= None RECON_LOC= lowest level of location in forecast project
Location Top-Down	Forecast disaggregated down from top level to lowest level of location.	RECON HIER=Location LOWEST LEVELS ONLY=Off RECON_MERCH= None RECON_LOC= top level of location in forecast project
Location Middle-Out	Forecast aggregated up and disaggregated down from a location level in the middle.	RECON HIER=Location LOWEST LEVELS ONLY=Off RECON_MERCH= None RECON_LOC= middle level of location in forecast project

5.5 Forecast and Reconciliation

The level scope of merchandise, location, and the reconciliation parameters determine the levels of merchandise and location that are forecasted and reconciled. The lowest time level that is specified in the forecast job is forecasted. The system uses all the history that is loaded for the forecast.

This solution supports reconciliation for Top-down (TD), Bottom-up (BU), and Middle-out (MO) options across the merchandise or location dimensions. It does not support reconciliation across the time dimension.

The examples given in the following table assumes that the merchandise levels 1 to 5 and location levels 1 to 3 are specified in the scope of the forecast job.

Table 5. Merchandise and Location Levels

Recon_hier	Recon_loc	Recon_merch	Recon_single	Forecasted	Reconciled
Merch	None	None	On	Merch 5, Loc 3 only since recon_single is On	Nothing
Merch	None	None	Off	Merch 1-5 at Loc 3	Nothing
Merch	None	Level 1	Off	Merch 1-5 at Loc 3	Merch Top-Down
Merch	None	Level 5	Off	Merch 1-5 at Loc 3	Merch Bottom-Up
Merch	None	Level 3	Off	Merch 1-5 at	Merch Middle-out

Recon_hier	Recon_loc	Recon_merch	Recon_single	Forecasted	Reconciled
				Loc 3	from level 3
Loc	None	None	Off	Loc 1-3 at Merch 5	Nothing
Loc	Level 1	None	Off	Loc 1-3 at Merch 5	Loc Top-Down
Loc	Level 3	None	Off	Loc 1-3 at Merch 5	Loc Bottom-Up
Loc	Level 2	None	Off	Loc 1-3 at Merch 5	Loc Middle-Out from level 2

To present a complete cube of the forecast data, following bottom-up (BU) aggregations are run after HPF_RECONCILE procedure:

- Dimension not being reconciled through HPF_RECONCILE is aggregated bottom-up.
- Time dimension is always aggregated bottom-up from the lowest time level.

To provide a single level of forecast (not a complete cube) set parameter RECON_SINGLE = On.

CHAPTER

6 New Products

6.1 Creating a New Product Forecast

To initiate forecast for a new product:

1. Create a manual entry of one or more like-item assignments through SAS Merchandise Planning worksheet editor for each new style or item.
2. Create like-item history.
3. Forecast or merge like-item history into the forecast.

Example entries of like-item parameters are shown in the following figure.

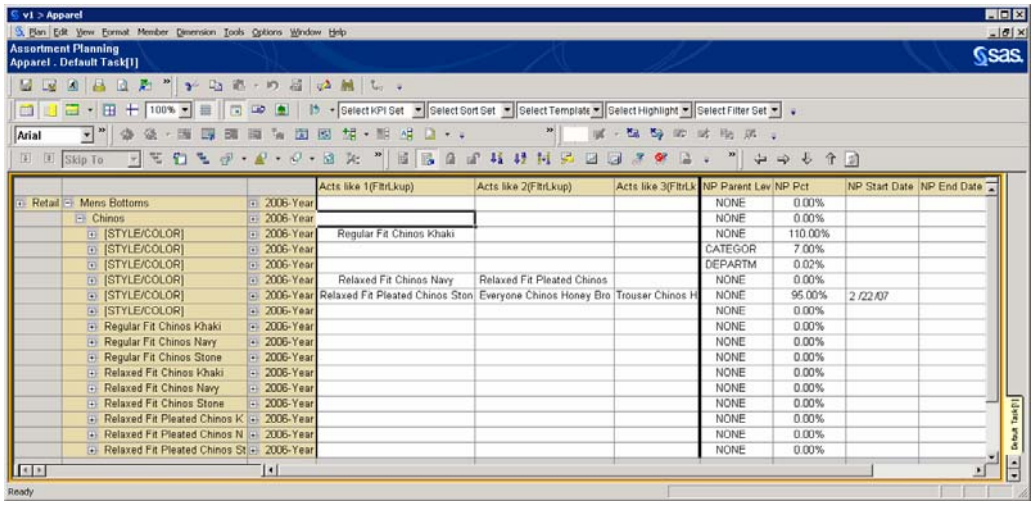
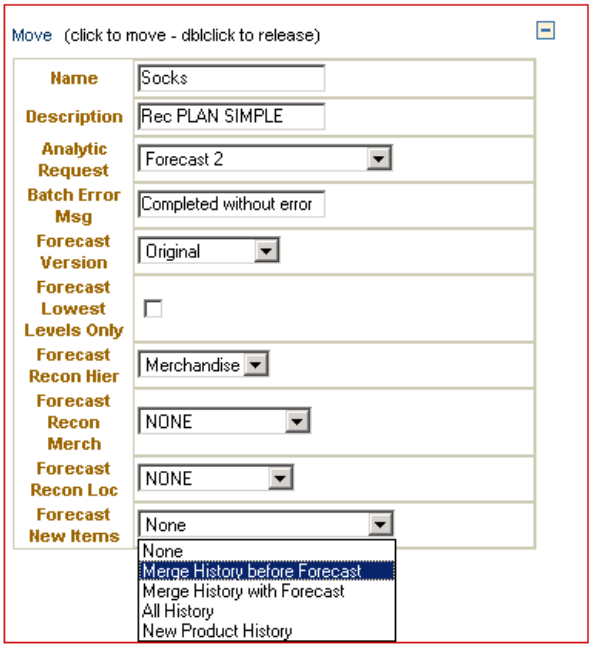


Figure 16. Like-Item Parameters

6.2 Running New Product Forecasts

The user interface in SAS Merchandise Planning has a parameter that instructs the system to forecast by using new product like-item assignments.

The parameter DFR_WF_NEWITEM is set to '1' in Jobparam data set to enable loading new product history. This takes the acts-like history data and merges it with normal product history data before forecast is run.



Note: The new product forecast applies only to the lowest level of merchandise within a forecast job request.

6.2.1 New Forecast Selection

User can select any of the following options from the **Forecast New Items** list (shown in the previous figure):

- **None:** This is the default option. If this option is selected, new product assignment records are not considered. Also, no function is performed on these records. If there are new product records, they are not forecasted because such records do not have a history.
- **Merge History before Forecast:** This option creates like-item history, merges this history with history for the regular (non-new) items, and then forecast both new and regular items using the merged history.
- **Merge History with Forecast:** This option forecasts regular items, then shifts new-item like-history to the forecast periods, and finally merges it with the regular item forecasts before reconcile process.
- **All History:** This option merges all history for new and regular items, and imports this history into the system without statistical forecasting. The timeframe of items that are imported is historical and not the forecasted. This feature generates like-item history and combines this history with normal regular item history that can be used in the planning process.
- **New Product History:** This option includes only history for new items and imports it into the system without statistical forecasting. The timeframe of the items that are imported is historical and not forecasted.

6.3 New Store Forecasts

A new store can be assigned a like-store. This assignment is set at the store level. If used, this function substitutes like-store data for the new store for all merchandise data.

The store attribute that sets the like-store is called Acts-like. This attribute can be set from the **Location Attributes Selection** screen in **KPI Editor** at store level.

Set the value of parameter DFR_WF_NEWLOC to ‘1’ in Jobparam data set (or as a default in Jobparam data set) to activate this function.

CHAPTER 7

Functionality of Demand Forecasting for Retail

7.1 Parameters

The solution provides Jobparam data set, which is a single SAS data set that contains all the parameters for a job. The parameters in this data set controls diagnose and execution of a forecast, overall workflow of the process, metadata, printing, filtering, and other forecast features. The user interface (UI) parameters are also copied to this data set when a job is executed.

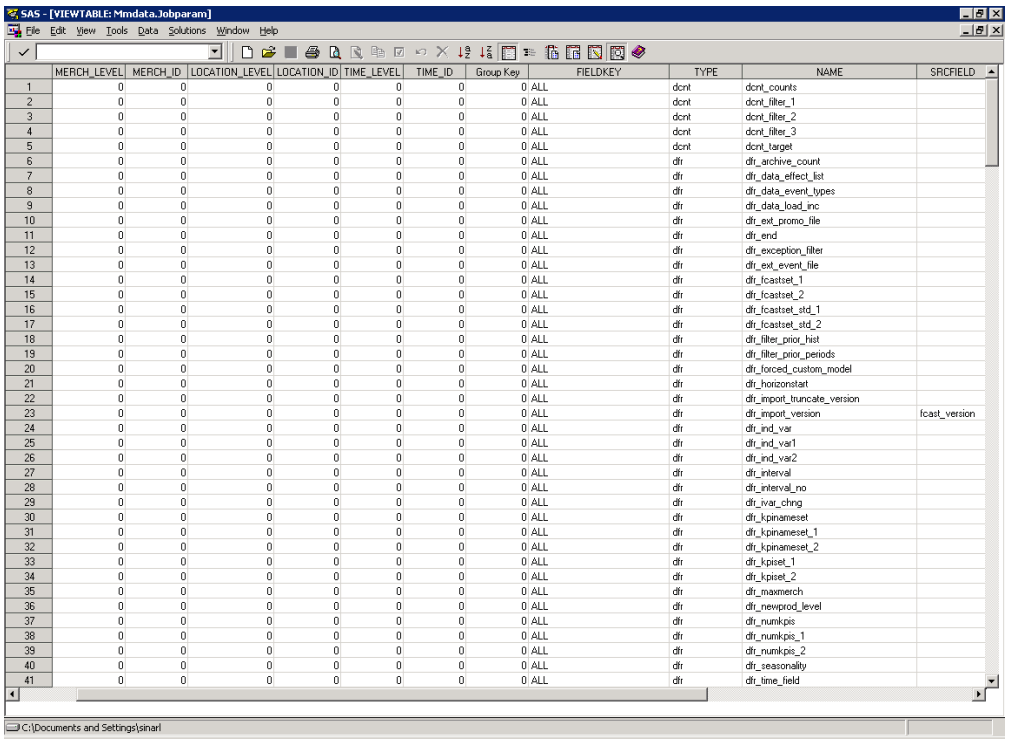
Each job has its own set of parameters. These parameters are assigned default values that can be overridden at the job or merchandise level, location level, time level, and KPI level depending on the parameter type.

The default parameters are set in **`/data/jobparam`** data set file. These parameters are used by all forecast jobs unless overridden for each job. Each job produces a Jobparam data set with the filename, jobparam_xxxx where `xxxx` is the projid (or jobid).

During initial implementation of the solution, following tasks are recommended:

- Review default parameter values and set them according to the customer needs.
- Review and modify (if required) metadata fields for source and target forecast fields.
- Review source field (**`srcfield`**) if new UI elements are required. Setting the source field lets parameters in Planworksheet table to be automatically copied to Jobparam data set and made available as parameters for the forecast.

The data set contains one row for each parameter. The following figure is an example of a data set.



	MERCH_LEVEL	MERCH_ID	LOCATION_LEVEL	LOCATION_ID	TIME_LEVEL	TIME_ID	Group Key	FIELDKEY	TYPE	NAME	SRCFIELD
1	0	0	0	0	0	0	0	ALL	dcnt	dcnt_counts	
2	0	0	0	0	0	0	0	ALL	dcnt	dcnt_filter_1	
3	0	0	0	0	0	0	0	ALL	dcnt	dcnt_filter_2	
4	0	0	0	0	0	0	0	ALL	dcnt	dcnt_filter_3	
5	0	0	0	0	0	0	0	ALL	dcnt	dcnt_target	
6	0	0	0	0	0	0	0	ALL	dfr	dfr_archive_count	
7	0	0	0	0	0	0	0	ALL	dfr	dfr_data_effect_list	
8	0	0	0	0	0	0	0	ALL	dfr	dfr_data_event_types	
9	0	0	0	0	0	0	0	ALL	dfr	dfr_data_load_inc	
10	0	0	0	0	0	0	0	ALL	dfr	dfr_ext_promo_file	
11	0	0	0	0	0	0	0	ALL	dfr	dfr_end	
12	0	0	0	0	0	0	0	ALL	dfr	dfr_exception_filter	
13	0	0	0	0	0	0	0	ALL	dfr	dfr_ext_event_file	
14	0	0	0	0	0	0	0	ALL	dfr	dfr_fcastset_1	
15	0	0	0	0	0	0	0	ALL	dfr	dfr_fcastset_2	
16	0	0	0	0	0	0	0	ALL	dfr	dfr_fcastset_std_1	
17	0	0	0	0	0	0	0	ALL	dfr	dfr_fcastset_std_2	
18	0	0	0	0	0	0	0	ALL	dfr	dfr_filter_prior_hist	
19	0	0	0	0	0	0	0	ALL	dfr	dfr_filter_prior_periods	
20	0	0	0	0	0	0	0	ALL	dfr	dfr_forced_custom_model	
21	0	0	0	0	0	0	0	ALL	dfr	dfr_horizonstart	
22	0	0	0	0	0	0	0	ALL	dfr	dfr_import_truncate_version	
23	0	0	0	0	0	0	0	ALL	dfr	dfr_import_version	fcast_version
24	0	0	0	0	0	0	0	ALL	dfr	dfr_ind_var	
25	0	0	0	0	0	0	0	ALL	dfr	dfr_ind_var1	
26	0	0	0	0	0	0	0	ALL	dfr	dfr_ind_var2	
27	0	0	0	0	0	0	0	ALL	dfr	dfr_interval	
28	0	0	0	0	0	0	0	ALL	dfr	dfr_interval_no	
29	0	0	0	0	0	0	0	ALL	dfr	dfr_ivar_chng	
30	0	0	0	0	0	0	0	ALL	dfr	dfr_kpinameset	
31	0	0	0	0	0	0	0	ALL	dfr	dfr_kpinameset_1	
32	0	0	0	0	0	0	0	ALL	dfr	dfr_kpinameset_2	
33	0	0	0	0	0	0	0	ALL	dfr	dfr_kpiset_1	
34	0	0	0	0	0	0	0	ALL	dfr	dfr_kpiset_2	
35	0	0	0	0	0	0	0	ALL	dfr	dfr_maxmerch	
36	0	0	0	0	0	0	0	ALL	dfr	dfr_newprod_level	
37	0	0	0	0	0	0	0	ALL	dfr	dfr_numkpis	
38	0	0	0	0	0	0	0	ALL	dfr	dfr_numkpis_1	
39	0	0	0	0	0	0	0	ALL	dfr	dfr_numkpis_2	
40	0	0	0	0	0	0	0	ALL	dfr	dfr_seasonality	
41	0	0	0	0	0	0	0	ALL	dfr	dfr_time_field	

Figure 17. Sample Job Parameter Data Set

The following table lists the fields in each parameter (row).

Table 6. Parameter Database Table

Field	Description	Sample Value Job Level	Sample Value Forecast Level
Projid	This is a mandatory field that signifies the ID of the project or worksheet. The value for this field is defaulted to '0' for the default project.	2252	2252
Type	This is a mandatory field. The value in this field must be in lowercase.	dfr	dfr
Name	This is a mandatory field and signifies the parameter name. The parameter name must be exactly as in the default job level parameter.	dfr_wf_extract	diag_criterion

Field	Description	Sample Value Job Level	Sample Value Forecast Level
Value	This is value of the Name field.	1	MAPE
Fieldkey	The value in this field must be ALL or a text string from the dfr_kpinameset_x field(s). The value in this field must be in all caps.	All	Sales
Merch_Level	Merchandise level. Possible Values: 1—10 or 0 if not set.	0	5
Location_Level	Time level. Possible Values: 1 to 4 or 0 if not set.	0	3
Time_Level	Location level Possible Values: 47 to 51 or 0 if not set.	0	51
Merch_ID	Lvxctree_id Possible Value: 0 if not set.	0	1001
Location_ID	Lvxloc_id Possible Value: 0 if not set.	0	1123
Time_ID	Lvxtime_lkup_id Possible Value: 0 if not set.	0	4124
Group_Key	The group key number	0	2
Srcfield	Planworksheet source field. Included if the UI parameter from the Planworksheet table is to be copied to the parameter at job execution.	fcast_wf_extract	fcast_wf_extract
Description	Describes the source field. This field has default value that can be changed by the job if desired.	Workflow flag for extract process	Workflow flag for extract process

7.1.1 UI Parameters

Any job level parameter listed in Jobparam data set can be provided to the user in the user interface (UI) of SAS Merchandise Planning. UI field data is stored in the table maxdata.planworksheet. If provided, the **`SRCFIELD`** in the Jobparam data set must be filled with corresponding field name in planworksheet table. The fields are then automatically copied. Some source fields are already provided in the UI. For example, the field **`fcast_version`** holds the forecast version that a user can select.

The UI can provide data as a text field, check box, or drop-down list box with values from a lookup table. See *SAS Merchandise Planning internal metadata document Metadata_611*, that is available in the *SAS Retail Documentation Group*, to set new fields and lookup tables.

7.1.2 Workflow Parameters

Workflow parameters control the job workflow. These parameters are at job level and cannot be overridden at merchandise or location level and KPI level.

The running of a workflow parameter depends on its value, which can either be '0' or '1'. A parameter executes the workflow steps only if it has value '1', a parameter that has value '0' is not executed.

Table 7. Workflow Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_WF_EXTRACT	1= Extract data from database	1
dfr	DFR_WF_DATAPREP	Cluster aggregation and time fills 0 = No data preparation is done 1= Fill missing time periods with zeros	1
dfr	DFR_WF_EXECMETHOD	1= Normal forecast execution Executes diagnose, select, fit, forecast steps	1
dfr	DFR_WF_DIAGNOSE	Diagnose Step	1
dfr	DFR_WF_SELECT	Select Step	1
dfr	DFR_WF_FIT	Fit Step	1
dfr	DFR_WF_FORECAST	Forecast Step	1
dfr	DFR_WF_REVISIT	Revisit Step	0
dfr	DFR_WF_RECON	Reconciliation Step	1
dfr	DFR_WF_ARCHIVE	Forecasts Archival Step	1
dfr	DFR_WF_IMPORT	Forecasts Import Step-Needs database	1
dfr	DFR_WF_PURGE	Purge Temporary Files	0
dfr	DFR_WF_FORECAST_INTEL_EXE	Intelligent forecast workflow controls diagnose, select, fit, and forecast execution flows that are based on previous runs. It ensures that only valid workflow occurs in the SAS High-Performance Forecasting steps.	1
dfr	DFR_WF_SPLIT_TYPE	This workflow parameter is used to increase the performance of SAS Demand Forecasting for Retail. 0 = No Performance Split 1 = Data set Split based on number of Time Series	0

Analytic Key	Parameter Name	Description	Default Value
		2 = Data set Split based on number of data sets	
dfr	DFR_WF_OPTIMIZED	Used for production optimized extract, log options, file size	0
dfr	DFR_WF_GRID	0: Grid disabled 1: Grid is enabled	0
dfr	DFR_NO_TS	If Performance Split is On and split process is based on number of Time Series, then dfr_no_ts attribute is used. 0= No split, or positive integer number	0
dfr	DFR_NO_DS	If Performance Split is On and split process is based on number of data sets, then dfr_no_ds attribute is used. 0= No split, or positive integer number	0
dfr	DFR_REMOVE_EST_CAT	0=est and catalog are not deleted before forecasting. 1=all data sets including est and catalogs are deleted before forecasting	0
dfr	DFR_RUNNING_MODE	0=Production Mode 1=Testing Mode	0
dfr	DFR_USER_DEFINE_GK	0=User defined group key is disabled 1=User defined group key is enabled	0
dfr	DFR_FORCED_CUSTOM_MODEL	Controls the NODIAGNOSE option of HPFDIAGNOSE	0

7.1.3 New Product Parameters

The following table lists new product parameters.

