Oracle® Retail Demand Forecasting

Configuration Guide

Release 13.0.1

June 2008

Oracle® Demand Forecasting Configuration Guide, Release 13.0.1

Primary Author: Gary O'Hara

Value-Added Reseller (VAR) Language

(i) the software component known as **ACUMATE** developed and licensed by Lucent Technologies Inc. of Murray Hill, New Jersey, to Oracle and imbedded in the Oracle Retail Predictive Application Server – Enterprise Engine, Oracle Retail Category Management, Oracle Retail Item Planning, Oracle Retail Merchandise Financial Planning, Oracle Retail Advanced Inventory Planning and Oracle Retail Demand Forecasting applications.

(ii) the **MicroStrategy** Components developed and licensed by MicroStrategy Services Corporation (MicroStrategy) of McLean, Virginia to Oracle and imbedded in the MicroStrategy for Oracle Retail Data Warehouse and MicroStrategy for Oracle Retail Planning & Optimization applications.

(iii) the **SeeBeyond** component developed and licensed by Sun MicroSystems, Inc. (Sun) of Santa Clara, California, to Oracle and imbedded in the Oracle Retail Integration Bus application.

(iv) the **Wavelink** component developed and licensed by Wavelink Corporation (Wavelink) of Kirkland, Washington, to Oracle and imbedded in Oracle Retail Store Inventory Management.

(v) the software component known as **Crystal Enterprise Professional and/or Crystal Reports Professional** licensed by Business Objects Software Limited ("Business Objects") and imbedded in Oracle Retail Store Inventory Management.

(vi) the software component known as **Access Via™** licensed by Access Via of Seattle, Washington, and imbedded in Oracle Retail Signs and Oracle Retail Labels and Tags.

(vii) the software component known as **Adobe Flex™** licensed by Adobe Systems Incorporated of San Jose, California, and imbedded in Oracle Retail Promotion Planning & Optimization application.

(viii) the software component known as **Style Report™** developed and licensed by InetSoft Technology Corp. of Piscataway, New Jersey, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(ix) the software component known as **WebLogic™** developed and licensed by BEA Systems, Inc. of San Jose, California, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

(x) the software component known as **DataBeacon™** developed and licensed by Cognos Incorporated of Ottawa, Ontario, Canada, to Oracle and imbedded in the Oracle Retail Value Chain Collaboration application.

Contents

Preface

Oracle Retail Configuration Guides are designed so that you can view and understand the application's "behind-the-scenes" processing, including such information as the following:

- Key system administration configuration settings
- Technical architecture
- Functional integration dataflow across the enterprise
- Batch processing

Audience

Anyone who has an interest in better understanding the inner workings of the RDF system can find valuable information in this guide. There are three audiences in general for whom this guide is written:

- System analysts and system operation personnel:
  - who are looking for information about RDF processes internally or in relation to the systems across the enterprise.
  - who operate RDF on a regular basis.
- Integrators and implementation staff who have the overall responsibility for implementing RDF into their enterprise.
- Business analysts who are looking for information about processes and interfaces to validate the support for business scenarios within RDF and other systems across the enterprise.

Related Documents

For more information, see the following documents:

- *Oracle Retail Demand Forecasting Release Notes*
- Oracle Retail Predictive Application Server documentation

Customer Support

- https://metalink.oracle.com

When contacting Customer Support, please provide:

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step-by-step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.

Review Patch Documentation

For a base release (".0" release, such as 13.0), Oracle Retail strongly recommends that you read all patch documentation before you begin installation procedures. Patch documentation can contain critical information related to the base release, based on new information and code changes that have been made since the base release.

Oracle Retail Documentation on the Oracle Technology Network

In addition to being packaged with each product release (on the base or patch level), all Oracle Retail documentation is available on the following Web site:

http://www.oracle.com/technology/documentation/oracle_retail.html

Documentation should be available on this Web site within a month after a product release. Note that documentation is always available with the packaged code on the release date.

Conventions

Navigate: This is a navigate statement. It tells you how to get to the start of the procedure and ends with a screen shot of the starting point and the statement "the Window Name window opens."

> **Note:** This is a note. It is used to call out information that is important, but not necessarily part of the procedure.

```
This is a code sample
   It is used to display examples of code
```

A hyperlink appears like this.

1

Configuring the Retail Demand Forecasting Solution

Overview

Oracle Retail Demand Forecasting™ is a statistical forecasting solution that uses state-of-the-art modeling techniques to produce high quality forecasts with minimal human intervention. Forecasts produced by the Demand Forecasting system enhance the retailer's supply-chain planning, allocation, and replenishment processes, which enables a profitable and customer-oriented approach to predicting and meeting product demand.

Forecast information is often required for items at the lowest levels in a hierarchy. Problems can arise when historic sales data for these items is too sparse and too noisy to identify clear selling patterns. In such cases, generating a reliable forecast requires aggregating sales data from a low level up to a higher level in the hierarchy. After a forecast is generated at the higher level, the resulting data can be allocated (spread) back down to the lower level. This is based on the lower level's relationship to the total. Before you can spread forecast data back down to a lower level, you should have an understanding of the relationship between the lower level and the higher level dimensions. Frequently, an additional forecast will be generated at the low level to help determine this relationship. This low level is called the final forecast level. Forecast data at this level might be sufficient to generate reliable percentage-to-whole information, but the actual forecast numbers will be more robust when they are generated at an aggregate level. This aggregate level from which forecast data is spread is referred to as the source forecast level.