Table 8. New Product Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_WF_NEWITEM_EXTERNAL	0 = Off 1 = Load New items from external file	0
dfr	DFR_WF_NEWITEM	0=None 1=Merge History before Forecast 2=Merge History with Forecast	0

Analytic Key	Parameter Name	Description	Default Value
		3=All History 4=New Product History	
dfr	DFR_WF_NEWLOC	1= Use location substitution history for new locations 0 = Do not use substitute history for new locations	0
dfr	DFR_NEWPROD_LEVEL	Working field set by system Merchandise level where new product assignments occurs	6

7.1.4 Filtering Parameters

Filtering parameters are job level and *cannot* be overridden at merchandise or location level, and KPI.

Table 9. Filtering Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_EXCEPTION_FILTER	Specifies the records to extract for review. If blank the exception filter will not be executed.	MAPE>50
dfr	DFR_FILTER_PRIOR_PERIODS	Number of total periods for filter.	52
dfr	DFR_FILTER_PRIOR_HIST	Number of periods that must have history of total period in PRIOR_PERIODS parameter.	26
dfr	DFR_FILTER_START_DATE	Stand-alone start date for filter.	26MAY2008
dfr	DFR_FILTER_END_DATE	Stand-alone end date for filter.	02NOV2008

7.1.5 Reporting and Archive Parameters

These parameters are job level and *cannot* be overridden at merchandise or location level and KPI level.

Table 10. Reporting and Archive Parameters That Cannot Be Overridden

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_ARCHIVE_COUNT	Number of periods to save archive	3

The following parameters can be overridden at the merchandise or location level and KPI.

Table 11. Reporting and Archive Parameters That Can Be Overridden

Analytic Key	Parameter Name	Description	Default Value
dfr	DIAG_PRINT	Specifies the printed output desired	None

Analytic Key	Parameter Name	Description	Default Value
dfr	ENG_PLOT(EXPERIMENTAL)	Specifies the graphical output desired	None
dfr	ENG_PRINT	Specifies the printed output desired	None

7.1.6 Metadata Parameters

The metadata parameters describe the source and target database fields to be used during integration logic for extract and import.

These metadata parameters are job level and *cannot* be overridden at merchandise or location level and KPI.

Forecast Field Lists

The parameter DFR_KPINAMESET_1 contains a unique name of the forecast fields. This name is limited to six characters. Each field in this list has a corresponding source field in DFR_KPISET_1, forecast datastore field in DFR_FCASTSET_1, and a standard deviation field in DFR_FCASTSET_STD_1. A field in DFR_FCASTSET_STD_1 is required *only* if revisit logic is being used.

Forecast Type

Forecast type is the ability to define different sets of KPIs to forecast. It enables the user to select the set of KPIs to use. The user selects these KPIs from batch_analytic_lkup table. Valid forecasting in batch_analytic_lkup is 1 to 9.

Each analytic type has corresponding metadata fields that are designated by _1, _2, and so on. If analytic ID is set to '1' the first set of values is used, if analytic ID is set to '2', then the second set of values is used, and so on. Metadata for _1 and _2 are provided, but similar fields can be added to parameter data set up to _9.

Source Data Versions

Any fact table in Merchandise Planning can be used for source data. For example, Mfinc, mcomp, minventory, and so on. The table name is included in the metadata field, dfr_kpiset_*.

Table 12. Metadata Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_NUMKPIS_1	Number of forecast KPIs.	2
dfr	DFR_KPINAMESET_1	MMAX forecast KPI business name. Space delimited list and limited to six characters. Each name must be unique.	sales units
dfr	DFR_KPISET_1	Corresponding MMAX source database table.field. Comma delimited list.	mfinc.posit_sales mfinc.posit_mvmt
dfr	DFR_FCASTSET_1	Corresponding MMAX target database mfact_version field. Comma delimited list.	mfact_version.fc_sales_rmfact_version.fc_sales_u
dfr	DFR_FCASTSET_STD_1	Corresponding import std deviation fields. Comma delimited list.	mfact_version.fc_std mfact_version.fc_std_u

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_NUMKPIS_2	Second set of KPIs, if forecast type = 2. Number of forecast KPIs.	1
dfr	DFR_KPINAMESET_2	MMAX forecast KPI business name. No spaces allowed.	sales
dfr	DFR_KPISET_2	MMAX source database table.field.	mfinc.posit_sales
dfr	DFR_FCASTSET_2	MMAX target database mfact_version field.	mfact_version.fc_sales_r
dfr	DFR_FCASTSET_STD_2	Corresponding import standard deviation fields.	mfact_version.fc_std
dfr	DFR_DATA_EFFECT_LIST	List with one or more of: promo halo pantry price weather delay.	No value here
dfr	DFR_NON_NEGATIVE_FORECAST	0=Negative forecasts will be processed as it is. 1= Negative forecasts will be modified to the zero and then processed.	0
dfr	DFR_ZERO_FILL_EXTRACT	Fills the missing KPI values with the provided value.	0
dfr	DFR_ZERO_FILL_FILTER	Filters the KPI values that are greater than the provided value.	0

7.1.6.1 Troubleshooting

If HPFDIAGNOSE generates a Floating Point Exception Error then user should manipulate the value of the parameters DFR_ZERO_FILL_EXTRACT and DFR_ZERO_FILL_FILTER such that their value is greater than zero but less than one (0>DFR_ZERO_FILL_EXTRACT <1 and 0>DFR_ZERO_FILL_FILTER<1).

The value of these parameters must be changed until the system stops giving Floating Point Exception Error. The user can try with different values such as 0.00001, 0.0001, or 0.001. Additionally, the value provided to the parameter DFR_ZERO_FILL_EXTRACT must be significantly smaller than the value provided to DFR_ZERO_FILL_FILTER. For example, if DFR_ZERO_FILL_EXTRACT = 0.0001 then DFR_ZERO_FILL_FILTER = 0.1.

Note: If the system does not give a Floating Point Exception Error then it is recommended that the values of parameters DFR_ZERO_FILL_EXTRACT and DFR_ZERO_FILL_FILTER is kept default (zero).

7.1.6.2 Secondary Effect Parameters

Causal data such as promotions, price, and other similar factors can affect the forecasts. These are secondary variables and their impact can be attributed in the forecasting process.

Note: These variables are applicable only for job level and 'sales' KPI.

Table 13. Secondary Effect Parameters

Analytic Key	Parameter Name	Description
dfr	DFR_DATA_EFFECT_LIST	List with one or more comma-separated values: Promo, Halo, Pantry, Price, Weather, and Delay.
dfr	DFR_EXT_PROMO_FILE	Name of the external CSV promotion file.
dfr	DFR_DATA_PROMO_TYPES	List of promotion types that need to be taken care of while forecasting. The values that can go in this parameter are the values of 'promo_type_cd' field of the external CSV file for promotions.
dfr	DFR_EXT_EVENT_FILE	Name of the external CSV events file.
dfr	DFR_EXT_PRICE_ABT	Name of the external CSV price effect file.
dfr	DFR_EXT_WEATHER_ABT	Name of the external CSV weather effect file.

Note: The default value for all the parameters listed in the previous table is blank.

Table 14. Details of the Secondary Effect Parameters

Effect	Effect on the Forecast	Input Data Required and Load Process
Promotions	This effect indicates the reaction of demand to different promotions. It reveals whether sale has increased or reduced because of these promotions. The rise in sales in the periods of promotions is called uplift and can be measured. With future data of promotions being available, all forecasts would consider the effect of historical promotions.	Provide the promotion data externally in a CSV file with all the required fields. SAS Demand Forecasting for Retail automatically reads the file and converts this file into a SAS data set that is to be used for promotion.
Price	Effect on demand can be measured because of changes in price. This is based on price elasticity of demand. Price data will be attached with sales data to figure out if there is any impact of price on the forecasts. If future data for price is available (future data means those time periods for which forecasting is being done) then the forecasts will have the effect of price.	Provide the price data externally in a CSV file with all the required fields. SAS Demand Forecasting for Retail automatically reads the file and converts the file into a SAS data set that is to be used for price effects.
Weather	Weather conditions also affect demand. Currently low and high temperatures are being considered so that in case of adverse weather forecasts, demand can be adjusted accordingly to consider its effect. Here, the focus is not on	Provide the weather data externally in a CSV file with all the required fields. SAS Demand Forecasting for Retail automatically reads the file and converts the file into a SAS data set that is to be used for weather effect.

Effect	Effect on the Forecast	Input Data Required and Load Process
	forecasting weather, instead future data of weather is used to adjust the demand that is being forecast.	
Halo/ Cannibalization	This effect accounts for cross-product or related product effects of promotions on a product.	Part of promotional effects. No extra table needs to be populated.
Delay	Customer Delayed Purchasing is also related to promotions. This effect can tell if there was a dip in sales before promotions.	Part of promotional effects. No extra table needs to be populated.
Pantry loading	This is related to and runs automatically with promotions. This effect would tell us whether there is a dip in sales in the periods after promotions.	Part of promotional effects. No extra table needs to be populated.

See section *dfr_data_gen.sas* for details of these effects.

SAS Merchandise Planning solution feeds sales data into DFR data mart. Data can be extracted from Merchandise Planning solutions for any of the levels of merchandise (M), location (L), and time (T) hierarchies as it is already in a time-series format aggregated at a week, month, or quarter level. The data is extracted into a data set that is used as an input for ABT preparation. Along with the history data set, other parameters that mention the workflow execution parameters, SAS High-Performance Forecasting execution parameters for M, L, and T hierarchy levels, and IDs are loaded in the Jobparam data set. This data set is passed on to the SAS Demand Forecasting for Retail solution.

The extracted data set has the following columns:

- Merchandise Hierarchy Columns: Mid_1 to Mid_n, where 'n' is an integer value between 1 and 10.
- Location Hierarchy Columns: Lid_1 to Lid_n, where 'n' is an integer value between 1 and 4.
- Time Hierarchy Columns, time_5, time_4 (as applicable).
- KPI Columns: Sales, units ($) and others as applicable.

The following table explains the values to be provided to Jobparam parameters, if effects of promotion, weather, events, and price are to be considered in the forecast.

Note: Extract process must be run before the user adds or removes the independent variable's effect.

Table 15. Implementing the Independent Variables

Effect to Consider	Jobparam Parameter	Change to be Implemented	Example	Remarks
Events	DFR_EXT_EVENT_FILE	Provide name of the external event file	ext_event_file.csv	See table 16 to add Events effect.
Promotions	• DFR_EXT_PROMO_FILE	• DFR_EXT_PROMO_FILE: Name	ext_promo_file.csv	See table 17 to add Promotion

Effect to Consider	Jobparam Parameter	Change to be Implemented	Example	Remarks
	• DFR_DATA_EFFECT_LIST • DFR_DATA_PROMO_TYPES	of the external promotion file • DFR_DATA_EFFECT_LIST : Writes "PROMO" in Effect list • DFR_DATA_PROMO_TYPES: Writes the types of promotions to consider	PROMO FRE WWW	effect.
Price	• DFR_EXT_PRICE_ABT • DFR_DATA_EFFECT_LIST	• DFR_EXT_PRICE_ABT: Name of the external price effect file • DFR_DATA_EFFECT_LIST: Writes 'PRICE' in Effect list	ext_price_abt PRICE	See table 21 to add price effect.
Weather	• DFR_EXT_WEATHER_ABT • DFR_DATA_EFFECT_LIST	• DFR_EXT_WEATHER_ABT: Name of the external weather ABT • DFR_DATA_EFFECT_LIST: Writes 'WEATHER' in Effect list	ext_weather_abt.csv WEATHER	See table 22 to add weather effect.
Halo cannibalization	• DFR_EXT_PROMO_FILE • DFR_DATA_EFFECT_LIST	• DFR_EXT_PROMO_FILE: Name of the external promotion file • DFR_DATA_EFFECT_LIST: Writes	Note required	Is taken care by promotion effects

Effect to Consider	Jobparam Parameter	Change to be Implemented	Example	Remarks
		'PROMO HALO' in Effect list		
Delay	• DFR_EXT_PROMO_FILE • DFR_DATA_EFFECT_LIST	• DFR_EXT_PROMO_FILE: Name of the external promotion file • DFR_DATA_EFFECT_LIST: Writes 'PROMO DELAY' in Effect list	Not required	Is taken care by promotion effects
Pantry	• DFR_EXT_PROMO_FILE • DFR_DATA_EFFECT_LIST	• DFR_EXT_PROMO_FILE: Name of the external promotion file • DFR_DATA_EFFECT_LIST: Writes 'PROMO PANTRY' in Effect list	Not required	Is taken care by promotion effects

While providing external comma separated files, refer to the following tables for description of each column. List the columns in the order specified here.

Note: The user-specified values for any field or column that ends with '_Code' (for example, Event_Code and Promo_Type_Code) must not contain spaces or special characters. Also, the external independent variable names must not contain the keywords such as SALES, UNITS, F_SALES, F_UNITS and so on.

Table 16. External Event Data File

Column Name	Data Type	Description
Lid_1 To Lid_4	Number	Specifies all location levels or lowest location levels for the worksheet being forecasted.
KPI	Character	The KPI corresponding to each event that is to be forecasted.
Event_Start_Date	Date Format: yymmdd8.	Start date for the event.
Event_End_Date	Date	End date for the event.

Column Name	Data Type	Description
	Format: yymmdd8.	
Event_Code	Character	Code for each event that is specified by the user.
Event_Desc	Character	Description of the events.
Event_Keyname	Character	Key name for the events.
Event_Nature_Code	Character	H: If the event is general, put H for all the location levels. R: If the event specified is repeated with some duration, then put R. N: If the event specified is for a particular location level and has no periodic occurrence, then put N.
Event_Type_Code	Character	Event type code.
Duration_After	Number	Number of durations after the timing value.
Duration_Before	Number	Number of durations before the timing value.

To populate external event data file:

- Events must be provided for the lowest level of location only.
- Events that are applicable for all locations should have '0' at all location levels.
- If the event is applicable for all KPIs then put 'ALL' in the KPI field.
- **EVENT_KEYNAME** must match with key names covered by SAS High-Performance Forecasting 2.3.
- **EVENT_NATURE_CODE** is a compulsory field. Possible values for this field are H, R, and N where,
  - H – Holiday
  - R – Recursive Events
  - N – Non-recursive Events
- If location-specific events are to be defined, then either specify all the location level IDs along with the corresponding event or specify only the lowest location level IDs along with the event.

Table 17. External Promotion Data File

Column Name	Data Type	Description
Merch_Lvl	Number	The lowest level of merchandise that is under promotion.
Item_Category_Rk	Number	The merchandise identification no. for the product under promotion.
Loc_Lvl	Number	The location level corresponding to merch_rk.
Location_Rk	Number	The location number corresponding to merch_rk.

Column Name	Data Type	Description
Effective_From_Dt	Date yymmdd8.	The date that marks as the starting point for the promotion.
Effective_To_Dt	Date Format: yymmdd8.	The date that marks as the ending point for the promotion.
Ret_Month_End_Dt	Date Format: yymmdd8.	Retail month end date.
Ret_Month_Start_Dt	Date Format: yymmdd8.	Retail month start date.
Ret_Quarter_End_Dt	Date Format: yymmdd8.	Retail quarter end date.
Ret_Quarter_Start_Dt	Date Format: yymmdd8.	Retail quarter start date.
Ret_Week_End_Dt	Date Format: yymmdd8.	Retail week end date.
Ret_Week_Start_Dt	Date Format: yymmdd8.	Retail week start date.
Promo_Type_Code	Character	The code used for a particular type of promotion.

To populate the external promotion data file:

- **`merch_rk`** field must hold merchandise ID from SAS Merchandise Planning.
- **`location_rk`** field must hold the location ID from SAS Merchandise Planning.
- Date variables must be provided in yymmdd8. format only.
- If PROMO_TIME_MODE = CAL, then all the **`Ret_*`** fields in the external file must be blank.
- If PROMO_TIME_MODE = RET, then depending upon the time interval of data, user must put dates in the appropriate field of the external file.
- If PROMO_TIME_MODE = RET and data is monthly, then user must put appropriate date values in fields **`Ret_Month_Start_Dt`** and **`Ret_Month_End_Dt`**. All other **`Ret_*`** fields must be kept blank.

Table 18. External Price Data File

Column Name	Data Type	Description
Item_Rk	Number	Surrogate key for the unique item identifier.
Retail_Outlet_Rk	Number	Specify location ID corresponding to the merchandise ID.
Time_Period_Rk	Date Format: yymmdd8.	Surrogate key for the unique time period value.
Actual_Price	Number	Actual price of product including price on promotion for both the historical as well as forecast horizon.
List_Price	Number	Price of item when not on promotion for both

Column Name	Data Type	Description
		historical as well as forecast horizon.
Weights	Number	Provide weight associated with the corresponding observation. Weight is calculated by dividing sales of a day by sales for the entire day.
Loc_Lvl	Number	Lowest Location Level.
Merch_Lvl	Number	Lowest Merchandise Level.

To populate the external price data file:

- **item_rk** field must hold the merchandise ID from SAS Merchandise Planning System for the SKU.
- **retail_outlet_rk** field must hold the location ID from SAS Merchandise Planning System.
- **weights** must have a numerical value ranging from 0 to 1. Any other number ignores the corresponding observation while processing data.

Table 19. External Weather Data File

Column Name	Data Type	Description
Retail_Outlet_Rk	Number	Surrogate key for the unique retail outlet identifier.
Time_Period_Rk	Date Format: yymmdd8.	Surrogate key for the unique time period value.
Low_Temperature	Number	Low temperature measured in corresponding location.
High_Temperature	Number	High temperature measured in corresponding location.
Loc_Lvl	Number	Location level.

To populate the external weather data file:

- **retail_outlet_rk** field must hold the location ID corresponding to the worksheet used.
- In the current version of SAS Demand Forecasting for Retail, provide only one level of location. **loc_lvl** field should have only one location level. More than one distinct value in this field results in error.
- **time_period_rk** must be provided in yymmdd8. format only.

When integrated with SAS Merchandise Planning, if the location or merchandise levels are skipped within the merchandise or location hierarchy then, the secondary effects must be given at the lowest level of the extracted data.

For example, if levels 1, 3, and 4 (level 2 is skipped) are used then, the secondary effect IDs must be given at level 3. In this case, level 1 of input data remains at level 1, level 3 becomes level 2, and level 4 becomes level 3 (for forecasting only).

7.1.7 Custom Secondary Effects

Custom secondary effects can be added to the extract file by providing additional data sets (or CSV format files) and specifying metadata in DFR_IND_VAR parameter. Each file added for this data must follow the format described below. The files should be placed in the **`/data/input data`** directory.

Filename: User-defined.

Purpose: Provides secondary casual data for the forecast job. Data is integrated during extract process. If the extract process is bypassed then this data must be merged into the f_xxxx input data set.

Format: Delimited flat file or SAS data set.

Jobparam: Parameters such as DFR_IND_VAR1 and DFR_IND_VAR2 hold casual variable definitions. The numbers 1, 2, and so on, in these variables correspond to the analytic template number used in the solution.

Example: Test1.csv.cause1, Test2.sas7bdat.sales_cost, maxdata.mfinc.net_sales_cost

In these examples, three fields are defined as secondary effects fields. Each field has three parameters separated by a period "**.**".

- Parameter 1: Signifies the data filename. Test1 and Test2 are filenames in the two given examples. If maxdata.mfinc syntax is used, then the field is to be extracted from SAS Merchandise Planning data mart and no corresponding external file is expected.
- Parameter 2: Signifies the data file extension format. It must either be a SAS data set (Sas7bdat) or a comma-delimited file, CSV. First example above uses CSV and the second example use a SAS data set.
- Parameter 3: Signifies the field name. In the given examples, **`cause1`** and **`sales_cost`** are the field names.

Note: The external files provided for the custom secondary effects must have unique names irrespective of their file formats (.CSV or .sas7bdat). Also, the custom secondary effect variable names provided by the user must not contain the keywords such as SALES, UNITS, F_SALES, F_UNITS and so on.

Column Name	Data Type	Comments
Merch_id	Numeric 11	Merchandise parent level ID. N=sequential list of parent levels merch_from_level to merch_to_level. Example: mid_5, mid_6, mid_7
Location_id	Numeric 11	Location parent level ID. N=level ID of parent and must contain columns for loc_from_level to loc_to_level. Example: lid_3, lid_4
Time_id	Date	Period start date. Format: ddMMMyyyy
[KPI_n]	Numeric 20.5	1…n KPIs as given by parameter. One column per KPI. Example: Sales and Units will be two columns.

Note: The input CSV file must contain the columns (listed in the above table) in the following order: Merch_id, Location_id, Time_id, and KPI_n.

When integrated with SAS Merchandise Planning, if the location or merchandise levels are skipped within the merchandise or location hierarchy then, the secondary effects must be given at the lowest level of the extracted data.

For example, if levels 1, 3, and 4 (level 2 is skipped) are used then, the secondary effect IDs must be given at level 3. In this case, level 1 of input data remains at level 1, level 3 becomes level 2, and level 4 becomes level 3 (for forecasting only).

7.1.8 Forecast Parameters

The Jobparam_[projid] data set contains many parameters that can be set for each forecast project. The diagnose parameters control the HPFDIAGNOSE process and engine parameters control the HPFENGINE process.

These parameters can be overridden at location level or/and merchandise level and by individual forecasted KPIs or the group key.

The following parameters are used for forecasting process.

Table 20. Forecast Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_HORIZONSTART	Working SAS Date of last history period. Set by system.	Not Applicable
dfr	DFR_INTERVAL	Working week, month, and quarter. Set by system.	Week
dfr	DFR_SEASONALITY	Number of periods.	52
dfr	DFR_TIME_FIELD	Field name for time of input data. Set by system.	time_5
dfr	DFR_WEEKDAY	The day in a week to which observations in history are aligned to.	2
dfr	DFR_INTERVAL_NO	The difference in days between two successive data points in history data.	7
dfr	DFR_HPF_WAIT_SEC	The wait time in seconds for checking the status of parallel threads running the forecast. Increasing the value of wait period might affect the performance if threads execution is faster than the wait period. Decreasing the value of wait period on a large volume run might generate heavy logs.	5

7.1.9 SAS High-Performance Forecasting 2.3 Diagnose Parameters

The following parameters can be overridden at merchandise level, location level, and KPI. To override these parameters, provide the job (or worksheet) ID, level, KPI name, or ID of the merchandise, time, and location key.

Table 21. SAS High-Performance Forecasting 2.3 Diagnose Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	DIAG_ALPHA	Conf level size in conf limits.	.05
dfr	DIAG_ARIMAX	Options of ARIMAX statement.	criterion=sbc estmethod=cls method=minic siglevel=0.05 outlier=(detect=maybe filter=subset) identify=arima
dfr	DIAG_BACK	Number of observations before the end of the data.	0
dfr	DIAG_BASENAME	Prefixes the model specification filename.	DIAG
dfr	DIAG_CRITERION	Criterion for model diagnosis.	RMSE
dfr	DIAG_CUSTOM_MODEL_FILE	.sas file path and name.	
dfr	DIAG_DELAYEVENT	Delay lag for the events.	0
dfr	DIAG_DELAYINPUT	Delay lag for input variables.	<user-specified>
dfr	DIAG_ENTRYPCT	Threshold for comparison of different models.	0.1
dfr	DIAG_ERROR_CONTROL	Allows finer control of message printing.	(severity=all stage=all)
dfr	DIAG_ESM	Options of ESM statement.	method=best
dfr	DIAG_EVENT_REQUIRED	Controls inclusion of events.	YES
dfr	DIAG_FORECAST_OPTION	Options of FORECAST statement.	accumulate=NONE TRIMMISS=both zeromiss=NONE
dfr	DIAG_HOLDOUTPCT	% of the holdout sample.	0
dfr	DIAG_ID_OPTION	Options of ID statement.	accumulate=NONE trimmiss=both zeromiss=NONE align=ending
dfr	DIAG_IDM	Options of IDM statement.	intermittent=2 base=auto
dfr	DIAG_INCLUDE_ARIMAX	Indicator variable to control ARIMAX model fit.	1
dfr	DIAG_INCLUDE_ESM	Indicator variable to	1

Analytic Key	Parameter Name	Description	Default Value
		control ESM model fit.	
dfr	DIAG_INCLUDE_IDM	Indicator variable to control IDM model fit.	0
dfr	DIAG_INCLUDE_UCM	Indicator variable to control UCM model fit.	0
dfr	DIAG_INPUT_LAG_USER	Binary variable to control the DELAYINPUT option.	0
dfr	DIAG_INSELECTNAME	Name of catalog for model selection list.	bests
dfr	DIAG_MINOBS	Number on which SEASONAL or TREND model fitting depends.	(SEASON=2 TREND=1)
dfr	DIAG_PREFILTER	Way of handling missing and extreme values.	YES
dfr	DIAG_PRINT	Specifies the print option.	NONE
dfr	DIAG_SELECTBASE	Prefixes model selection list filename	DIAG
dfr	DIAG_SELECTEVENT	Number of events to select.	SELECT
dfr	DIAG_SELECTINPUT	Number of input variables to select.	SELECT
dfr	DIAG_SIGLEVEL	Significance level.	0.05
dfr	DIAG_SPECBASE	Prefixes model specification filename.	DIAG
dfr	DIAG_TESTINPUT	Type of testing for input variables.	BOTH
dfr	DIAG_TRANSFORM	Options of TRANSFORM statement.	type=auto transopt=mean siglevel=0.05
dfr	DIAG_TREND	Options of TREND statement.	diff=auto sdiff=auto siglevel=0.05
dfr	DIAG_UCM	Options of UCM statement.	component=ALL siglevel=0.05 refineparms=(factor=all)

7.1.10 SAS High-Performance Forecasting 2.3 Engine Parameters

The following table lists the parameters of SAS High-Performance Forecasting 2.3 Engine.

Table 22. SAS High-Performance Forecasting 2.3 Engine Parameters

Analytic Key	Parameter Name	Description	Default Value

Analytic Key	Parameter Name	Description	Default Value
dfr	ENG_ADJUST_VARLIST	List of adjustment variables.	-
dfr	ENG_BACK	Specifies the number of observations before the end of the data that the multistep forecasts are to begin.	0
dfr	ENG_ERROR_CONTROL	Enables finer control of message printing.	(severity=all stage=all)
dfr	ENG_FORECAST_OPTION	Options for forecast statement.	accumulate=NONE trimmiss=both zeromiss=NONE
dfr	ENG_GLOBALSELECTION	Name of catalog for model selection list.	bests
dfr	ENG_ID_OPTION	Options of ID statement.	accumulate=NONE trimmiss=both zeromiss=NONE align=ending
dfr	ENG_LEAD	Number of periods ahead to forecast.	40
dfr	ENG_OUTMODELINFO	Detailed information about selected forecast model.	outmodelinfo
dfr	ENG_OUTSTATSELECT	Contains statistics of fit for all candidate models.	outstatselect
dfr	ENG_PLOT	Graphical output desired, default no output.	
dfr	ENG_POST_ADJUST	Post adjustment operation.	NONE
dfr	ENG_PRE_ADJUST	Pre-adjustment operation.	NONE
dfr	ENG_PRINT	Printed output desired, default no printing.	

7.1.10.1 Revisit Parameters

These parameters are job level and *cannot* be overridden at merchandise level, location level, and KPI.

Table 23. Revisit Parameters

Analytic Key	Parameter Name	Description	Default Value
dfr	REVISIT_INNER	Inner valid boundary.	0.85
dfr	REVISIT_OUTER	Outer valid boundary.	0.95

dfr	REVISIT_OBS	Number of observations to be considered for a revisit.	7

7.1.11 SAS High-Performance Forecasting 2.3 Reconcile Parameters

These parameters are job level and *cannot* be overridden at merchandise level, location level, and KPI.

Table 24. SAS High-Performance Forecasting 2.3 Reconcile Parameters

Analytic Key	Parameter Name	Source Field	Description	Default Value
dfr	RECON_SINGLE	fcast_reconcile	1= Single level only 0 = All levels	0
dfr	RECON_HIER	fcast_recon_hier	0=No recon 1=Merch 2=Loc	0
dfr	RECON_LOC	fcast_recon_loc	0=None Not equal to '0', then location level. If TD, then 'From' level. If BU then, 'To' level of worksheet.	4
dfr	RECON_MERCH	fcast_recon_merch	0=None Not equal to '0', then merchandise level. If TD, then 'From' level. If BU then, 'To' level of worksheet.	10
dfr	RECON_STDDIFBD	None	Boundary for % difference between reconciliation standard error and original standard error.	0.25
dfr	RECON_CLMETHOD	None	Method to compute conf limit for reconciliation forecast.	SHIFT
dfr	RECON_DISAGGREGATION	None	Loss function for top-down reconciliation.	DIFFERENCE
dfr	RECON_ERRTRACE	None	Resolution at which warnings are printed.	DATASET
dfr	RECON_SIGLEVEL	None	Significance level.	0.05
dfr	RECON_STDMETHOD	None	Method to find standard error of	UNCHANGED

Analytic Key	Parameter Name	Source Field	Description	Default Value
			recon forecasts.	
dfr	RECON_DIFF_REQUIRED	None	Controls the RECDIFF option.	1
dfr	RECON_WEIGHT_REQUIRED	None	Controls the WEIGHTED option.	0

7.1.12 Generic and Job Scope Parameters

These parameters are job level parameters and cannot be overridden at merchandise level or location level and KPI.

The location, merchandise, and time parameters listed in the following table define scope of the forecast to be generated. These parameters are copied from SAS Merchandise Planning maxdata.planworksheet table.

Table 25. Generic and Job Scope Parameters

Analytic Key	Parameter Name	Source Field	Description
Ma	MMAX_VERSION	-	SAS Merchandise Planning version (5 or 6). The default value is 6.
Ma	JOBNAME	Name	Name of the forecast job.
Ma	REQUEST_ID	Analytic_lkup_id	Job analytic type. Forecast requests should be 1-9 with at least type 1 required. See Batch_Analytic_lkup table.
Ma	PATH_INP	-	Allows setting of different path for input data. Blank value uses default of **`mx_analytics/data/input`**.
Ma	PATH_OUT	-	Allows setting of different path for input data. Blank value uses default of **`mx_analytics/data/output`**.
Ma	PATH_REPORT	-	Allows setting of different path for input data. Blank value uses default of **`mx_analytics/data/reports`**.
Ma	PATH_STAGE	-	Allows setting of different path for input data. Blank value uses default of **`mx_analytics/data/staged`**.
Ma	PATH_ARCHIVE	-	Allows setting of different path for input data. Blank value uses default of **`mx_analytics/data/archive`**.

Analytic Key	Parameter Name	Source Field	Description
Ma	LOC_FROM_ID	from_loc_id	Topmost location ID. Provided by Merchandise Planning.
Ma	LOC_FROM_LEVEL	from_loc_level	Top location level to forecast.
Ma	LOC_TEMPLATE	loc_template_id	Location member definition provided by Merchandise Planning.
Ma	LOC_TO_LEVEL	to_loc_level	Lowest location level to forecast.
Ma	MERCH_FROM_ID	from_merch_id	Topmost merchandise ID provided by Merchandise Planning.
Ma	MERCH_FROM_LEVEL	from_merch_level-10	Topmost merchandise level to forecast.
Ma	MERCH_TEMPLATE	merch_template_id	Merchandise member definition provided by Merchandise Planning.
Ma	MERCH_TO_LEVEL	to_merch_level-10	Lowest merchandise level to forecast.
Ma	CALENDAR	time_path_id	Calendar ID provided by Merchandise Planning.
Ma	FROM_TIME_ID	from_time_id	Topmost time member or date first time member to forecast.
Ma	TIME_FROM_LEVEL	from_time_level-46	Topmost time level to forecast.
Ma	TIME_TEMPLATE	time_template_id	Time member definition provided by Merchandise Planning.
Ma	TIME_TO_LEVEL	to_time_level-46	Lowest time level to forecast. Must be 3 (quarter), 4 (month), or 5(week).
Ma	TO_TIME_ID	-	Last time member to forecast. Date when used.
Ma	MA_MAX_RSUBMIT	-	The maximum threads that execute simultaneously.
Dfr	ORIG_MERCH_TO_LEVEL	-	Working merchandise to level at the beginning of process.
Dfr	DFR_IVAR_CHNG	-	1: If independent variables are changed from the previous run. 0: If independent variables are not changes from the previous run. This is the default value.

Analytic Key	Parameter Name	Source Field	Description
Dfr	ORIG_LOC_TO_LEVEL	-	Working location to level at beginning of process.

7.1.13 Working Parameters

Working parameters are internal system parameters and are temporary. These parameters are at job level and cannot be overridden at merchandise level or location level and KPI.

Table 26. Working Parameters

Analytic Key	Parameter Name	Source Field	Description	Default Value
Ma	JOB_RC	analytic_err_id	Return code.	0
Ma	JOB_RC_MSG	bat_error_msg	Return message.	-
Dfr	HIST_FROM_TIME	-	Starting date of history.	-
Dfr	FROM_TIME	-	Working forecast start date.	-
Dfr	HIST_TO_TIME	-	Working history end date.	-
Dfr	TO_TIME	-	Working forecast end date.	-
Dfr	DFR_KPINAMESET	-	Working variable.	-
Dfr	DFR_NUMKPIS	-	Working variable.	-
Dfr	DFR_IND_VAR	-	Working variable. Holds independent variables to load.	-

7.2 Exception Management Using SAS Forecast Studio

Execution of a forecast request produces an exception data set. This data set can be loaded in SAS Forecast Studio and used for analysis and reforecast.

The exception file is produced with exception criteria. The parameters in this file are set by using the parameter DFR_EXCEPTION_FILTER that is defaulted to MAPE > 50. This parameter can be set to any SQL syntax filter using the fields within the SAS High-Performance Forecasting Outstat data set.

The following workflow shows creation and export of a SAS Forecast Studio project after SAS Forecast Studio is installed.

7.2.1 Creating a New Forecast Studio Project

To create a new SAS Forecast Studio project:

1. If **SAS Forecast Studio** is being used for the first time, create a library in **SAS Management Console** with the advanced option **Pre-assigned** set to **On**.

2. Launch **SAS Forecast Studio** (after starting the AP Server) and logon with appropriate credentials.

3. From the **File** menu, select **New Project**. The **New Project Wizard** appears.

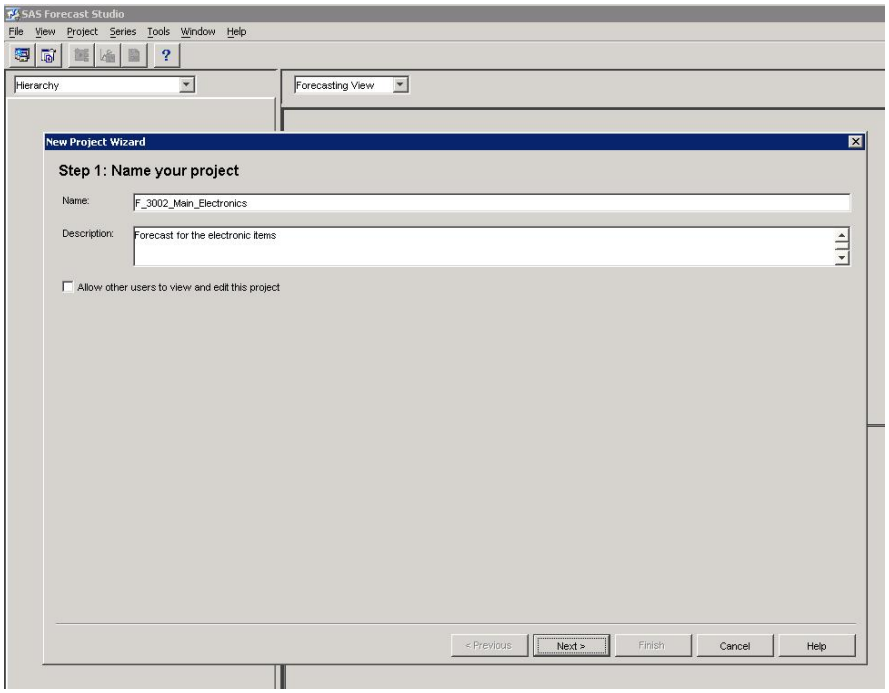
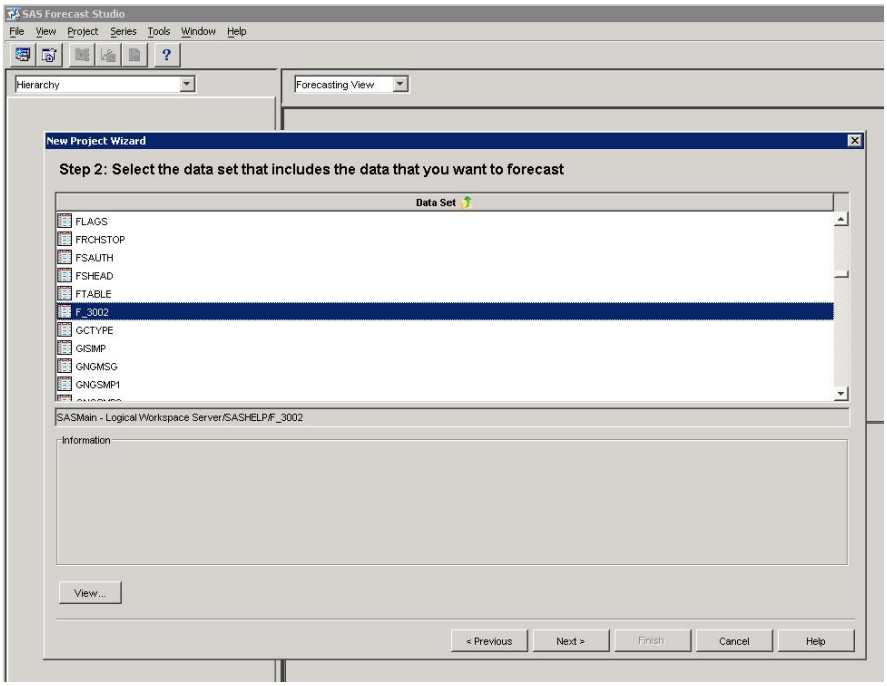


Figure 18. New Project Wizard

4. In the **New Project Wizard**, provide name and description of the new project and click **Next**.

5. Select the data set that is to be forecasted and click **Next**. The data set name follows FE_[projid] convention.



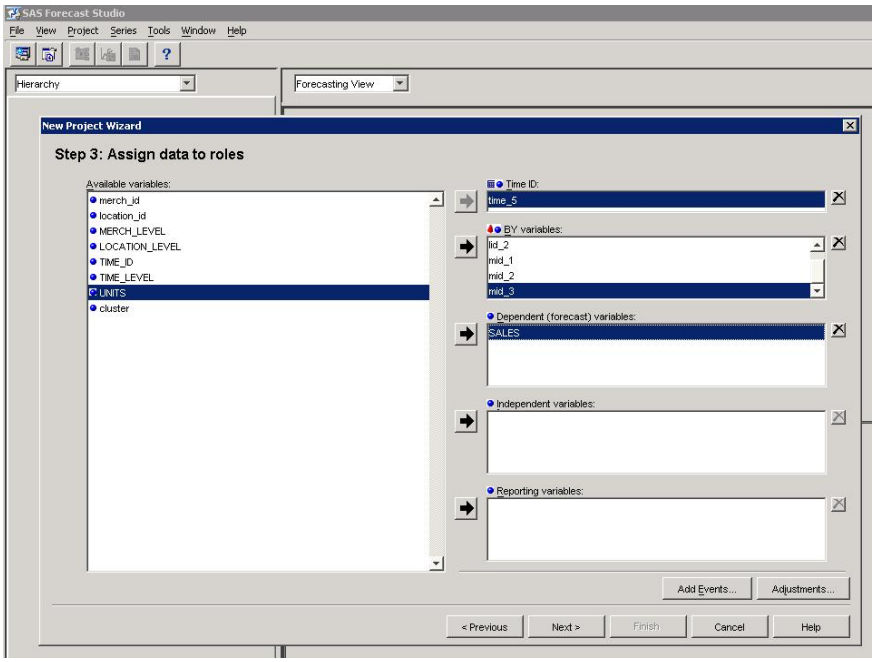
6. Set the variables as explained here:

   a. Select time_[x] in the Time ID field.

   d. Select all the location IDs into the BY Variables. The order of location names in BY Variables must be in ascending order. For example, if the data set has four location levels, then entries in the BY Variables can be Lid_1, LID_2, and so on.