Some high-volume items may possess sufficient sales data for robust forecast calculations directly at the final forecast level. In these cases, forecast data that is generated at an aggregate level and then spread down to lower levels can be compared to forecasts that are run directly at the low level. Comparing the two forecasts, each generated at a different hierarchy level, can be an invaluable forecast performance evaluation tool.

The RDF solution may include multiple final forecast levels. Forecast data must appear at some final level for the data to be approved and exported to other systems.

Using the RDF Plug-In, final and source forecast levels are defined for the RDF solution.

Note: The ability to configure the RDF solution may be limited. This is based on your licensing agreement.

Forecasting Calendar Hierarchy Requirement

With any RDF solution, configuration of the calendar hierarchy must always include a "day" dimension level name. There are no configuration requirements for the dimensions of the merchandise or location hierarchies.

Forecasting Limitations Using the Partition Hierarchy

Any dimension along the partition hierarchy that is used as an intersection to forecast must be unique across all domains. This requirement especially applies to Alternate Hierarchies. For example, if the forecast level is supplier\str\week, my Supplier dimension cannot have a supplier position that exists in multiple domains. However, additional support for clean partitioning of Alternate Hierarchies is provided through the RDF Transformation programs used to integrate RMS foundation data for RDF. See the *RPAS Administration Guide* for more information on data integration programs.

Forecasting Pre-Configuration Data Requirements

There are several parameters within the RDF configuration that may reference other measures that are configured external to the solution, specifically:

- Source Data
- Plan Data
- Spreading Profile
- Seasonal Profile

Prior to configuring an RDF solution, it is required that these measures already exist within the Project.

Source Data

The RDF plug-in populates a pick-list with all non-Boolean and non-string measures that have been created in the Project.

Spreading Profiles and Seasonal Profiles

If Curve will be used to produce “Spreading Profiles” or “Seasonal Profiles” to support your Forecasting solution, these profiles should already have been configured in the Curve solution. If these profiles are being defined external to Curve, these measures should already exist within the Project.

Plan Data

If the “Plan Data” that will be used to support Bayesian forecasting is being defined within another solution, this measure should already exist. The entry of this parameter is not required within the configuration, and it can be entered in the resulting domain(s).

Registering the RdfFunctions Library

Prior to configuring the RDF Solution, register the "RdfFunctions" library to support proper validation of the RDF-specific rules:

Open the Function Library Manager and add "RdfFunctions."

Note: If Promote is implemented, the following rules will display as invalid however these should be ignored:

Rule: PREF_PlHolder

RuleGroup: PREF_place

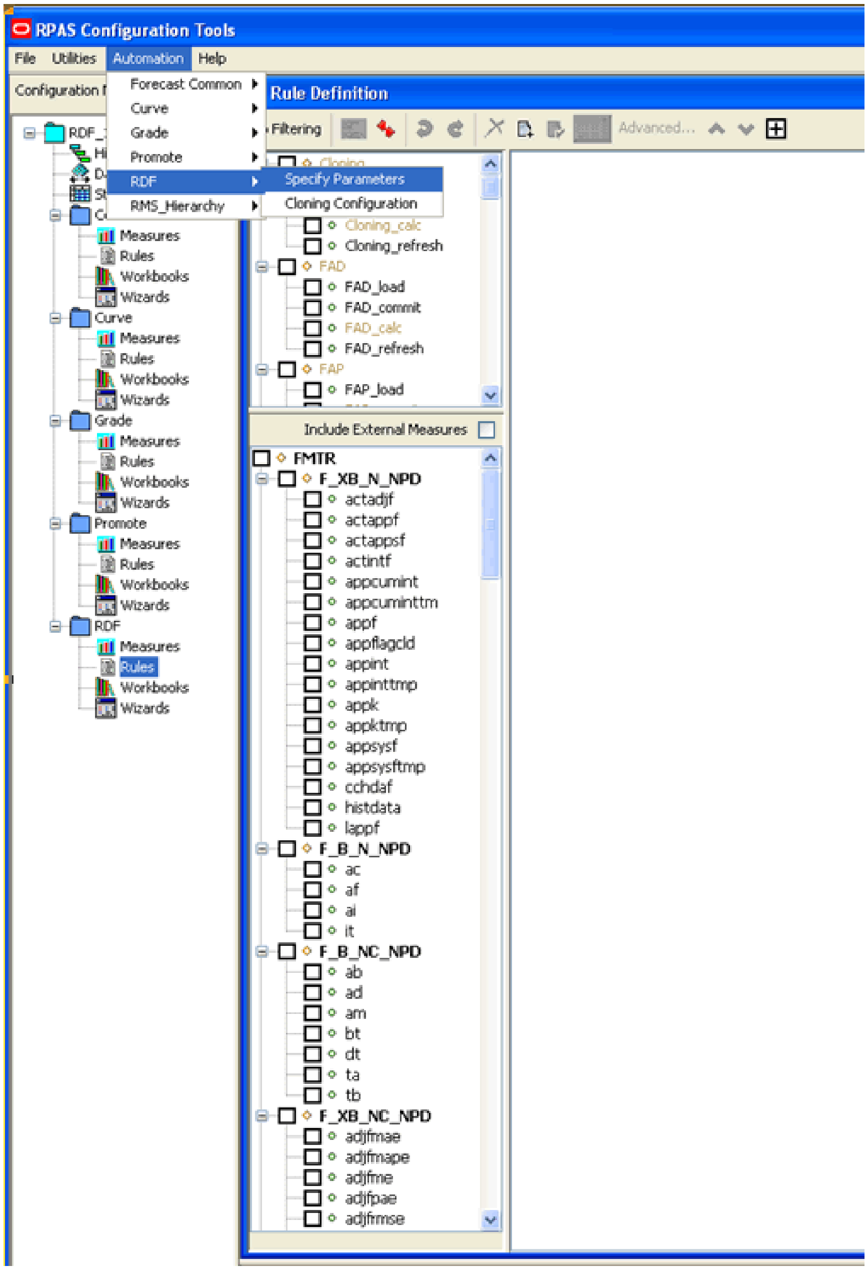
Rule Group: PRMA_place

Rule Group: PRPL_place

Creating a RDF Solution Extension

To create an RDF solution extension:

1. Open an existing configuration **in which the Curve solution has already been defined**.
2. From the Configuration Tools toolbar, select the **Automation** menu. From the **RDF** option, select **Specify Parameters**. The following sections outline the process for configuring forecast levels.



Configuration Tools Menu Options

Configuring a Final Forecast Level

To create a final level:

1. On the Forecasting Parameters utility, click the **F** icon.

   A new final level is added, and it is assigned the next available level number.
2. Specify the properties for the final level. See Editing Forecast Level Parameters for details.

Configuring a Source Forecast Level

To create a source level:

1. On the Forecasting Parameters utility, highlight the final level number in which the new source level will be associated from the Level window.
2. Click the **S** icon.

   A new source level is added, and it is assigned the next available number.
3. Specify the properties for the source level. See Editing Forecast Level Parameters for details.

> **Note:** New source of final forecast levels cannot be added or patched into an existing domain.

Editing Forecast Level Parameters

Edit forecast parameters:

- **Level Name** – The Level Name is the system-assigned level number when a forecast level is created. This is a read-only parameter.
- **Level Label** – The Level Label is the level description that will be viewed by the user once the domain is created.
  - Level Labels may not exceed forty characters.
  - It is recommended, but not required, that Level Labels include the Level Name (the system-assigned level number). Within the Forecast Administration workbook, the Default Source Level may be edited. This pick-list is populated with the Level Name for all levels that are associated with a final level. Since this value can also be specified within this configuration, this recommendation may not be necessary if changes to the Default Source Level are not expected within the application.
  - RPAS will automatically put () around Forecast Level Labels. The configuration specialist should not include these in their level label configuration or the installer will fail. An example of a Forecast Level Label that would violate this requirement is (1:itm/str/week - Final). This example is acceptable as:1-item/str/week - Final.
  - '-' should not be used before or after the Forecast Level Label. An example of a Forecast Level Label that would violate this requirement is: -1:itm/str/week – Final-. This example is acceptable as: 1-itm/str/week – Final
  - ‘:’ should not be used at all in the Level Label. An example of a Level Label that would violate this requirement is 1**:** itm/str/week-

- **Intersection** – The Intersection is the hierarchy dimensions that define the forecasting level.
- **Default Source Level** – Assigned only at the Final level, the Default Source Level is the primary level at which the aggregate, more robust forecast is run. The desired Source Level must first be created within the RDF configuration for it to be a selection in the pick-list. For more information on Source Level Forecasting, refer to the *RDF User Guide*.

  If no source level is required, the final level should be selected.
- **Source Data** – Assigned only at the Final level, the Source Data is the measure to be used as the input data (for example, POS) for the generation of forecasts. The values in this pick-list are populated with all non-string and non-Boolean type measures that are configured in the Project.
- **Periodicity** – Periodicity is the number of periods within the Calendar dimension, which are defined in the forecast level intersection. For example, if an intersection is defined at Week/Item/Store the Periodicity value will be 52 (since there are 52 weeks within a year).
- **Forecast Method** – The Forecast Method window displays all forecast generation methods that may be defined for a forecast level. The Default Forecast Method is also determined here.

  > **Note:** The "Causal" method should be selected as a valid method ONLY for levels in which causal forecasting will be used.

  The following is a list of Forecast Methods that may be selected. See the *RDF User Guide* for more information on each method.

  - No Forecast
  - Average
  - Simple
  - Intermittent
  - Simple/Intermittent
  - Trend
  - Additive Seasonal
  - Multiplicative Seasonal
  - Seasonal
  - AutoES
  - Causal

    > **Note:** This method should only be selected as a valid method for levels that will use Causal Forecasting. If **Causal** is selected and Promote is not licensed or configured, the RDF batch forecast will not generate.

> When enabling Causal as a valid forecast method for a source level, note that RDF Promotion variables need to be provided at the same dimension along the Product and Location hierarchies as the forecast level for which Causal forecasting is executed (Final or Source). RDF Causal does not support aggregation of promotion variables along any hierarchies other than Clnd. Aggregation of promotion variables along product and/or location hierarchies needs to be handled externally through configuration. Aggregation along the calendar hierarchy is support by RDF Causal, using specified aggregation and spread profiles. Please refer to the *RDF User Guide* for details.

  - Bayesian
  - Profile-based
  - LoadPlan
  - Copy
- **Plan Data** – Assigned only at the final level, Plan Data (sales plans) provide details of the anticipated shape and scale of an item's selling pattern. This information is required when Bayesian forecasting is used as a Forecast Method. The value in this parameter is a measure name.
- **Seasonal Profile** – A seasonal profile provides details of the anticipated seasonality of an item's selling pattern. The seasonal profile is required in conjunction with the Profile-based Forecast Method. The seasonal profile can be generated or loaded, depending on your configuration. The value in this parameter is a measure name.
- **Spreading Profile** – Assigned only at the source forecasting level, the Spreading Profile is used to spread source level forecasts down to the final forecast level. The value in this parameter is a measure name, a profile level name, or any combination of these separated by commas.
  - If Curve is used to dynamically generate the spreading ratios, this parameter should be populated with the final profile level name (profile number) configured. For example: 01 (this is profile level 01).
  - If Curve is used to generate the static (manually approved) spreading ratios, this parameter should be populated with the Approved Profile measure. For example: apvp11 (this is the Approved Profile for Curve level 11).

> **Note:** For more information on Source Level Forecasting, see the *RDF User Guide*.

Autogenerating Hierarchies, Measures, Rules and Workbook Templates

The following is the process to autogenerate the hierarchies, measures, rules, and workbook templates that are required by RDF to support the forecasting configuration entered in the RDF plug-in:

On the Forecasting Parameters utility, click the **OK** icon.

The system automatically generates the following:

- **Hierarchies** – The DATA hierarchy will be updated with the flvl, fbrt and fmtr dimensions.
- **Measures** – All measures necessary to support the base RDF solution will be created.
- **Rules** – All Rule Sets, Rule Groups, and Rules to support the base RDF solution will be created.
- **Workbook Templates** – All pre-defined workbook templates to support the base RDF solution will be created.

You may continue to make changes to the RDF plug-in configuration, and the autogeneration process may be repeated as often as needed prior to the installation.

Deleting a Forecast Level

Deleting a forecast level will cause the system-assigned enumerated values in the Level Name to renumber such that levels are in consecutive order starting with forecast level 01. Deleting a forecast level may impact any solution configuration that uses a specific level.

If the domain using the configuration has previously been installed, there is potential to lose data associated to a level that has been deleted or renumbered.

To delete a level:

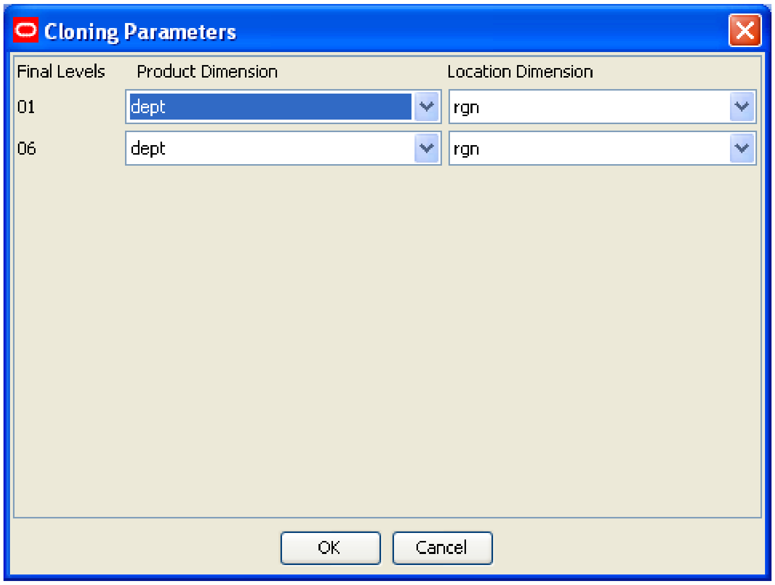
1. On the Forecasting Parameters utility, highlight the number of the level that you want to delete from the Level window.
2. Click the **X** icon.

   The level is deleted. If you delete a final level, any source levels that are associated with it will also be deleted.
3. Select the **OK** icon to regenerate the solution with the changes to the cluster configuration.

Configuring the Cloning Administration Workbook

Product/Location Cloning Administration Workbook allows users to specify Clone Products by a configurable dimension in the Location hierarchy and Clone Stores by a configurable dimension in the Product hierarchy. For example, users can specify a different Clone SKU for a different Region.

These dimensions can be specified from the **Cloning Configuration** menu option under **RDF Automation**. When the user clicks on this menu, the Cloning Parameters dialog box appears.



Final Levels	Product Dimension	Location Dimension
01	dept	rgn
06	dept	rgn

Cloning Parameters Dialog Box

For each Final forecast level, the user is prompted to select a Product Dimension and a Location dimension. The values selected here drive the dimensionality of the Product Cloning and Location Cloning Worksheets in the Cloning Administration Workbook. Note that the” Product dimension” selected here actually drives the Location Cloning Worksheet and the “Location Dimension” drives the Product Cloning Worksheet. For example, the “Product dimension” is the dimension by which Clone Users want to specify Location clones and vice versa.

For example, if final level 01 is at SKU/Store/Week and the user has chosen Dept for Product Dimension and Region for Location Dimension, then for Final level 01 the Product Cloning Worksheet will be generated at SKU/Region and the Location Cloning Worksheet will be at Store/Dept.

Note that if the Cloning configuration menu option is not invoked, then the Cloning Administration Workbook and associated measures will not be generated in the configuration.