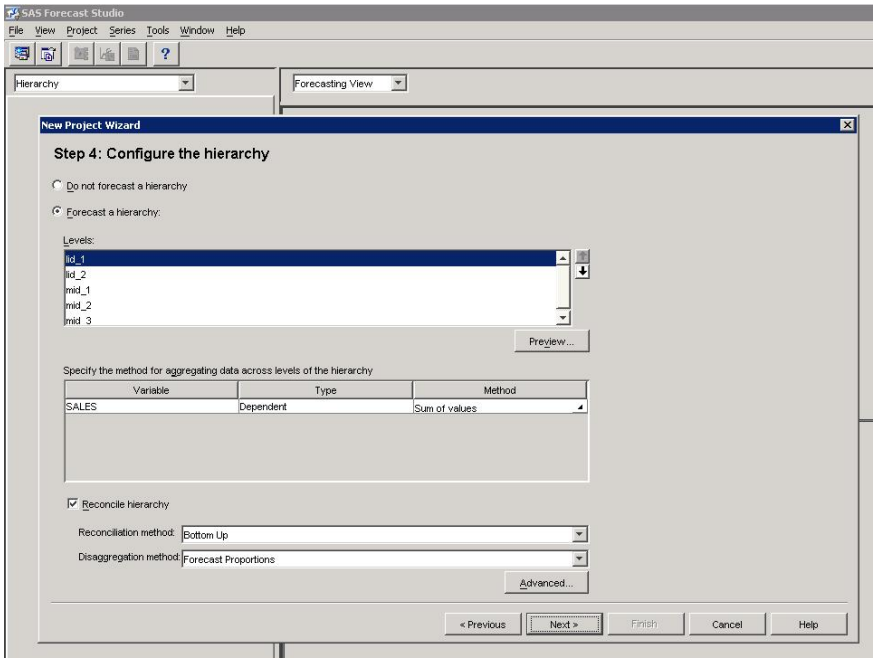
   e. Select all the merchandise levels into the BY Variables after the last location ID. The order of merchandise names in the BY Variables should be in ascending order. For example, if a data set has seven merchandise levels, then BY Variables should have Mid_2, Mid_3, Mid_4, and so on.

   f. Select SALES or UNITS (or another forecast variable) as the Dependent Variable. The window must appear as displayed here.

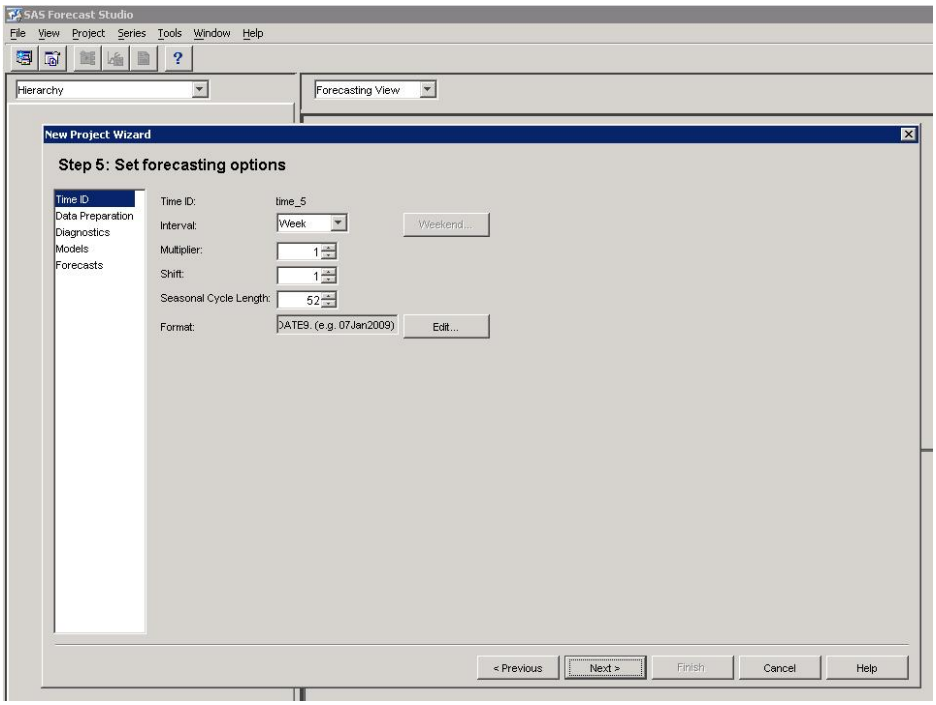


7. Click **Next**.

8. Select hierarchy levels for variables and the method for aggregating data across hierarchy levels. Also, select appropriate values in fields **Reconciliation Method** and **Disaggregation Method**. Click **Next**.



9. Set the forecasting options such as, **Interval** and **Seasonal Cycle Length**. Click **Next**.



10. Click **Finish** in the confirmation screen that displays the message "`Your Project is Now Complete`". The **SAS Forecast Studio** now runs the forecast and displays a window similar to the one displayed in the following figure.

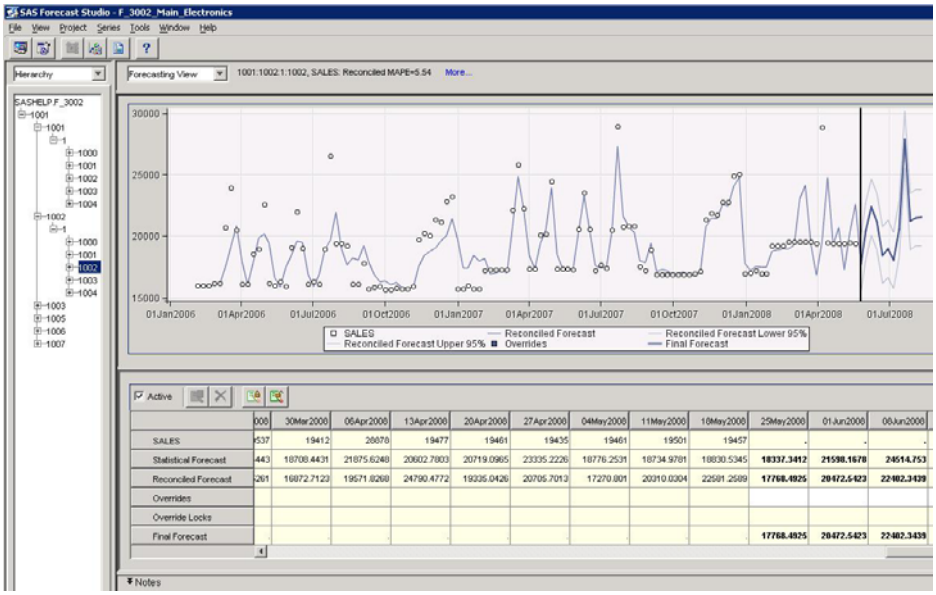


Figure 19. Forecast Studio Workbench

Forecast results are NOT intended to be re-imported into SAS Merchandise Planning.

Only a complete batch run of the entire forecast can give the forecast results and exception forecasts cannot be merged.

The parameters and model work done in SAS Forecast Studio must be entered into the SAS Demand Forecasting for Retail parameter data set and custom model files.

7.3 Reports

The solution supports all the standard reports that are generated by PROC HPFDIAGNOSE and HPFENGINE. Parameters DIAG_PRINT, ENG_PRINT, and ENG_PLOT are used for printing and plotting the results.

The generated forecast report is compiled into an HTML file and placed in **`/data/reports`** directory by using the name fr_[project name]_[projid].htm.

The forecast project worksheet provides a link to the report that is available when the worksheet is opened using the **Tools** menu.

Note: To get the reports on UNIX operating system using ENG_PLOT, the worksheet must be run under X-Windows environment.

7.4 Revisit

The revisit process evaluates the quality of forecasting model. In this process, the difference between forecasted value and actual value (obtained after forecasting process) is calculated as an error. This error is the input to PROC SHEWHART. Inner and outer limits (in percentage) have to be specified for the request in Jobparam data set. The statistical quality control procedure evaluates the effectiveness of forecasting process by the error value.

The following rules determine the worthiness of the forecasting model:

- If the error lies between inner limits, then model fitted is considered good enough and the forecast for next time period is generated.
- If the error lies within outer limits and outside of the inner limits, then model requires tuning, and therefore revisit process recommends only model fitting.

- If the error lies outside the outer control limit, then the entire model requires revisited again.

The following figure shows the inner and outer limits of the forecasting model.

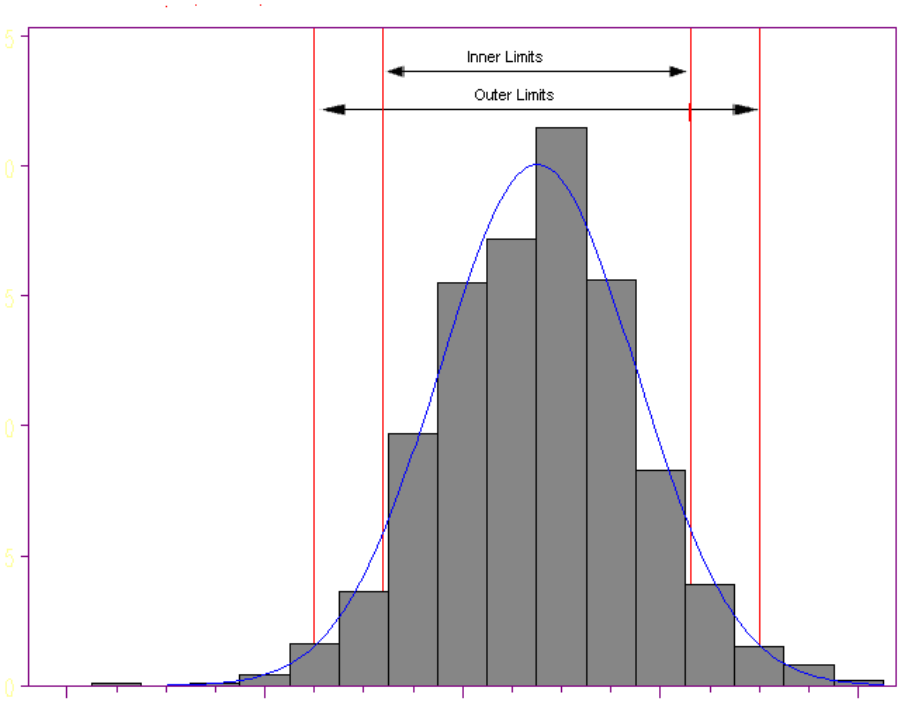


Figure 20. Model Revisit Process

If observations are insufficient (less than the value of revisit_obs):

- If revisit and the intelligent workflow are set On (dfr_wf_revisit=1 and dfr_wf_forecast_intel_exe=1), then diagnose, fit, select, and forecast processes are run irrespective of the user's choice.
- If user does not want the diagnose, fit, select, and forecast processes to run, then the intelligent workflow must be turned Off (dfr_wf_forecast_intel_exe=0).

7.5 Custom Model

A custom model file is available through parameter, DIAG_CUSTOM_MODEL_FILE. This file is available before the diagnose step and is included in the diagnose procedure when executed by SAS High-Performance Forecasting. The file must be placed in the input data directory.

There are two ways to use custom model:

1. **Suggest Custom Model**: This option allows the user to suggest a custom model along with the models suggested by PROC HPFDIAGNOSE. In this process, user-suggested models are compared with other models and best model is used for forecasting.

2. **Forced Fit Custom Model:** In this option, only the user-suggested model is used. PROC HPFDIAGNOSE does not generate any model for that time series. This can be achieved by new options of SAS High-Performance Forecasting 2.3.

The user can select the method of custom model by setting appropriate value of the parameter DFR_FORCED_CUSTOM_MODEL in the Jobparam data set. For Forced Fit Custom Model, the value of parameter DFR_FORCED_CUSTOM_MODEL is set to ‘1’, else it takes the default value ‘0’ for first model, which is Suggest Custom Model.

Note: In custom model file, the value of parameters MODELREPOSITORY and REPOSITORY must be "dia.Cat_F_&proj._&get_ds_ref_no." (MODELREPOSITORY = dia.Cat_F_&proj._&get_ds_ref_no. and REPOSITORY= dia.Cat_F_&proj._&get_ds_ref_no.).

7.6 Forecast Flow

The following workflow flags are available for each of the four steps in the process:

- Diagnose
- Select
- Fit
- Forecast

To execute the process, consider using either Manual Modification mode to modify the workflow flags, or the Intelligent Workflow mode.

Manual Modification

SAS High-Performance Forecasting workflow rules can be overridden by manually modifying the workflow flags. For example, a forecast job with Diagnose, Select, and Fit flags turned On (values set to 1). Execution of this job simply requires a reforecast because the Diagnose, Select, and Fit output data from the previous run is still available. In the second run, the user can set Diagnose, Select, and Fit to Off (values set to 0) and Forecast to On (value set to 1).

Intelligent Workflow

Intelligent Workflow enables the system to set the workflow steps that are based on user request. It ensures that all valid workflow steps are executed.

The user can turn the Intelligent Workflow On or Off by setting the value of the parameter DFR_WF_FORECAST_INTEL_EXE to '1' and '0' respectively.

The following is an example of the Intelligent Workflow.

Consider the following figure. In Run1, user sets Select and Forecast flags to '1'. In this case, no prior execution data exists and the system must execute all the steps to run a forecast. Therefore, system overrides the user input and ensures that all steps are executed. See System Run1.

In Run 2, user sets Select and Forecast flags to '1'. The system detects that the Select process is being run in Run2, so Fit must also run in Run2. Therefore, it automatically sets Fit flag On and overrides user's Fit flag. See System Run2.

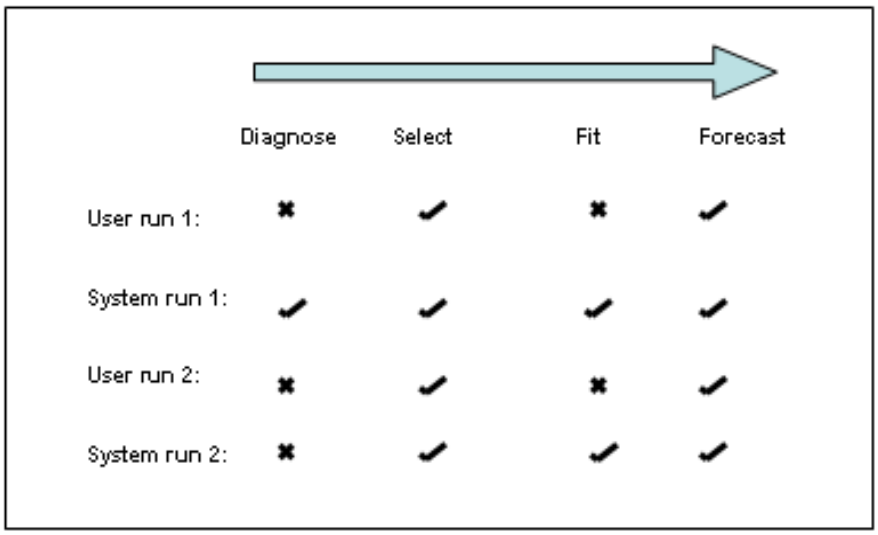


Figure 21. Example of an Intelligent Workflow

7.7 Filtering Time-Series History

Filters can be applied to generate an exception file (see *Exception Management Using SAS Forecast Studio*) or to filter out time-series based on the number of historical records that are displayed.

The filter eliminates:

- old products with no recent sales
- products with negative sales only
- new products without enough history to forecast

This elimination is done by removing all the time series that have history less than the value of parameter DFR_FILTER_PRIOR_HIST. History is counted from last loaded week and backward in time. For example, if DFR_FILTER_PRIOR_HIST=26 and DFR_FILTER_PRIOR_PERIODS=52, the system eliminates all the time series having less than 26 positive values within last 52 weeks of the history. Filter periods are always expressed in number of weeks.

In standalone mode additional parameters such as DFR_FILTER_START_DATE and DFR_FILTER_END_DATE are used. These parameters are set in ddMMMyyyy format. The start and end dates define the time period for the filter.

7.8 New Products Assignment and Forecast

To enable the new product history load, the parameter DFR_WF_NEWITEM is set to '1'. This parameter gets acts-like history and merges it with normal product history before a forecast is run.

New product forecasts apply only to the lowest level of merchandise within a forecast job request

7.8.1 Like-Item Assignment

Three ACTS LIKE parameters (ACTS LIKE1, ACTS LIKE2, and ACTS LIKE3) are available for assigning the acts-like merchandise members that are used to substitute history for new merchandise members. Filtered lookup function of SAS Merchandise Planning can be used to filter and select a member. Assignments can also be copied and then pasted within a worksheet.

This function considers the prior like-item assignments. Therefore, a chain of linked assignments can be used to formulate a complete historical record. For example, if item D has like-item C and item C has like-item B, then the system pulls the history of both, B and C for the item D. Start-date field defines the beginning date of history for new item. End-date defines the end date of the history for the like-items.

To adjust the final average history values of the assignments, use the PERCENTAGE ADJUST parameter.

7.8.1.1 Changing the Merchandise Level

The solution supports new products at merchandise level six, which is typically Style/Color. To support other levels such as SKU, the SAS technical consultants must modify the following SQL as part of the solution implementation.

The following fields are added to the appropriate level (for SKU use lv10mast table).

```
alter table maxdata.lv6ctree add (
       acts_like1 numeric(10),
       acts_like2 numeric(10),
       acts_like3 numeric(10),
```

```
        act_parent_lev numeric(3),
        act_pct_adjust numeric(10,3),
        act_start_date date,
        act_end_date date
);
```

Also, modify the maxdata.actslike_lkup view to bring in attributes customer uses to determine like Styles or SKUs.

7.8.2 Like-Item Assignments Using External File

The solution can also load the like-item assignments from a flat-file instead of taking them from Merchandise Planning worksheet interface. The like-item assignment from an external flat file can be done be setting the parameter DFR_WF_NEWITEM_EXTERNAL to '1'.