Editing the RDF GA Configuration

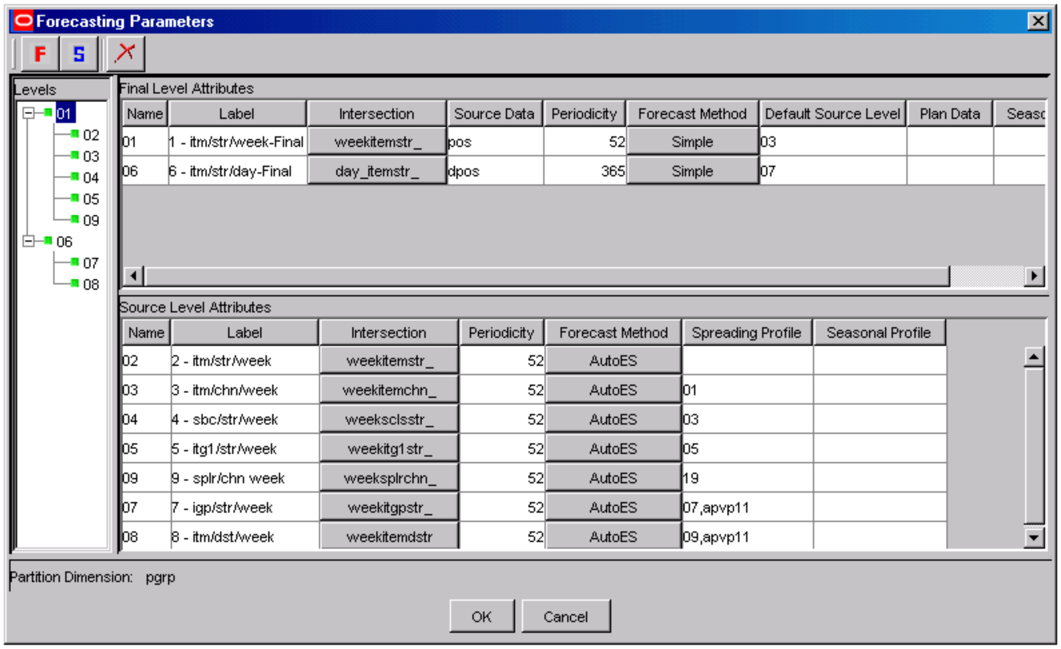
The autogeneration process creates hierarchies, measures, rules, and workbook templates that are required to support the essential RDF functionality. This base configuration is referred to as the “GA Configuration.” Certain changes to the GA Configuration are allowed. Once edits to the GA Configuration are made and the autogeneration process occurs again, valid changes to the configuration will be preserved. There is nothing in the RPAS Configuration Tools to prevent invalid changes from being made.

The following outlines acceptable changes and restrictions:

- **“RDF” Solution Extension Name** – The name assigned to the resulting RDF solution after autogeneration occurs **cannot** be edited.
- **Major and Minor Components** – Additional Major and Minor components may be added to the RDF GA Configuration. The Major and Minor components that are part of the GA Configuration may not be edited. This restriction also applies to Measure Names and Measure Labels.
- **Rules –** Additional Rule Sets, Rule Groups, and Rules may be added to the RDF GA Configuration. This includes support for adding new Rules to existing GA Configuration Rule Groups. It is recommended that new Rules added to the GA Configuration Rule Groups include “cust” (represents “Custom”) in the Rule Name. This allows for easy identification of Rules that are not part of the GA Configuration. Rule Sets, Rule Groups, and Rules that are part of the GA Configuration may not be renamed. Existing Rules that are part of the GA Configuration may not be modified in any way.
- **Workbook Templates** – Additional Workbook Templates may be added to the RDF GA Configuration. New Measures and Rules may also be added to the GA Configuration Workbook Templates. This is done by adding new Major and Minor components, and adding new Rules to existing Rule Groups in the GA Configuration.

RDF Example

An example of the Forecasting Parameters utility appears below.



Final Level Attributes

Name	Label	Intersection	Source Data	Periodicity	Forecast Method	Default Source Level	Plan Data	Seasc
01	1 - itm/str/week-Final	weekitemstr_	pos	52	Simple	03		
06	6 - itm/str/day-Final	day_itemstr_	dpos	365	Simple	07		

Source Level Attributes

Name	Label	Intersection	Periodicity	Forecast Method	Spreading Profile	Seasonal Profile
02	2 - itm/str/week	weekitemstr_	52	AutoES		
03	3 - itm/chn/week	weekitemchn_	52	AutoES	01	
04	4 - sbc/str/week	weeksclsstr_	52	AutoES	03	
05	5 - itg1/str/week	weekitg1str_	52	AutoES	05	
09	9 - splr/chn week	weeksplrchn_	52	AutoES	19	
07	7 - igp/str/week	weekitgpstr_	52	AutoES	07,apvp11	
08	8 - itm/dst/week	weekitemdstr	52	AutoES	09,apvp11	

Partition Dimension: pgrp

Forecasting Parameters Window

2

Configuring the Promote Solution

Overview

Promote (Promotional Forecasting) is an optional add-on solution to RDF that allows for the effects of promotional and causal events, such as radio advertisements and holiday occurrences, into time series forecasts. The promotional forecasting process uses past sales data and promotional information to forecast future demand.

Using the Promote Plug-In, promotions are defined that will be used within the Promote Solution.

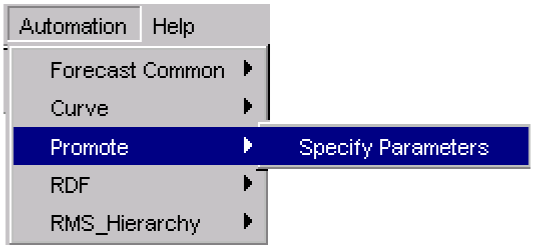
Creating a Promote Solution Extension

To create the Promote solution extension:

1. Open an existing configuration in which **the Curve and RDF solution** have already been defined.

   > **Note:** Promotion/causal forecasting levels are determined within the RDF Solution by selecting **Causal** as a valid Forecasting Method for source or final forecasting levels.

2. From the Configuration Tools toolbar, select the **Automation** menu. From the Promote option, select **Specify Parameters**. The following sections outline the process for configuring forecast levels.



Creating a Promotion

To create a Promotion:

1. On the Promote Parameters utility, click the **P** icon.

   A new promotion is added, and it is assigned a default promotion number for the Promotion Name (ex. P001).

2. Specify the properties for the promotion. See Editing Promote Parameters for details.

Editing Promote Parameters

Edit promotion parameters:

- **Default Intersection** – The Default Intersection is the intersection at which any new promotion will be defined. Editing the Default Intersection will not affect any existing promotions.
- **Promotion Name** – The Promotion Name is the internal system identifier of the promotion. The system will initially assign a generic Promotion Name (P001), but this value may be overwritten. The Promotion Name may not be greater than four characters. The following characters may not precede or follow the name that is entered in this field:
  - ‘( )’ Example: (xmas)
  - ‘-‘ Example: -xmas-

    The following may not be use at all in the Promotion Name:
  - ‘:’ Example: xmas:
- **Promotion Label** – The Promotion Label is the description of the promotion that will be viewed by the user once the domain is created.
  - Promotion Labels may not exceed forty characters.

    The following characters may not precede or follow the label that is entered in this field:
    - ‘( )’ Example: (xmas)
    - ‘-‘ Example: -xmas-

    The following may not be use at all in the Promotion Name:
    - ‘:’ Example: xmas:
- **Promotion Intersection** – Independent of the causal forecasting levels, the Promotion Intersection is the hierarchy dimensions that define the promotion. This will be pre-populated with the value set in the ‘Default Intersection’ at the time when the promotion is created.
- **Type** – The Type is the data type of the promotion variable. Promotion Variables may be defined as Boolean or Real types. The value in this parameter defaults to Boolean.
- **Database** – The Database displays the database that will be used to store promotion variable information. The value in this parameter defaults to the data/promo database.

Autogenerating Hierarchies, Measures, Rules and Workbook Templates

The following is the process to autogenerate the hierarchies, measures, rules, and workbook templates required by Promote to support the promotion configuration entered in the Promote plug-in:

- On the Promote Parameters utility, click the **OK** icon.

The system automatically generates:

- **Hierarchies** – The DATA hierarchy will be updated with the ptyp and prom dimensions.
- **Measures** – All measures necessary to support the base Promote solution will be created.
- **Rules** – Only the rules and rule groups necessary to support the installation of the Promote solution are visible in the configuration. Unique to Promote, the additional rules and rule groups needed to support the Promote workbook templates and batch forecast are generated within the domain and not within the Plug-in.
- **Workbook Templates** – All pre-defined workbook templates to support the base Promote solution will be created; however, the worksheets are not visible. Unique to Promote, the additional workbook templates needed to support the Promote solution are generated within the domain and not within the Plug-in. You may continue to make changes to the Promote plug-in configuration, and the autogeneration process may be repeated as often as needed prior to the installation.

> **Note:** After autogeneration completes, the following rules will display as invalid however these should be ignored:
>
> Rule: PREF_PlHolder
>
> RuleGroup: PREF_place
>
> Rule Group: PRMA_place
>
> Rule Group: PRPL_place

Deleting a Promotion

Deleting a promotion may impact any solution configuration that references the deleted promotion.

To delete a promotion:

1. On the Promote Parameters utility, highlight the promotion to delete from the configuration.
2. Click the **X** icon.

   The promotion is deleted.
3. Select the **OK** icon to regenerate the solution with the changes to the cluster configuration.

> **Note:** See the *RDF Administration Guide* for more information on patchable changes to the configuration.

Editing the Promote GA Configuration

The Promote autogeneration process creates all hierarchy dimensions and measures to support the essential Promote functionality; however, only the rules and workbook templates required to support the domain installation are visible in the configuration. Unique to Promote, the additional rules, rule groups and workbook templates needed to support the Promote solution and batch forecast are generated within the domain and not within the Plug-in.