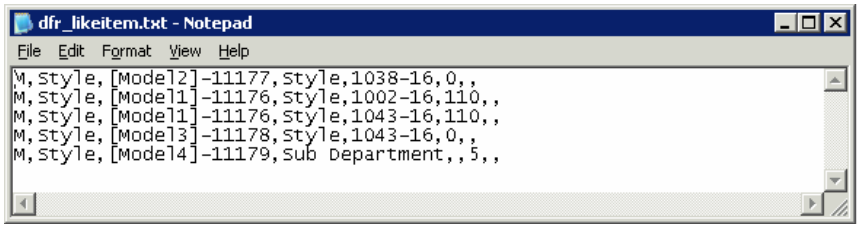
This feature uses a comma-delimited text file, dfr_likeitem.txt. This file is in the **`mx_analytics/data/input`** folder.

Table 27. Parameters for External Import

Field	Description	Example Value	Required
like_item_type	Like item assignment type.	M	Yes
new_item_level_name	The text level name of the new item. This name must be same as the level name as specified in the merchandise planning system.	Style	Yes
new_item_name	The text name of the like item. This value must be the same as the user ID field within the merchandise planning system.	[Model1]-12345	Yes
like_item_level_name	The text level name of the like item. This name must be the same as the level name as specified in the merchandise planning system.	Class	Yes
like_item_name	The text name of the like item. This value must be the same as the user ID field within the merchandise planning system.	1002-16	Yes
pct_of_parent	Percent of parent or adjustment to sibling like-item assignments. 110= 110% = 10% than existing. 20=20% = 20% of existing.	110	Optional
start_date	Start date when assignment begins.	09/12/2007	Optional
end_date	End date of assignment.	11/05/2007	Optional

For multiple like-item siblings, upto three rows for the same new item can be entered. The percentage of parent, start-date, and end-date must be same for each of these rows if these fields are used.

The following is a sample input file.



7.8.3 Parent Assignment

The solution enables using a parent value for the item instead of assigning the sibling items. Parent assignment is set using the parameter NP PARENT LEVEL that is available in the plan worksheet at merchandise member attribute level. The parent level that is selected must be above the current merchandise level where the assignment is made. If a parent level is selected, it overrides all like-item sibling assignments and the two values are not combined.

The parameter NP PCT ADJUST is set to specify the percentage of parent value assigned to the new product.

7.8.4 Start and End Dates

The start and end dates, if used, sets the boundary of the history being used for the new products. Setting a start date for a new product overrides any like-history.

7.9 Like-Item Filter

A filter can be used to filter a set of candidate like-items based on certain attributes.

To set like-items filter for any ACTS-LIKE parameter:

1. Select an item from a list of relevant items in the **FilterLookup** window.
2. To access the filter, select an **Acts-like KPI** for a new product in the worksheet and click **Filter**. In the following example, several attributes are provided at style level to enable filtering of styles so that an appropriate acts-like style can be selected.
3. The like-item filter fields can be configured through metadata. For more information to set up metadata, see *SAS Merchandise Planning Metadata_6 Guide.*

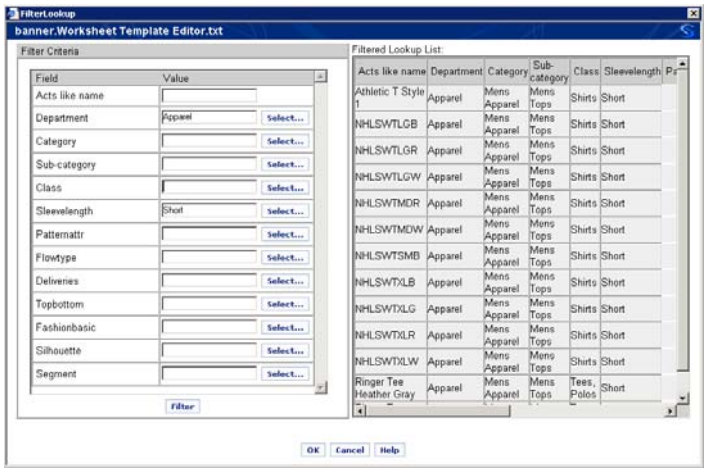
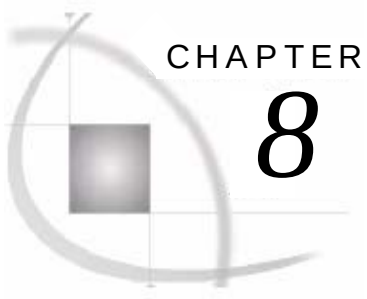


Figure 22. Like-Item Filter



CHAPTER 8

Administration of Demand Forecasting for Retail

8.1 Execution of a Forecast Request

This section provides details about the environment that needs to be set before executing the forecast requests. It also explains the procedures to handle errors, data archive, data purge, batch execution of jobs, and log maintenance for each job.

8.1.1 Performance

In SAS Demand Forecasting for Retail, the extract, forecast processing, and import processes are multitasked for enhanced performance. The user can optimize the execution time by setting number of threads and individual task data size appropriately.

The following parameters are used to tune the performance of SAS Demand Forecasting for Retail.

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_WF_OPTIMIZED	Used for production optimized extract, log options, and file size.	0
dfr	DFR_RUNNING_MODE	0=Production Mode 1=Testing Mode	0
ma	MA_MAX_RSUBMIT	Defines the maximum number of threads that can run simultaneously.	6
dfr	DFR_WF_GRID	0: Grid disabled 1: Grid is enabled	0

Analytic Key	Parameter Name	Description	Default Value
dfr	DFR_NO_TS	This attribute is used when Performance Split is 'On' and split process is based on number of time series. 0= No split or split value should be a positive integer number	0
dfr	DFR_NO_DS	This attribute is used when Performance Split is 'On' and split process is based on number of data sets. 0= No split or split value should be a positive integer number	0

8.1.2 Grid Execution

SAS Grid Manager can be used to scale-up the execution performance of SAS Demand Forecasting for Retail. The following figure depicts a typical architecture used in grid execution.

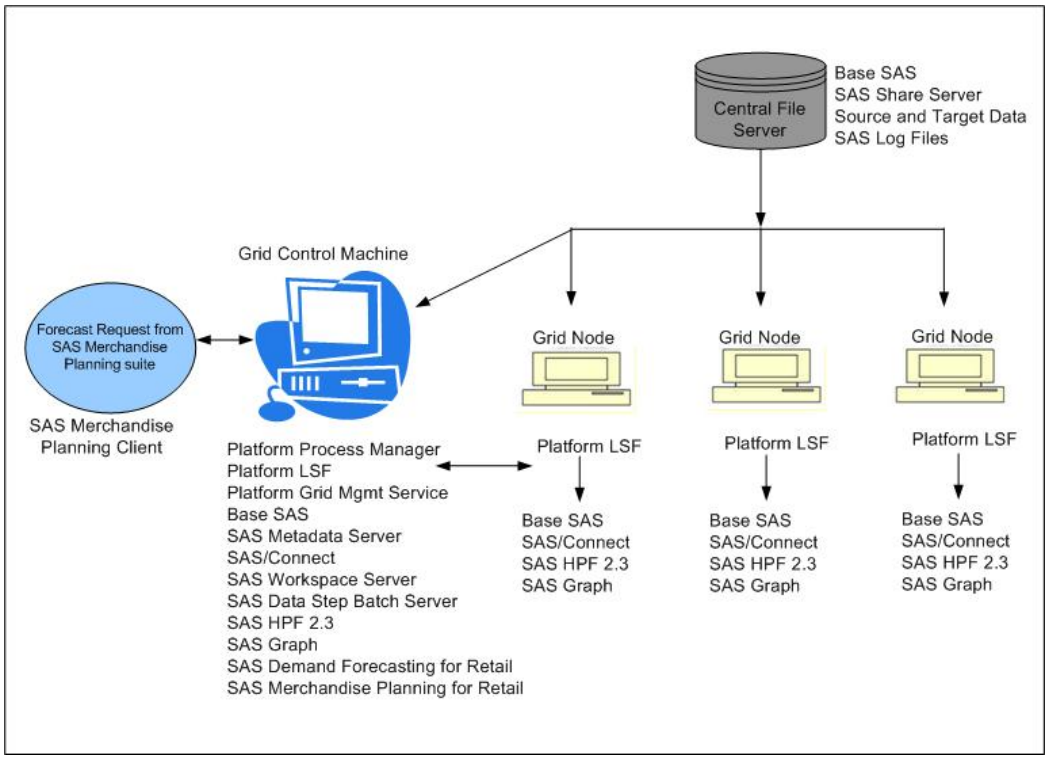


Figure 23. Grid Execution

Note: A separate SAS Grid Manager license is required to run the Grid Manager.

8.1.3 Process Server

The process server must be started before forecast requests are created in SAS Merchandise Planning.

The file process_server.sh (for UNIX) or process_server.bat (for Windows) is used to start SAS process server. This process runs in the background on the SAS Server and runs merchandise analytic requests such as Demand Forecast requests. An administrator schedules the run of process server.

A running process server window is displayed in the following figure.

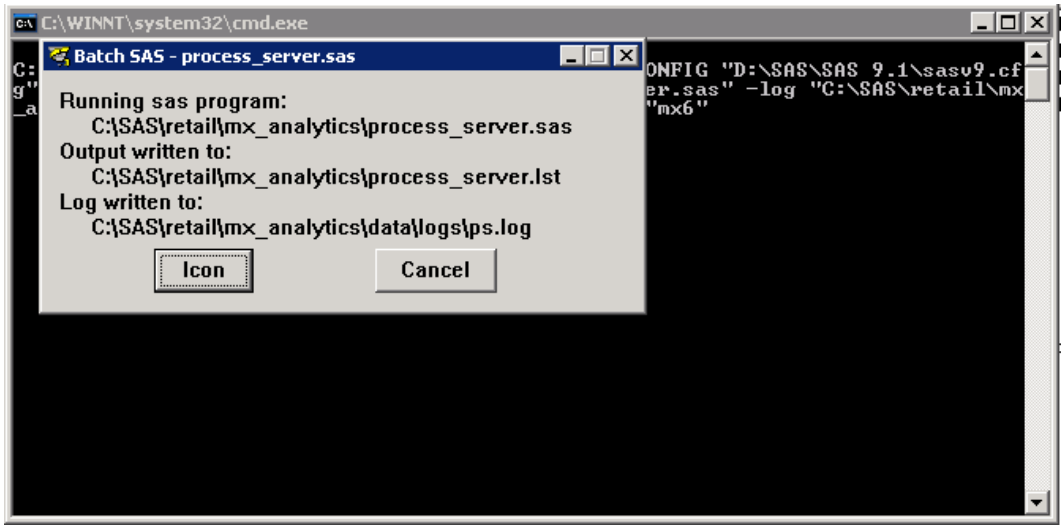


Figure 24. Process Server Execution

The process_server.bat (for Windows) or process_server.sh (for UNIX) files are located in **/mx_analytics** directory. These files must be configured and run before defining forecast jobs. The program, process_server.bat, picks up forecast requests that are issued either from the SAS Merchandise Planning user interface or from SAS Demand Forecasting for Retail batch program.

The execution string of the shortcut must be modified such that it points to the correct directories for both SAS system batch routine (sasbatch.bat) and **/analytics** directory.

The default shortcut is **/appl/SAS/SAS_9.1/sas -CONFIG /appl/SAS/SAS_9.1/sasv9.cfg -sysin ./mabatch.sas -log ./logs/mabatch.log -noterminal -sysparm "mx6"**

The directories can be validated and modified, if required.

This server places a log file, batch.log in the **/logs** directory (or at a location where it has been configured to save the log). This log is checked for errors if forecasts are not generated properly.

8.1.4 Batch Execution

When running in batch or stand-alone mode from a SAS console or through a scheduler program, use the following code. This code bypasses the **process_server**.

```
/*********************************/
/* INITIALIZE PARAMTERS */
/*********************************/
%let sysparm=mx6;
%include "c:/sas/retail/mx_analytics/params.sas";
/*options mprint mlogic symbolgen;*/
/*************************************/
/* RUN THE FOLLOWING FORECAST PROGRAMS */
/*************************************/
%mm_forecast(wktid=1004,dsn=&dsn.,dbtype=&dbtype.,paramfile=,
set_analytic=1);
```

More than one forecast can be run in sequence by adding more MM_FORECAST calls to the end of this code. The parameters for this code are given below.

Table 28. Parameters for Batch Execution

Parameter	Description
SYSPARAM	Database connection name. In Oracle this is the tnsname string.
WKTID	Forecast project ID (worksheet ID) Tip: Get the forecast job ID (worksheet ID) by looking in the lower left corner of the screen when in Merchandise Planning system.
PARAMFILE	Default=Null. Gives alternate parameter filename if a different parameter file is needed for same job.
SET_ANALYTIC	Normally is set to '1'. If more forecast types are added, then the value can be from 2 to 9.

This batch program is kicked-off using either a corresponding .sh or .bat file, which must also be created.

For example, the script file batch_sample.sh. The echo code is set to a return code for a scheduler application. This code can be removed if no scheduler is being used.

```
/sas/SAS_9.1/sas -sysin /sas/mx_analytics/batch_sample.sas -noterminal
rc=`grep "Job completed without errors" batch_sample.log |wc -l`
if (( $rc > 0 )) then
       echo "Completed without errors"
       exit 0
fi
echo "Failed"

exit 1
```

The exit code is set to return a code for a scheduler application. A return code '1' means failure and '0' signifies successful execution.

8.1.5 Error Handling

In the **Forecast Management** window, the error fields that can be viewed for the job are:

Error Message	Error Field	Type
Batch Error Msg	bat_error_msg	char(255)
Analytic Error ID	analytic_error_id	numeric(6)

If no error has occurred, the message "`Job Completed Without Error`" appears in the **Batch Error Msg** in the **Plan Management** window for the forecast project worksheet. However, if an error has occurred, appropriate message is displayed in this field.

The parameter JOB_RC is the error code that is set by calling RCSET error handling macro as shown here.

- **`%RCSet(&syserr);`**

  `syserr` is the system variable that SAS sets if there is any error in a DATA step or procedure.

- **`%RCSet(&sqlrc);`**

  `sqlrc` is the system variable that SAS sets if there is any error in PROC SQL.

- **`%RCSet(&syslibrc);`**

  `syslibrc` is the automatic system variable that SAS sets if there is any error in the LIBNAME statement.

- **`%RCSet(&sysfilrc);`**

  `sysfilrc` is the automatic system variable that SAS sets if there is any error in the filename statement**.**

Scheduler Error Return Codes

See the sample batch job, sample_batch.sh, for details on scanning an error log and returning error codes back to a scheduler.

8.1.6 Logs

The **`mx_analytics/data/logs`** directory contains the batch log when running process_server. The file is named ps.log.

For each forecast job that is executed, a log named **`ma_forecast project/worksheet_id.log`** is created in the **`logs`** directory. This log is scanned at the end of processing for the errors and error set within batch error message.

8.1.7 Archive

If DFR_WF_ARCHIVE workflow step is On (value set to '1'), the archive feature automatically takes backup of the Outstat forecast, Outsum forecast, and forecasted data sets that are available in the Out library.

The number of past runs that are kept archived can be set through the parameter DFR_ARCHIVE_COUNT.

8.1.8 Debug Versus Production

The parameter DFR_RUNNING_MODE is set to '1' to delete any previous diagnose or SAS High-Performance Forecasting estimate data. It is assumed that the following changes are not conducted in production mode:

- worksheet reconcile dimension
- changing forecasting KPI
- addition or removal of any independent variable (DFR_IVAR_CHNG=1)
- addition or removal of any new event and/or promotion (DFR_IVAR_CHNG=1)
- changing group key of any time series from the previous run
- changing number of split data sets (DFR_DS_NO) or number of time series (DFR_TS_NO)
- changing the revisit flag (DFR_WF_REVISIT) from the previous run

However, user can change any of the above listed tasks by physically removing the results of previous run (removing EST, control table, and Catalog). This can also be done by setting the value of parameter DFR_REMOVE_EST_CAT to '1'. The user must delete the extracted data also from the Inp library.

For DFR_REMOVE_EST_CAT = 0, the ESTs, Control Table, and Catalog from previous run remain in the system. Conversely, for DFR_REMOVE_EST_CAT = 1, all data sets including ESTs, Control Table, and Catalog are deleted before forecasting.

If the value of the parameter DFR_WF_OPTIMIZATION is set to '1', the system runs in optimized mode. This means that no names are used for merchandise or location members.

Performance is greatly enhanced in optimized mode. Users can also set the value of the parameter DFR_WF_ARCHIVE to '0' to turn-off archiving.

8.1.9 Data Purge

If purge workflow parameter DFR_WF_PURGE is On (value set to '1'), then the temporary files placed on server during execution of the forecast are deleted. Purge is the last step in the forecasting process.

Set the workflow parameter DFR_WF_PURGE to '1' to run the Data Purge process.

The following temporary data sets are deleted during Purge process:

- From For library:
  - URM_F_[projid]_[x]
  - OUTSUM_F_[projid]_[x]
  - OUTSTAT_F_[projid]_[x]
- From Stage library:
  - VARS_[projid]

8.2 Standalone Execution

The solution enables execution of forecasting process without connecting to the SAS Merchandise Planning (standalone execution). This feature is useful if many test runs of a forecast are to be run or the source data is provided by an external system.

To run in stand-alone execution mode, the forecast project must run completely at least once. The macro parameter STANDALONE is set to '1' in the call to MM_FORECAST macro, which is called by DFR_STANDALONE macro.

The DFR_STANDALONE macro is called to run the standalone forecast. This macro modifies Jobparam values and calls MM_FORECAST macro to run the forecast.

For example:

```
%include "C:/sas/retail/mx_analytics/params.sas";
%include "C:/sas/retail/mx_analytics/dfr_standalone.sas";

%dfr_standalone(
             standalone_name=SA1,
             id=2378,
             forecast_start=03AUG2008,
             forecast_num_periods=52,
             kpis=SALES,
             dfr_merch_level=5,
             dfr_loc_level=2,
             dfr_time_level=5,
             dfr_filter_start=01FEB2007, dfr_filter_end=31AUG2008,
             dfr_filter_periods=26);
```

Note: In standalone execution mode, the Extract and Import routines are not processed. Additionally, standalone execution mode does not support New Item, New Location, and Optimized (dfr_wf_optimized =0) processes.

8.2.1 Standalone Parameters

The macro parameters for standalone execution are listed below.

Parameter Name	Comments	Default/Example Value
STANDALONE_NAME	Unique name for the job.	SA1
ID	Unique numeric ID for the job.	1

Parameter Name	Comments	Default/Example Value
FORECAST_START	Start forecast date ddMMMyyyy format.	03FEB2008
FORECAST_NUM_PERIODS	Number of time periods to forecast.	52
KPIS	KPI names for input.	SALES
DFR_EXTRACT	F_xxxx 2 = SAS data set 3 = CSV file	2
DFR_ARCHIVE	Optional. 1=archive 0=no archive	0
DFR_PURGE	Optional. 1=purge 0=no purge	1
DFR_TIME_LEVEL	From time level. (5=week, 4=month, 3=quarter)	5
DFR_TIME_LEVEL_TO	Optional if RECON_SINGLE=1. To Time Level (5=week, 4=month, 3=quarter).	6
DFR_MERCH_LEVEL	From Merchandise Level	9
DFR_MERCH_LEVEL_TO	Optional if RECON_SINGLE=1. To Merchandise Level.	10
DFR_LOC_LEVEL	From Location Level.	1
DFR_LOC_LEVEL_TO	Optional if RECON_SINGLE=1. To Location Level.	4
DFR_FILTER_START	Optional. If filter is used then this date must be set.	03FEB2008
DFR_FILTER_END	Optional. If filter is used then this date must be set.	31DEC2008
DFR_FILTER_PERIODS	Optional. Number of periods within date range that must be greater than 0 for the time series to be forecasted.	26
DFR_RECON_SINGLE	1= means only forecast at a single level of merchandise, location, and time. 0= multiple levels of reconciliation are required.	0
DFR_RECON_HIER	Optional if RECON_SINGLE=1. 1= merchandise hierarchy 2= location hierarchy	1
DFR_RECON_MERCH	Optional if RECON_SINGLE=1. • top merchandise level for top-down reconciliation. • bottom level for bottom-up	6

Parameter Name	Comments	Default/Example Value
	reconciliation. • middle level for middle-out reconciliation.	
DFR_RECON_LOC	Optional if RECON_SINGLE=1. • top merchandise level for top-down reconciliation. • bottom level for bottom-up reconciliation. • middle level for middle-out reconciliation.	0
DFR_IND	Independent variables that are to be considered in the forecast. Format: x.y.cause1 where cause1 is the field name of the independent variable. x and y strings are for formatting purposes only.	x.y.cause1, x.y.cause2
DFR_REVISIT	Optional. 0= Revisit process is Off. 1= Revisit process is On.	0

8.2.2 Input Data File Format

For input history, fx_[id] as a SAS data set or fx_[id].csv, where projid is unique identifier for job. If the external data set is in SAS data set format then DFR_WF_EXTRACT is set to '2'. To import from a CSV file, the DFR_WF_EXTRACT is set to '3'.

In input time series history data for forecasting, data is provided in the format shown in the table below. The file must be named fx_[id]. Note that all history from the input file is used and is assumed to be contiguous time periods.

Table 29. Format: Comma Delimited Flat File

Column Name	Data Type	Comments
Mid_n	Numeric 11	Merchandise parent level ID N=sequential list of parent levels merch_from_level to merch_to_level Example: Mid_5, mid_6, mid_7
Lid_n	Numeric 11	Location parent level ID N=level ID of parent; it must contain columns for loc_from_level to loc_to_level Example: lid_3, lid_4
Time	Date	Period start date – ddmmmyyyy format
[KPI_n]	Numeric 20.5	1…n KPIs as given by parameter. One column per KPI. Example: Sales and Units.

If the parameter REVIST is set to On (DFR_REVISIT = 1) then, user must provide F_[KPI_n] and S_[KPI_n] for each KPI. Here, F_[KPI_n] are the forecasted values and S_[KPI_n] are the standard deviations of the forecasted values in the last run.

To analyze independent variables (custom secondary effect variables) attach independent variables as columns in the input data set. The names of the independent variables such as DFR_IND_VAR1 and DFR_IND_VAR2, must be same as given in the normal execution case. See Custom Secondary Effects for details.

In case of standalone execution, the parameter (for example, test1.csv.cause1) only the third argument is taken as the column name for further forecasting process.

CHAPTER 9

Code Overview of Demand Forecasting for Retail

9.1 Process and Data Flow Summary

The following diagram shows process and the data flow in SAS Demand Forecasting for Retail.

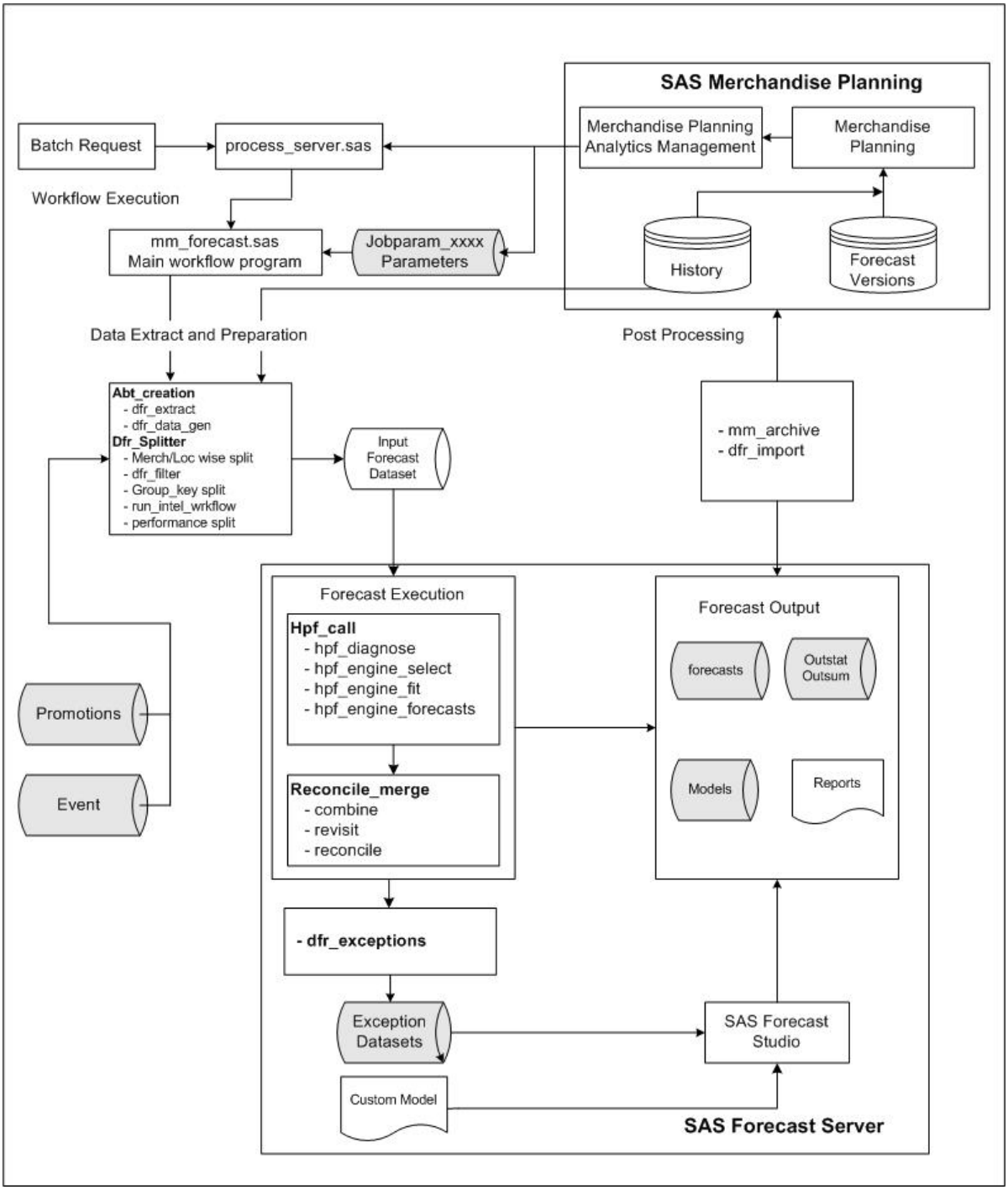


Figure 25. Process and Data Flow Summary

9.1.1 Data Set Names

The project ID (&projid) and plan worksheet ID are used as unique identifiers for all data files and job executions.

Level nomenclature example: _M1_L2 signifies merchandise level 1 and location level 2.

Table 30. Data Set Names

Process	Input Data Sets	Output Data Sets
Extract	mmdata.jobparam_[projid] history (mfinc)	input/e_[projid]
Data Preparation	input/f_[projid] input/events, promo, weather, price	input/f_[projid]
Split Input	input/f_[projid]	staged/f_[projid]_[x] . …
Filter	staged/f_[projid]_[x]	staged/filter_f_[projid]_[x] . … staged/f_[projid]_[x] . …
Forecast Diagnose	staged/f_[projid]_[x]	dia/d_EST_f_[Projid]_[x] dia/d_CAT_f_[Projid]_[x]
Forecast Select	staged/f_[projid]_[x] dia/d_EST_f_[Projid]_[x] dia/d_CAT_f_[Projid]_[x]	for/f_CAT_f_[Projid]_[x]
Forecast Fit	staged/f_[projid]_[x] for/f_CAT_f_[Projid]_[x]	for/f_EST_f_[Projid]_[x]
Forecast	staged/f_[projid]_[x] for/f_CAT_f_[Projid]_[x]	for/Urm_f_[Projid]_[x] for/OUTSUM_f_[Projid]_[x] for/OUTSTAT_f_[Projid]_[x]
Exception	for/OUTSTAT_f_[Projid]_[x]	output/fe_[project name]_[projid]
Combine3	for/Urm_f_[Projid]_[x]	for/urm_f_[Projid]_[y]
Revisit	for/urm_f_[Projid]_[y]	for/Urm_f_[Projid]_[y] dp/f_[projid]_ctrl_tab
Combine2	for/Urm_f_[Projid]_[y]	recon/URM_f_[Projid]_[y]
Reconcile	for/URM_f_[Projid]_[y]	recon/RM_f_[Projid]_[y]
Combine1	for/Urm_f_[Projid]_[y]	output/RM_f_[Projid]_M[Merch_lvl]_L[Loc_lvl]
Archive	for/outstat_f_[projid]_* for /outsum_f_[projid]_* for /rm_f_[projid]_*	archive/archive_##_[projid]_[archive date] where date is yyyymmdd
Import	staged/RM_f_[Projid]_M[Merch_lvl]_L[Loc_lvl]	output/rm_f_[projid]
Report	None	reports/fr_[project name] _[projid].htm

9.2 Workflow

Process server runs all analytic jobs and is configured by params.sas. **`mm_forecast`** in the main forecast workflow program.

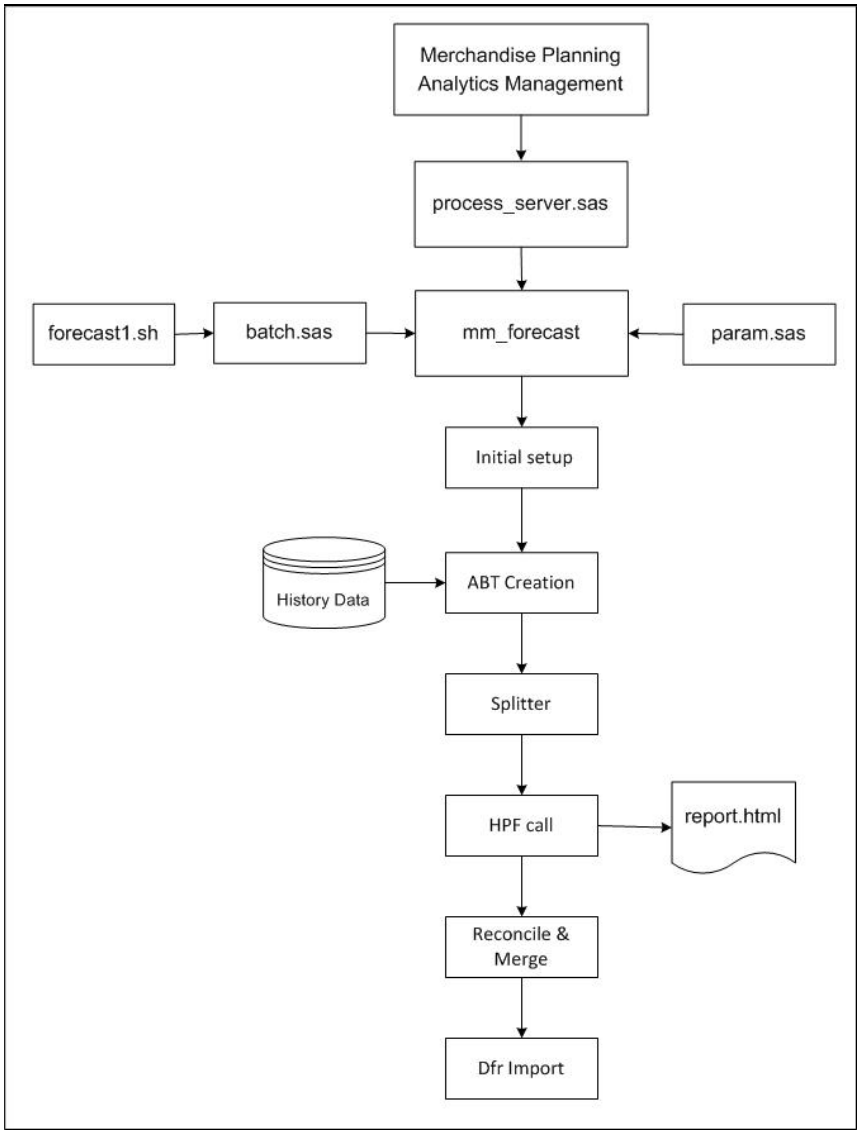


Figure 26. Execution Workflow

9.2.1 process_server.sas

Process_server is the server that initiates a call to MM_FORECAST to run a forecast job.

The main execution server calls params.sas to get the database connection and code paths. It then calls MM_FORECAST and requests are queued-up in maxdata.planworksheet.

9.2.2 MM_FORECAST

This is the main forecast workflow execution routine and is called by the Process Server to run a forecast job. It calls all the other codes for execution.

This routine runs the following macros or files. These macros are run in the order in which they are listed in the table.

Table 31. Macros in MM_FORECAST

Routine Name	Description
INITIALSETUP	This routine creates the jobparam_xxxx data set (if does not exists already), sets up Output Delivery System (ODS) and data libraries, reads, and sets

Routine Name	Description
	DFR parameters from jobparam_xxxx data set.
DFR_SETUP	This routine sets the metadata, dimension levels, and reconciliation parameters.
ABT_CREATION • DFR_EXTRACT • DFR_DATA_GEN	This routine prepares analytical base tables (ABTs). • DFR_EXTRACT Extract history including new item history Output data set: stage.e_[projid] Aggregate locations, add full time weeks, filter time-series, and create input forecast data set. Output data set: inp.f_[projid] • DFR_DATA_GEN Adds secondary effect to the input data set (f_projid) like promo, event, weather, and so on. Output data set: inp.f_[projid]
SPLITTER • SPLITTER1 • DFR_FILTER • SPLITTER2 • RUN_INTEL_WORKFLOW • SPLITTER3	This routine splits the input data sets. • SPLITTER1 This routine aggregates and prepares the data sets based on reconcile hierarchy dimension (merchandise or location). • DFR_FILTER This routine separates out the data that do not have enough forecast history. • SPLITTER2 This routine splits data sets according to merchandise or location or group key overrides. • RUN_INTEL_WORKFLOW This routine makes updates to the CTRL TABLE to store the previous run details for forecasting request. • SPLITTER3 This program splits input data sets for performance purpose.
HPF_CALL • HPF_DIAGNOSE • HPF_ENGINE • DFR_EXCEPTIONS • COMBINE3 • REVISIT	This routine calls HPFDIGNOSE, HPFENGINE. This will also produce exception data set based on filter and combine the data set (created by performance split) and finally evaluates the model. • HPF_DIAGNOSE This routine diagnoses the models. • HPF_ENGINE This routine selects and fits the diagnosed

Routine Name	Description
	models and performs forecasting. • DFR_EXCEPTIONS This routine produces exception data set based on filter data set. out.fe_[jobname]_[projid] • COMBINE3 This routine combines the separated forecasted data sets into corresponding merchandise and location levels. • REVISIT This routine decides the quality of forecasting models. **Note:** HPF_DIAGNOSE and HPF_ENGINE are run simultaneously for each split data set.
RECONCILE_MERGE • COMBINE2 • HPF_RECONCILE • COMBINE1	This routine calls COMBINE2, HPF_RECONCILE, and COMBINE1 routines. • COMBINE2 This routine combines the separated merchandise or location IDs forecasted data sets into the corresponding merchandise or location levels. • HPF_RECONCILE This routine reconciles all forecasted merchandise or location level files. • COMBINE1 This routine combines different KPIs into single data set.
DFR_IMPORT	Import forecast into merchandise planning. Data set in: stage.re_f_[projid]_m[merch lev]_l[loc level] Output: mfact_version merchandise planning table
MM_ARCHIVE	Archive the rm (forecast), outstat, and outsum data sets. Output: ARCHIVE_[projid]_[x]_[ArchiveDate yyyymmdd] Where 'x' is the archive folder number.
DFR_PURGE	Purge stage data sets.

The following figure shows the workflow of mm_forecast.sas.

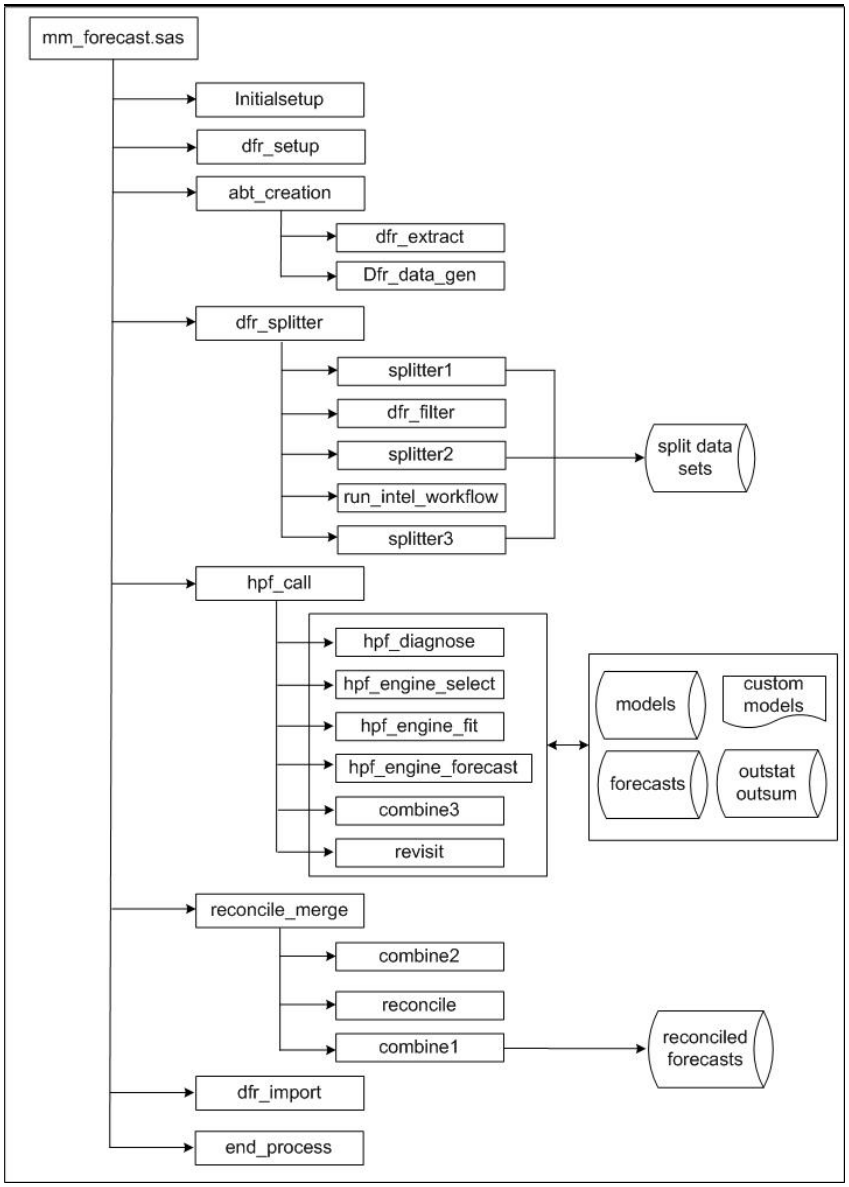


Figure 27. mm_forecast.sas Workflow

9.2.3 params.sas

Called By: process_server.sas

This program configures the main code path, data path, and database connection. It includes the utils2.sas utility file.

9.2.4 Utils2.sas

Called By: params.sas

This program contains the parameters, error handling, and ODS report utility macros.