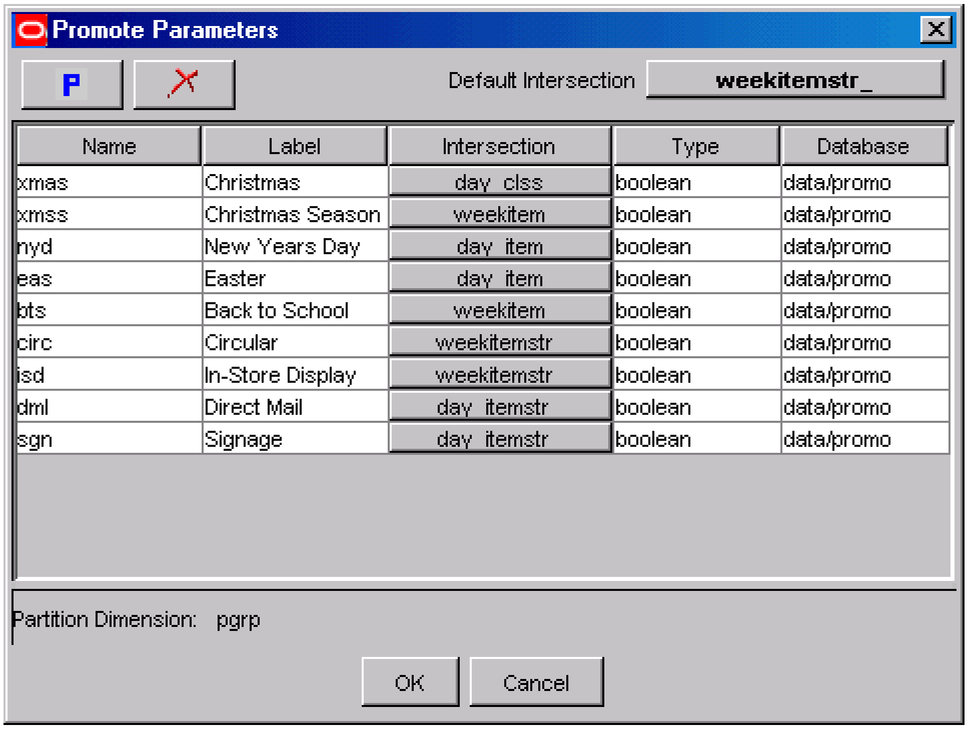
> **Note:** This limitation allows for fewer options than in RDF and Curve for edits to the GA Configuration.

The following outlines acceptable changes and restrictions:

- **"Promote" Solution Extension Name** – The name assigned to the resulting Promote solution after autogeneration occurs **cannot** be edited.
- **Major and Minor Components –** Additional Major and Minor components may be added to the Promote GA Configuration. The Major and Minor components that are part of the GA Configuration may not be edited. This restriction also applies to Measure Names and Measure Labels.
- **Rules –** Additional Rule Sets, Rule Groups, and Rules may be added to the Promote GA Configuration. This includes support for adding new Rules to existing GA Configuration Rule Groups. It is recommended that new Rules added to the GA Configuration Rule Groups include 'cust' (represents 'Custom') in the Rule Name. This allows for easy identification of Rules that are not part of the GA Configuration. Rule Sets, Rule Groups, and Rules that are part of the GA Configuration may not be renamed. Existing Rules that are part of the GA Configuration may not be modified in any way.
- **Workbook Templates** – Additional Workbook Templates may be added to the Promote GA Configuration; however, new Measures and Rules CANNOT be added to the GA Configuration Workbook Templates because the Promote worksheets are not visible in the configuration.

Promotion Example

An example of the Promote Parameters utility appears below.



Name	Label	Intersection	Type	Database
xmas	Christmas	day clss	boolean	data/promo
xmss	Christmas Season	weekitem	boolean	data/promo
nyd	New Years Day	day item	boolean	data/promo
eas	Easter	day item	boolean	data/promo
bts	Back to School	weekitem	boolean	data/promo
circ	Circular	weekitemstr	boolean	data/promo
isd	In-Store Display	weekitemstr	boolean	data/promo
dml	Direct Mail	day itemstr	boolean	data/promo
sgn	Signage	day itemstr	boolean	data/promo

Partition Dimension: pgrp

Promote Parameters Window

A

Appendix: Configuring the Clone Procedure

Cloning allows users to generate forecasts for new items and locations by copying, or cloning history, from other SKUs and stores. Users can map SKUs or stores that have similar business cases, clone the historical data, and begin generating forecasts. Cloning provides the ability to generate forecasts based on historical data and promotional calendar.

The “Clone Syntax” section contains the specifications and syntax for configuring the Forecast procedure.

The Clone procedure can be set up to clone sales history, promotion history. The Clone procedure can handle cloning of real, integer, Boolean and string measures.

> **Note:** The syntax is slightly different than the standard RPAS functions and procedures that are described in the “Rule Functions Reference Guide” section of the *RPAS Configuration Tools User Guide*.

Clone Requirements

The following libraries must be registered in any domain(s) that will use the Clone solution extension:

- RdfFunctions

Using the Clone Procedure

The following notes provide information about Clone functionality.

- Refer to the appropriate input parameters and output measures when using the Clone procedure.
- The CLND, PROD, and LOC hierarchies are required by the Clone expression. CLND must be the inner-most hierarchy.
- Cloning supports up to 3 parent SKUs or 3 parent stores with contribution percentages for each of these SKUs or stores.
- An adjustment ratio can be defined to modify the level of the cloned history for the new product or location.
- Users can specify different parent SKUs (or Stores) for different locations. For example, SKU A sells like SKU B in Region 1 and like SKU C in Region 2. These location or product levels can be configured through the RPAS Configuration Tools.
- The cloning (copying) of historical data is performed as part of batch process using the Clone special expression.
- The clone special expression is generic enough that it can be used to copy not just history, but forecast parameters, Casual histories and more, using the clone special expression.