Utilities in this program include:

- `%macro initialsetup(proj=,type=,standalone=);`

  Called at the beginning of mm_forecast to set up initial jobparam data set, ODS, and libraries.

- `%macro mm_endprocess(standalone=0);`

  Called at the end of mm_forecast to close the ODS connection and set the error codes.

- `getparams(proj=,loadrequest=0,type='ma',merch_id=0,merch_level=0,location_id=0,location_level=0,time_id=0,time_level=0,fieldkey=ALL);`

  This is the core macro that retrieves parameters at default (0) or specific levels and KPIs. **`Fieldkey`** is the forecast field from parameter, DFR_KPINAMESET.

An example of a call can be:

```
getparams(proj=&projid,merch_level=&m.,location_level=&l.,
fieldkey=SALES);

%RCSet(&syserr);
```

Where, RCSet is the error code check routine and is called after every DATA step, PROC sql, library, or file operation.

9.3 Initialsetup

Filename: utils2.sas

Macro Name: %INITIALSETUP

Called By: mm_forecast.sas

This program sets up the initial generic parameters for SAS Demand Forecasting for Retail. It creates jobparam_xxxx data set if it does not exist, and sets up ODS and data libraries,

9.4 dfr_setup

Filename: dfr_setup.sas

Macro Name: %DFR_SETUP

Called By: mm_forecast.sas

This program sets up the following metadata and dimension level parameters:

- KPI (field) metadata for extract, import, and forecast fields
- level parameters for merchandise, location, time
- reconciliation system parameters

9.5 ABT_CREATION

Filename: abt_creation.sas

Macro Name: %ABT_CREATION

Called By: mm_forecast.sas

This program prepares analytical base tables (ABTs).

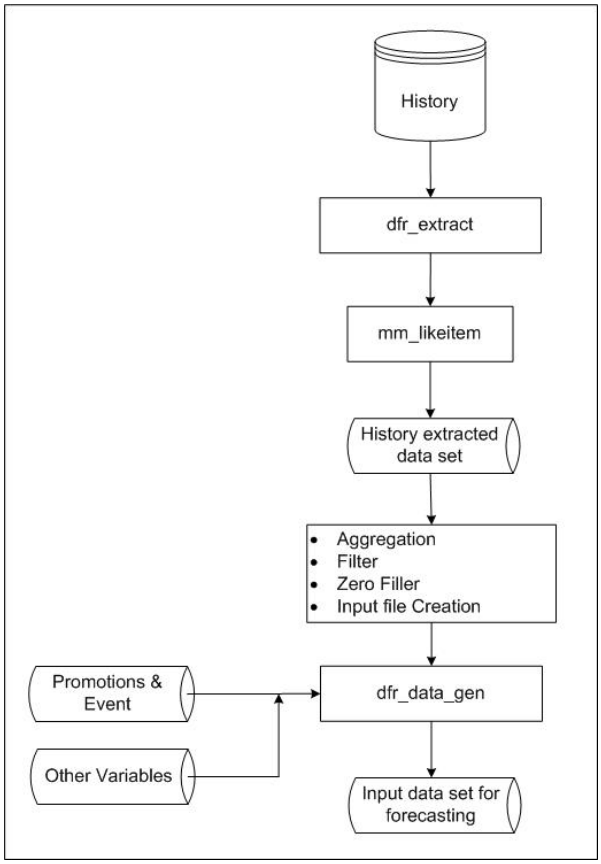


Figure 28. Data Preparation

9.5.1 DFR_EXTRACT

The DFR_EXTRACT program extracts data from history tables of SAS Merchandise Planning.

The program first calculates the beginning, end history, and forecast timeframes and then calls the MM_EXTRACT12 routine to get the history. If like-item new product processing is required then it executes that code and merges the two history extracts (regular and like-item history). The output is inp.f_[projid] data set.

After input data set extraction, the code performs the following data preparation functions before forecasting:

1. Aggregate locations (for clusters: store to cluster aggregation).
2. Add full-time weeks and ensures time gaps in the data. Uses zero values to fill.
3. Perform like-item history substitution.
4. Perform like-store substitution.
5. Write input data set inp.f_[projid].
6. Manipulate the missing values in KPI columns.

9.5.1.1 dfr_likeitem.sas

This macro finds like-item history and applies the average calculations to like-items. Average calculation is applied when more than one acts-like product is assigned to a new product. The average of history is taken for each time period that has the data.

9.5.1.2 mm_extract12.sas

This is the main history extract macro. Extract the history from the SAS Merchandise Planning system based on the given inputs of merchandise, location, and time level ID boundaries. This scope is pulled from forecast job worksheet.

This routine can run multiple extracts simultaneously against the database to speed-up the performance. The parameter MA_RSUBMIT_MAX controls the number of 'threads' to spin-off for extracting the data.

9.5.2 dfr_data_gen.sas

This program attaches the independent variable data to input sales data. It also adds number of observations to the input data set that is equal to the forecast horizon per time series. For example, if lead is 15 then 15 observations per time series for future time periods are added in the input data set. It uses DFR_DATA_EFFECT_LIST parameter.

The parameter, DFR_DATA_EFFECT_LIST specifies the effects that can be observed while forecasting. This parameter has a comma-separated list of effects. For example, DFR_DATA_EFFECT_LIST=Promo, Pantry, Delay, Halo, Price, Weather.

Note: The parameter might contain any or all of above values. The values can be in uppercase, lowercase, or mixed case characters.

9.5.2.1 Promotion

To observe the effect of promotions on the forecast:

- provide value 'PROMO' to the parameter DFR_DATA_EFFECT_LIST.
- provide name of the external CSV file in parameter DFR_EXT_PROMO_FILE.
- list all types of promotions given in **`promo_type_code`** field of the external CSV file in the parameter DFR_DATA_PROMO_TYPES separated by spaces. These values must match exactly with the **`PROMO_TYPE_CODE`** field values in the external file.

As per values in above parameters (DFR_DATA_EFFECT_LIST and DFR_EXT_PROMO_FILE) a list of one column per promotion type is added to the input sales. These columns are promo_type1_dummy, promo_type2_dummy, promo_type3_dummy, and so on, as per number of promotions. These columns are mainly indicator variable type that have values '1' or '0' depending upon whether a promotion was run during that time period or not. All these added columns are used as independent variables while forecasting.

9.5.2.2 Halo

This effect accounts for the cross-product or related product effects of promotions on a product. To observe effect of Halo, provide the value 'PROMO HALO' to parameter DFR_DATA_EFFECT_LIST.

9.5.2.3 Pantry

This effect is for pantry loading effect. This effect indicates whether there is a dip in sales in periods after promotions. To observe effect of Pantry, provide the value 'PROMO PANTRY' to parameter DFR_DATA_EFFECT_LIST.

9.5.2.4 Delay

This is for customer-delayed purchasing effect. This effect is also related to promotions. This effect can tell if there was a dip in sales before promotions. To observe effect of Delay, provide the value 'PROMO DELAY' to the parameter DFR_DATA_EFFECT_LIST.

9.5.2.5 Price

Price data is imported externally through a CSV file in dfr_utils.sas. Provide value 'PRICE' to DFR_DATA_EFFECT_LIST and name of the external CSV file in DFR_EXT_PRICE_ABT. Effect on demand due to changes in price can be measured based on price elasticity of demand. Price data is attached with sales data to figure out the impact of price on forecasts. If future data (time periods for which forecasting is being done) for price is available, then effect of price is included in the forecast.

9.5.2.6 Weather

Weather data is imported externally through a CSV file in dfr_utils.sas. Provide value 'WEATHER' to the parameter DFR_DATA_EFFECT_LIST and name of the external CSV file in DFR_EXT_WEATHER_ABT. Weather conditions can also affect the demand. Currently, the solution considers low and high temperatures so that in case of adverse weather forecasts, demand can be adjusted accordingly. The future data of weather is used to adjust the demand that is being forecast.

Input Data Sets	Output Data Sets
inp.f_[projid]	inp.f_[projid]. A copy of input data set f_[projid] is kept in stage directory.

9.5.2.7 Events

This effect follows the same principle as Promotions. Increase or decrease in demand can be measured by events. For the list of supported events, see *SAS High-Performance Forecasting 2.3: Reference Guide*.

Event data is imported externally through a CSV file in dfr_utils.sas. Provide name of the external CSV file in DFR_EXT_EVENT_FILE. Mention location-specific as well as generalized events in the external file only. SAS Demand Forecasting for Retail automatically applies the generalized events in all the locations and different location-specific events through EVENTBY data-set option. For standard events that can be supported through SAS High-Performance Forecasting, see *SAS High-Performance Forecasting 2.3: Reference Guide*.

All the external files must be placed in **`data/input`** directory.

9.6 SPLITTER

Filename: dfr_splitter.sas

Macro Name: %Splitter

Called By: mm_forecast.sas

This program splits the input data sets.

9.6.1 SPLITTER1

This program performs the following tasks.

1. Splitting the code history files into different data sets depending upon the dimension to be reconciled and corresponding from and to levels of merchandise and location. The core history is also split according to KPIs. Splitting does not happen according to the levels of non-reconciled dimension.

2. Aggregating input data. All KPIs are aggregated using PROC SUMMARY.

3. Creating EST tables for forecasting request if they are not available. These EST tables store the forecasting models.

This program populates 'META' table. The name format for this table name is F_&proj._meta. It is a metadata table that stores information about the forecasting request that is under process. The table has following columns.

Table 32. Meta Table

Column Name	Purpose
DS_NAME	Stores the name of the split data set. The name of the split data set has the following format: F_&proj._M&merch_level._L&loc_level._&M_id _ &L_id _&KPI The split data set is always at the lowest level of the non-reconciled dimension.
DS_LIB	Library in which split data set is stored. It is 'dp' normally.
M_LVL	Merchandise level of split data set.
L_LVL	Location level of split data set.
M_ID	Merchandise ID. Is '0' at the end of execution of splitter1.sas.
L_ID	Location ID. Is '0' at the end of execution of splitter1.sas.
KPI	Name of the KPI that the split data set contains.

The number of split data sets that are generated is equal to the difference between **`xxx_to_level`** and **`xxx_from_level`** of the reconciled dimension for each KPI. If user does not want to perform reconciliation and needs lowest level forecasts only, then the input history file is not split. It is used as it is for forecasting.

9.6.2 DFR_FILTER

The filter process eliminates a series without a percentage of past history data that is greater than zero. This routine eliminates old products with no recent sales, negative sales only, or new products without enough history to forecast.

This program creates a filtered file for forecasting into **`staged/f_[projid]_1`**, **`staged/f_[projid]_2`**, and so on.

The filter code counts the number of periods that are greater than zero in the parameter DFR_FILTER_PRIOR_PERIODS from the list of the prior periods. If number of periods is less than or equal to the value in the parameter DFR_FILTER_PRIOR_HIST then the series is eliminated.

DFR_FILTER_PRIOR_HIST is the number of periods of history within number of prior weeks and DFR_FILTER_PRIOR_PERIODS is from the last loaded week backward in time. Parameters are always expressed in number of time intervals.

When running in stand-alone mode additional parameters DFR_FILTER_START_DATE and DFR_FILTER_END_DATE are used. These parameters are set in ddMMMyyyy format. The start date and end date defines the period of time that is checked by the filter. If the number of periods between the defined start date and end date is less than DFR_FILTER_PRIOR_PERIODS, then the time-series is dropped.

9.6.3 SPLITTER2

The program Splitter2 works in two ways:

i. Data set split when override is based on Merchandise ID or Location ID.

ii. Data set split when override is based on Group Key.

Note: These overrides (Merchandise/Location ID and Group Key) cannot be used together at the same time. Only one override (either Merchandise/Location override or Group Key override) can be used at a time.

9.6.3.1 Data Set Split Based on Merchandise ID or Location ID

This program splits the output of Splitter1 if merchandise ID or location ID has been overridden in Jobparam data set. If no ID has been overridden in the Jobparam, then this program is not run.

This program makes following assumptions about the entries in Jobparam data set.

- The entry of unreconciled dimension is either equal to '0' or at the lowest level. Therefore, Jobparam cannot have an entry of unreconciled dimension that is other than '0' or the lowest level.
- Merchandise and location IDs are not overridden if corresponding levels are not overridden. This means that in Jobparam, an entry cannot be MERCH_LEVEL=0, LOCATION_LEVEL=0, MERCH_ID=1001, and LOCATION_ID=2001.
- Reconciled dimension cannot have an entry that is less than 0 and more than its **`xxx_to_level`**.

All observations that violate these assumptions are deleted from the Jobparam data set and remaining observations are taken for processing. Forecasting code runs on each record in the Jobparam data set. The order in which each record is processed is decided by the precedence logic that is shown in the tables below.

Example of Splitting

Assume that an input data set has four Merchandise Levels and three Location Levels. It has two KPIs to be forecasted namely, Sales and Units. Assume that the data set has following entries in Jobparam and reconciliation dimension is Merchandise.

Table 33. Jobparam Data Set

Merch_Level	Loc_Level	Merch_ID	Loc_ID	Group_Key	FIELKEY	Parameter Set
4	3	4001	3001	0	sales	P1
4	3	4001	3001	0	units	P2
4	3	0	0	0	sales	P3
4	3	0	0	0	units	P4
0	0	0	0	0	all	P5

For the above Jobparam, the following data sets are generated after splitting process is completed.

Table 34. Data Sets after Splitting Process

Data Set Description	Forecast Action
F_[projid]_M4_L3_4001_3001_0_0_sales	M4L3 data set having Mid=4001 and Lid=3001 and KPI=sales. Parameter set P1 is used for forecasting.
F_[projid]_M4_L3_4001_3001_0_0_units	M4L3 data set having Mid =4001 and Lid=3001 and KPI=units. Parameter set P2 is used for forecasting.
F_[projid]_M4_L3_0_0_0_0_sales	M4L3 data set having all IDs except Mid =4001 and Lid=3001 and KPI=sales. Parameter set P3 is used for forecasting.
F_[projid]_M4_L3_0_0_0_0_units	M4L3 data set having all IDs except Mid =4001 and Lid=3001 and KPI=units. Parameter set P4 is used for forecasting.

Data Set Description	Forecast Action
F_[projid]_M3_L3_0_0_0_0_sales	M3L3 data set having all Mids and Lids and KPI=sales. Parameter set P5 is used for forecasting.
F_[projid]_M3_L3_0_0_0_0_units	M3L3 data set having all Mids and Lids and KPI=units. Parameter set P5 is used for forecasting.
F_[projid]_M2_L3_0_0_0_0_sales	M2L3 data set having all Mids and Lids and KPI=sales. Parameter set P5 is used for forecasting.
F_[projid]_M2_L3_0_0_0_0_units	M3L3 data set having all Mids and Lids and KPI=units. Parameter set P5 is used for forecasting.
F_[projid]_M1_L3_0_0_0_0_sales	M1L3 data set having all Mids and Lids and KPI=sales. Parameter set P5 is used for forecasting.
F_[projid]_M1_L3_0_0_0_0_units	M1L3 data set having all Mids and Lids and KPI=units. Parameter set P5 is used for forecasting.

9.6.3.2 Data Set Split Based on Group Key

This program splits output of Splitter1. User can apply forecasting parameters according to user-defined group key. The user-defined group key can be borrowed from stock profiling. Group key helps the user to assign same parameters to all the time series with same group key. The group key is applied only at lowest level and can be used as an alternative hierarchy.

If DFR_USER_DEFINE_GK = 1 and new override observation for a group key along with any forecasting parameter is inserted in Jobparam data set then split process is run.

If DFR_USER_DEFINE_GK = 1 and no override observation in Jobparam data set then data set is not split.

The valid group key value needs to be specified in the override observation.

Table 35. Jobparam Data Set

Merch_Level	Loc_Level	Merch_ID	Loc_ID	Group _Key	Fieldkey	Parameter Set
4	3	0	0	1	sales	P6
4	3	0	0	2	sales	P7
4	3	0	0	0	sales	P8

Table 36. Data Sets after Splitting Process

Data Set Description	Forecast Action
F_[projid]_m4_l3_0_0_1_0_ SALES	M4L3 data set having group key=1 and KPI=sales. Parameter set P6 will be used for forecasting.
F_[projid]_m4_l3_0_0_2_0_ SALES	M4L3 data set having group key=2 and KPI=sales. Parameter set P7 will be used for forecasting.
F_[projid]_m4_l3_0_0_0_0_ SALES	M4L3 data set having group key=0 and KPI=sales. Parameter set P8 will be used for forecasting.

9.6.4 RUN_INTEL_WORKFLOW

The Revisit and Intelligent Workflow options have an impact on the forecast workflow run. If Revisit and Intelligent Workflow options are Off, then flags set by the user for DFR_WF_DIAGNOSE, DFR_WF_SELECT, DFR_WF_FIT, and DFR_WF_FORECAST are used as is.

Note: Intelligent Workflow ensures valid forecast run. The system never turns Off a workflow item that a user has requested (set to On). It ensures running of the prerequisite steps if a request is made.

The rules for Intelligent Workflow are:

1. Any downstream process requires previous processes to be run first. For example, Select requires Diagnose process run first. The run can be in current execution or a previous run of the same job.
2. No gaps in process requests are allowed. Therefore, if Select and Forecast are On, then Fit must also run.
3. All workflow steps that are explicitly turned On by the user are executed.

The following example illustrates the Intelligent Workflow.

In Run 1, the user sets up Select and Forecast On. Since no prior execution data exists, to run a forecast, all steps must run so the system overrides the user input and ensures that all steps are run.

On Run 2, the user leaves select and forecast on. The system detects that select is being run so fit must run if select data has changed and a forecast is to be run. It sets fit to On.

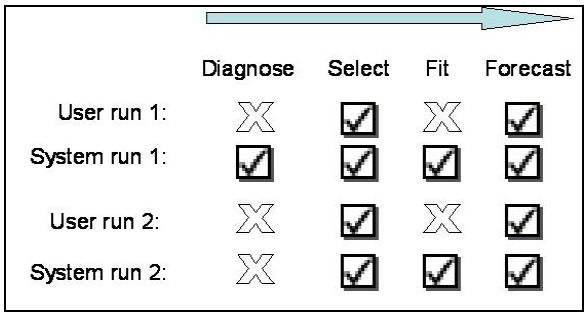


Figure 29. Intelligent Workflow Execution

Forecast Workflow Control Tables

These tables store history of workflow steps that are run for forecasting requests. They are used as input before any workflow step run, to select the time series that is needed to be executed in the workflow step. The following section gives the details of the control table.

9.6.4.1 Intel_Workflow

This program updates a control table that keeps history of last runs of the forecast request. The structure of this control table CTRL_TAB is given in the following table. There is one CTRL_TAB for each forecasting request. The name of CTRL_TAB is in the format dp.f_[projid]_ctrl_tab.

Table 37. CTRL_TAB Table

Column Name	Comment
PROJID	Project ID of the forecasting request.
M_LVL	Merchandise Level.
L_LVL	Location Level.
M_ID	Merchandise ID.
L_ID	Location ID.
KPI	KPI Name.
GROUP_KEY	The group key number.
UMD_DIA	Modified value of DFR_WF_DIANOSE parameter given by the

Column Name	Comment
	user.
UMD_SEL	Modified value of DFR_WF_SELECT parameter given by the user.
UMD_FIT	Modified value of DFR_WF_FIT parameter given by the user.
UMD_FOR	Modified value of DFR_WF_FORECAST parameter given by the user.
HIST_DIA	Last run value of DFR_WF_DIANOSE parameter.
HIST_SEL	Last run value of DFR_WF_SELECT parameter.
HIST_FIT	Last run value of DFR_WF_FIT parameter.
HIST_FOR	Last run value of DFR_WF_FORECAST parameter.
REV_DIA	Revisit recommendation for diagnose process.
REV_SEL	Revisit recommendation for select process.
REV_FIT	Revisit recommendation for fit process.
REV_FOR	Revisit recommendation for forecast process.
HIST_MOD_DTTM	Datetime stamp when history columns (hist_***) were last updated.
REV_MOD_DTTM	Datetime stamp when revisit recommendation columns (rev_***) were last updated.
UMD_MOD_DTTM	Datetime stamp when user modified columns (umd_***) were last updated.
PROCESS	Error Flag.

9.6.4.2 Processing Logic

The processing of Intelligent Workflow considers:

- All distinct combinations of merchandise level, merchandise ID, location level, location ID, and KPI from a forecasting request are taken. Existence of each combination is checked in the CTRL_TAB.
- If combination is not found in the CTRL_TAB then it is treated as new request and entry of that combination is done in the control table. All history columns of CTRL_TAB are updated to '0' and revisit recommendation columns are updated to '1'.
- If combination of merchandise level, merchandise ID, location level, location ID, and KPI from a forecasting request exists and DFR_IVAR_CHNG parameter is '1', then update all history columns of CTRL_TAB to '0' and revisit recommendation columns to '1'.
- If combination of merchandise level, merchandise ID, location level, location ID, and KPI from a forecasting request exists and DFR_IVAR_CHNG parameter is '0' then, check the parameter DFR_WF_FORECAST_INTEL_EXE. If it is '0', then all user modified columns (UMD_***) are updated with DFR_WF_*** columns in the input data set.

If parameter DFR_WF_FORECAST_INTEL_EXE is '1', then the system intelligence is set to On. In this case user's workflow parameters are modified.

If DFR_WF_REVISIT is '1', then revisit recommendation columns in CTRL_TAB and user's input work flow parameters are used to update the user modified columns (UMD_***).

If DFR_WF_REVISIT is '0', then history columns in CTRL_TAB and user's input work flow parameters are used to update the user modified columns (UMD_***).

For example, if user's workflow parameters are:

DFR_WF_DIA=0, DFR_WF_SEL=1, DFR_WF_FIT=0 DFR_WF_FOR=1

This request is already processed in the previous run and therefore entries in CTRL_TAB are as follows:

HIST_DIA=1, HIST_SEL=1, HIST_FIT=1, HIST_FOR=1

Now, if DFR_WF_FORECAST_INTEL_EXE is On, then the following updates are done in UMD_*** columns of the control table.

UMD_DIA=0, UMD_SEL=1, UMD_FIT=1, UMD_FOR=1

Note: If user's input are invalid, system automatically corrects it to UMD_FIT=1.

If DFR_WF_FORECAST_INTEL_EXE is Off, then there are no updates in UMD_*** columns of the control table. Therefore UMD_*** columns will be exactly same to DFR_WF_*** columns.

UMD_DIA=0, UMD_SEL=1, UMD_FIT=0, UMD_FOR=1

Note: All the user's workflow parameters that have value '1' are not modified by this process. If the value of user's workflow parameters is '0', the system takes the decision to make them '1' depending on the historical run or revisit recommendations.

9.6.5 SPLITTER3

Splitter3 routine splits the data sets for performance benefit. Data sets split works based on number of time-series value (DFR_NO_TS) or number of data set values (DFR_NO_DS) that are provided in Jobparam. The data sets created after Splitter2 are used as input to Splitter3. This routine is run when value for DFR_WF_SPLIT_TYPE parameter is set in Jobparam data set. If this value is '0', then no split is performed, if '1' then split performed is based on time-series value else if it is '2', then split is based on the number of data sets value.

1. **Split Based on Number of Data Sets**

   In Jobparam, DFR_NO_DS parameter value needs to be specified. If it is '0' then no split process is applied. Minimum two or more judicious positive integer values must be specified for splitting process. If DFR_NO_DS = 20, the maximum data set count is 20 and not each data set is split into 20 data sets. The routine optimally decides which data sets must split further. It takes the input data set count, which is the number of data sets created after Splitter2. If data set count is eight and value for DFR_NO_DS = 20, then using these eight data sets 20 data sets are created.

Table 38. Data Sets after Splitting Process

Ds_ref_no	Data Set Name	Data Set Description
1	F_[projid]_[x]	F_3007_m6_l4_0_0_0_1_SALES
2	F_[projid]_[x+1]	F_3007_m6_l4_0_0_0_2_SALES
3	F_[projid]_[x+2]	F_3007_m6_l4_0_0_0_3_SALES
4	F_[projid]_[x+3]	F_3007_m6_l4_0_0_0_4_SALES
5	F_[projid]_[x+4]	F_3007_m6_l4_0_0_0_5_SALES

2. **Split Based on Number of Time Series**

   In Jobparam, DFR_NO_TS parameter value needs to be specified. If it is '0', then no split process is applied. Minimum two or more judicious positive integer value needs to be specified for splitting process. If DFR_NO_TS = 200 then each data set contains maximum

200 time-series data or less. The data sets created after Splitter2 routine is used as input to this process.

Table 39. Data Sets after Splitting Process

Ds_ref_no	Name	Data Set Description
1	F_[projid]_[x]	F_3007_m6_l4_0_0_0_1_UNITS
2	F_[projid]_[x+1]	F_3007_m6_l4_0_0_0_2_ UNITS
3	F_[projid]_ [x+2]	F_3007_m6_l4_0_0_0_3_ UNITS
4	F_[projid]_ [x+3]	F_3007_m6_l4_0_0_0_4_ UNITS
5	F_[projid]_ [x+4]	F_3007_m6_l4_0_0_0_5_ UNITS

Caution: Performance split depends on the values of parameters DFR_NO_DS and DFR_NO_TS. Therefore these values must be specified judiciously.

9.7 HPF_CALL

Filename: hpf_call.sas

Macro Name: %HPF_CALL

Called By: mm_forecast.sas

SAS High-Performance Forecasting 2.3 is automated with this workflow to execute forecasts. A separate Diagnose, Fit, Select, and Forecast run is made for each distinct set of time series as defined by merchandise level, location level, and KPI.

The split and combine logic divide input history into separate data sets for each forecast execution and recombine them after the run. The Revisit and Reconcile logic is executed in one pass after the forecast executions.

If individual merchandise and location ID parameters are provided, the forecast routines are run for these distinct time-series as well. If ID or individual time-series are run then the Splitter2 routine is used to split the data sets and Combine2 is used to recombine these individual time-series. The Splitter1 and Combine1 routines will split and recombine the level and KPI sets of time-series.

Note: With appropriate configuration, this routine can also be executed in parallel mode.

9.7.1 HPF_DIAGNOSE

This program runs the PROC HPFDIAGNOSE. All the required parameters for the procedure are initialized by the macro GETPARAMS, before the procedure runs. First, all the time series in the forecasting request that are to be diagnosed are selected. This macro is runs for each merchandise level, location level, and the KPI. If there is an override at a particular level, then those IDs are diagnosed using overridden values. The model repository and EST tables are created in the Dia library. All the time series that have been diagnosed are updated in the control table making the HIST_DIA column as 1. User can suggest their models and also force the SAS Demand Forecasting for Retail to explicitly use their model.

9.7.2 HPF_ENGINE

This program accepts a parameter called as TASK, which can be Select, Fit, or Forecast.

When TASK=Select, this program runs the PROC HPFENGINE with TASK=Select option. All the required parameters for the procedure are initialized by the macro GETPARAMS, before the

procedure runs. First, all the time series in the forecasting request for which models need to be 'Select' are selected. This macro runs for each merchandise level, location level, and the KPI. If there is an override at a particular level, then those IDs are selected using overridden values. The model repository and EST tables (created in 'For' library) are used for model selection. All the time series that are selected successfully are updated in the control table setting HIST_SEL, HIST_FIT, and HIST_FOR to '1'.

When TASK=Fit, this program runs the PROC HPFENGINE with TASK=Fit option. All the required parameters for the procedure are initialized by the macro GETPARAMS, before the procedure runs. First, all the time series in the forecasting request for which models need to be 'Fit' are selected. This macro runs for each merchandise level, location level, and the KPI. If there is an override at a particular level, then those IDs are selected using overridden values. The model repository and EST tables (created in 'For' library) are used for the model selection. All the time series that are selected are updated in the control table by setting the HIST_FIT and HIST_FOR to '1'.

When TASK=Forecast, this program runs the PROC HPFENGINE with TASK=Forecast option. All the required parameters for the procedure are initialized by the macro GETPARAMS, before the procedure runs. First, all the time series in the forecasting request for which models need to be 'Forecast' are selected. This macro runs for each merchandise level, location level, and the KPI. If there is an override at a particular level, then those IDs are selected using overridden values. The model repository and EST tables (created in the 'For' library) are used for model selection. All the time series that are selected successfully are updated in the control table by setting the HIST_FOR column to '1'.

The user has the choice to either retain the negative forecast or change the negative forecast values to zero.

The forecasting variable can only be adjusted by the independent variables using the ADJUST statement in HPFENGINE. The adjusting variables must be one of the independent variables provided by the INPUT statement.

The program HPF_ENGINE also generates a table called Forecast Meta. This table has all the levels for which forecasts have been generated and their corresponding data set names.

Note: The forecast horizon might increase when intermittent demand models (IDM) is selected.

The name format for this table name is F_&proj._Forecast_meta. This is a metadata table that stores information about the forecasted data sets. This table has the following columns.

Table 40. Forecast Meta Table

Column Name	Purpose
DS_NAME	Name of the forecasted data set URM_F_[projid]_[x/y]. The number (x or y) depend on the total levels of reconciled dimension.
DS_REF_NO	Data set reference number.
DS_LIB	Library in which forecasted data set is stored. Normally, it is For.
M_LVL	Merchandise Level of split data set.
L_LVL	Location Level of split data set.
M_ID	Merchandise ID. This is '0' at the end of execution of splitter1.sas.
L_ID	Location ID. This is '0' at the end of execution of splitter1.sas.
GROUP_KEY	User-defined key to group different items based on business sense.
P_SPLIT	Performance split key.
KPI	Name of the KPI that the split data set contains.

DS_DESC	Data set description.
REF_MERCH_ID	Corresponding merchandise ID for reference.
REF_LOCATION_ID	Corresponding location ID for reference.

9.7.3 DFR_EXCEPTIONS

Creates an exception data set based on parameter filter that can be loaded into SAS Forecast Studio. See section Exception Management Using SAS Forecast Studio for details to load data set into SAS Forecast Studio. The results of studio parameters and models can be added back into the parameter data set and custom model file so as to influence the next run of the forecast job.

Note: In case of multiple KPI forecasting, the exception data set is produced only for those time series for which the filter criteria (supplied by user) matches with all KPIs.

9.7.4 Combine3

The code combines the forecasted data sets that are created after the execution of PROC HPFENGINE. If data sets are split based on number of time-series or number of data sets, then Combine3 routine is applied and the data sets are combined. This routine combines all the data sets (if any) that are split for performance purpose.

9.7.5 Revisit

This process runs only after an elapsed time, which is mentioned in parameter, &REVISIT_OBS. Therefore, if &REVISIT_OBS value is '7 weeks', then the revisit process is fired after the forecasting process is run for seven times. This is because PROC SHEWHART needs certain minimum number of observations to decide the quality of the models. In this case, seven records that are most recent time periods and have forecasted and actual values, are given as input to PROC SHEWHART. Revisit process runs for each KPI, limit data sets (inner and outer) as required for PROC SHEWHART are generated from user's input parameter REVISIT_OUTER and REVISIT_INNER and CTRL TABLE is updated for the good and bad time series.

9.8 RECONCILE_MERGE

Filename: dfr_reconcile_merge.sas

Macro Name: % RECONCILE_MERGE

Called By: mm_forecast.sas

This routine combines merchandise, location, or group key overrides and then applies reconcile process. Finally, it merges multiple KPI data sets.

9.8.1 COMBINE2

The code combines the output of PROC HPFENGINE. The output from SAS High-Performance Forecasting shows a data set corresponding to only those data sets that have overridden IDs. these data sets are merged into corresponding merchandise and location level data sets.

9.8.2 HPF_RECONCILE

The code runs the PROC HPFRECONCILE for the records in the FORECAST CONTROL table. The reconciliation can be done by Top-Down, Bottom-Up, or Middle-Out approach.

9.8.3 COMBINE1

The code combines the output of reconciliation. All KPIs at a particular Merchandise or Location level are combined in a single data set. Therefore, at this stage the final data sets consist of all the KPIs and their merchandise and location level.

9.9 DFR_IMPORT

Import involves the EXCEPTION, ARCHIVE, IMPORT, and PURGE processes. The following figure depicts post-processing.

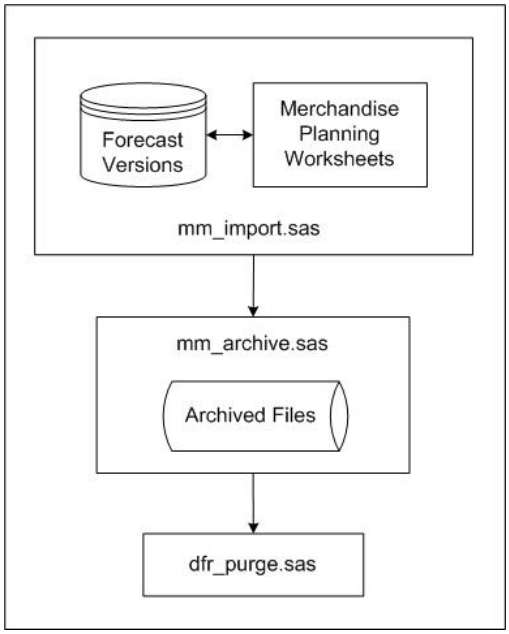


Figure 30. Post Processing

Filename: dfr_import.sas

Macro Name: %DFR_IMPORT

Called By: mm_forecast.sas

If DFR_WF_IMPORT=1 and parameter STANDALONE =0 (connected to the database), then the IMPORT function is run. It combines the forecast output data, aggregates up the time hierarchy, and copies data into the mfact_version merchandise planning version table. Existing data with the same coordinates within mfact_version is replaced. DFR_IMPORT_VERSION parameter defines the version of data to be loaded.

If DFR_IMPORT_VERSION is set to > 0 in Jobparam data set, then the data is loaded into mfact_version table. Else, if DFR_IMPORT_VERSION is set to ‘0’ in Jobparam data set, then the data is loaded into a non mfact_version table (a fact table without version keys). This table must have partitions of mfver_[time_level]_[time_id].

When multiple jobs write data to the same version they must use the same lowest level of time. For example, if one job has week as the lowest time level then all other jobs writing to the same version must also use week as the lowest level.

9.10 MM_ARCHIVE

Filename: mm_archive.sas

Macro Name: %MM_ACRHIVE

Called By: mm_forecast.sas

If the parameter DFR_WF_ARCHIVE=1 then ARCHIVE function is run. It creates an **`archive`** folder and copies the staged library Outstat, rm (forecast), and Outsum files. An **`archive`** folder is created for each projid. The user can configure the maximum number of saved **`archive`** folders by defining the value of DFR_ARCHIVE_COUNT parameter. The default value of this parameter is '3'.

9.11 DFR_PURGE

Filename: dfr_purge.sas

Macro Name: %DFR_PURGE

Called By: mm_forecast.sas

Deletes all working files saved to disk after running the forecast. These files are:

- for/outsum_f_[projid]* files
- for/outstat_f_[projid]* files
- for/outfor_f_[projid]* files
- for/urm_f_[projid]* files
- stage/vars_[projid]* files

CHAPTER

10 Implementation

10.1 Implementation Overview

This section provides a summary of implementation tasks, different project roles involved, and considerations for SAS Demand Forecasting for Retail. The SAS Consulting department provides a full explanation of implementation tasks and their scope.

10.1.1 Project Roles

Project Manager

The project manager coordinates all analysis, implementation, validation, and customer sign-off steps.

Business Consultant

A business consultant works with the forecast analyst and technical consultant to ensure that forecasts are created with correct merchandise parameters, so that the forecast data supports the merchandise planning business process and data requirements.

Merchandise Planning Consultant

A merchandise planning consultant is involved to help formulate and validate the business process requirements for the forecast data. The merchandise planning consultant can also help to set up the forecast projects and train the forecast administrators.

Forecast Analyst

A forecast analyst determines the nature of historical data, merchandise, and other analytical questions. Sample data might be necessary to support this exploration and analysis. An analyst might be involved in the initial analysis, setting up specific configuration parameters and models for the forecast, and in follow-up ‘tuning’ exercises to ensure ongoing accuracy of the forecasts.

Technical Consultant

A technical consultant helps in installing the software and modifying the forecast version database tables. The technical consultant can also help to set up the forecast projects and train the forecast administrators.

10.1.2 Requirement Assessment

The following information must be gathered for each project implementation. The information is used to set up the metadata, database, and forecast projects in the SAS Merchandise Planning suite.

- Business process steps and usage: pre-season versus in-season
- Merchandise levels

- ❑ Location levels
- ❑ Time horizon
- ❑ Time level
- ❑ Frequency
- ❑ Forecast version
- ❑ Reconciliation approach
- ❑ KPI(s) to forecast
- ❑ New style level forecasts, if any. Possibilities of acts-like assignments
- ❑ Batch versus ad hoc requirements
- ❑ Secondary variables needed such as events, promotional data, price, weather, and so on, and the availability of the data for these variables
- ❑ Data preparation requirements
- ❑ Reporting and forecast accuracy measurement requirements
- ❑ Runtime window availability

Note: Other implementation tasks include — Metadata Configuration, Forecast Project Creation, Data and Forecast Validation, Forecast Tuning, Training, and Production.

Glossary

assortment

a retailer's selection of merchandise.

assortment plan

a list of merchandise, which indicates what should be stocked in a particular category.

dimensions

a data warehouse dimension provides the means to "slice and dice" data in a data warehouse. Dimensions provide structured labeling information to otherwise unordered numeric measures. For example, "Customer", "Date", and "Product" are all dimensions that could be applied meaningfully to a sales receipt.

halo and cannibalization effects

impact of the sale of products/categories on each other. Halo is a positive effect and cannibalization is a negative effect.

key performance indicators (KPI)

are financial and non-financial metrics used to quantify objectives to reflect the performance of an organization.

pre-season planning

create more accurate pre-season forecasts based on consumer demand and last season’s performance.

in-season planning

safeguard revenues and margins through a planning lifecycle approach to pricing that includes everyday pricing, promotional planning, and markdown optimization.

analytical base tables (ABT)

analytical base tables are input tables for SAS® Enterprise Miner, for performing data analysis.

analysis of variance (ANOVA)

is a collection of statistical models, and their associated procedures, in which the observed variance is partitioned into components due to different explanatory variables.

cannibalization effect

if a promotion on a given product results in a decrease in sales of other products the product is said to cannibalize these products.

customer delayed purchase

a decrease in sales in the periods leading to a promotion sometimes occurs. This effect is defined as customer-delayed purchasing.

halo effect

promotion of one product increases sale of another product the effect is defined as a halo effect.

mean absolute percentage error (MAPE)

is a measure of accuracy in a fitted time series value in statistics, specifically trending. It usually expresses accuracy as a percentage.

pantry loading

in the period after a promotion is run, it is very likely that sales will drop below the normal level as people have used the opportunity to purchase the product for use in the periods after the promotion, resulting in lower sales following the promotional period. This effect is defined as pantry loading.

price elasticity

it is the ratio of the proportional change in demand with respect to proportional change in price.

stock keeping unit (SKU)

it is a uniquely identifiable line within a product range. A particular product can have different variations like 20 per cent extra free, price marked, and so on. Each of these variations would be a unique SKU.