- A mask measure is used to define when cloning is performed. When the mask measure is TRUE, cloning is performed; setting the mask to FALSE stops the cloning process. A business rule may be defined (using RPAS rules) to set the mask measure to FALSE when it is desired to stop cloning the item/location. Note that if no mask measure is specified, cloning is performed for all item/locations that have a like item and/or like location specified.
- The input parameters include: A source array, up to 6 map measures, 2 contribution measures, an optional mask measure, and optional Adjustment Ratio Measure, and a destination array.
- The source and destination array will have to be at the same intersection (This will be validated by the special expression) and this will be intersection at which cloning is performed.
- The map measure, contribution measure and the mask measures can be at higher intersections to the source and destination arrays.
  - In the event that any of these are at a higher intersection, standard replication will be used for spreading values down to the source and destination Arrays.
- The map arrays and contribution measures are optional, with at least one map measure and contribution measure being necessary. The number of contribution measures should be equal to the number of map measures.
- Two additional optional start and end date measures can be passed, which specify the start end and end date indexes of the cloning process.
  - Similar restrictions for the intersection of the Start and End Date Index measures apply as mentioned with map measures above.
  - The index should be an index along a calendar dimension equal to the Calendar Dimension along Source and Destination array. For example, if the Source and Destination Arrays are at the SKU/Store/Week level intersection, then the Start and end date index measures should contain Index values of the Week dimension.
  - If these values are not passed, they will default to start and end dates of calendar hierarchy.

About Cloning

Cloning is handled a little differently depending on the type of measures being clones. This topic address the various manner in which cloning is handled for the following measure types:

- Real or integer measure
- Boolean measures
- String measures

Cloning Real or Integer Measures

Since up to 3 parent SKUs and 3 parent Stores can be specified, the number of mapping measures can consist of up to 9 combinations of SKU/Stores (# of parent SKUs specified x # pf parent Stores specified) and corresponding % contributions.

Example:

The following example illustrates how real or integer values are calculated using the Clone expression.

SKU1

SKU2	SKU3	SKU4
20%	20%	60%

STR1

STR2	STR3	STR4
20%	50%	30%

The Special Expression will calculate mappings and contributions as follows:

- SKU2/STR2 at 4%
- SKU2/STR3 at 10%
- SKU2/STR4 at 6%
- SKU3/STR2 at 4%
- SKU3/STR3 at 10%
- SKU3/Str4 at 6%
- SKU4/STR2 at 12%
- SKU4/STR3 at 30%
- SKU4/Str4 at 18%

Example 2 – Subset of possible values provided:

If only a subset of values is populated, then the Clone expression performs its calculations as illustrated below.

SKU1

SKU2	SKU3	
20%	80%	

STR1

STR2	STR3	
50%	50%	

The Special Expression will calculate mappings and contributions as follows:

- SKU2/STR2 at 10%
- SKU2/STR3 at 10%
- SKU3/STR2 at 40%
- SKU3/STR3 at 40%

Example 2 – Subset of possible values provided with Adjustment Ratio:

If only a subset of values is populated and an Adjustment Ratio is defined, then the Clone expression performs its calculations as illustrated below.

SKU1

SKU2	SKU3	
20%	80%	

Adjustment Ration = 1

STR1

STR2	STR3	
50%	50%	

Adjustment Ration = 0.5

The Special Expression will calculate mappings and contributions as follows:

- SKU2/STR2 at 5% (=20% x 0.5x50%)
- SKU2/STR3 at 5%
- SKU3/STR2 at 20% (=80%x0.5x50%)
- SKU3/STR3 at 20%

Cloning Boolean Measures

It is possible to clone promotion variables, which could be Boolean measures. This topic provides information about how the cloning of Boolean measures, specifically multiple Boolean measures, is handled.

The special expression supports the use of multiple Like SKUs or Like Stores for cloning Boolean measures. Users need to specify a method of combining the multiple measures, which could be an AND or an OR.

Example:

The following example illustrates how Boolean measures values are handled using the Clone expression.

SKU1

SKU2	SKU3	SKU4

STR1

STR2	STR3	STR4

The Special Expression will calculate SKU1/STR1 as follows:

SKU2/STR2 OR SKU2/STR3 OR SKU2/STR4 OR SKU3/STR2 OR SKU3/STR3 OR SKU3/STR4 OR SKU4/STR2 OR SKU4/STR3 OR SKU4/STR4

Example – Cloning Promotion Measures for New SKU with OR Aggregation type defined:

When only cloning promotion measures for new SKUs and user has specified an OR aggregation type:

SKU1

SKU2	SKU3	

For SKU1/STR1, the special expression calculates the following:

SKU2/STR1 OR SKU3/STR1

Cloning and String Measures

It is possible to clone multiple string measures. When more than one clone SKUs or stores are specified, the special expression concatenates individual string measures.

When cloning Boolean or String measures, the following parameters are **ignored** by the special expression:

- SKUCONTRIBUTION1
- SKUCONTRIBUTION2
- SKUCONTRIBUTION3
- SKUADJUSTMENTRATIOMEAS
- STRCONTRIBUTION1
- STRCONTRIBUTION2
- STRCONTRIBUTION3
- STRADJUSTMENTRATIOMEAS

Valid Parameters during Cloning

When cloning Boolean or String measures, the following parameters are **used in the same way as with real or integer measures:**

- SOURCE_MEAS
- SKUMAPMEAS1
- SKUMAPMEAS2
- SKUMAPMEAS3
- STRMAPMEAS1
- STRMAPMEAS2
- STRMAPMEAS3
- STARTDATEINDEX
- ENDDATEINDEX
- MASKMEASURE

When cloning Boolean measures, the following **additional** parameters can be specified:

- Boolean Operator (indicating whether to use an AND or an OR operator for combining multiple measures). If this measure is not specified, and when cloning Boolean measures, the special expression defaults to an OR operation for combining multiple measures.
  - Boolean operator parameter can be specified as a scalar constant, a scalar measure, or a non-scalar measure which has a base intersection that is equal to or higher than the dest/src measure intersection.
  - The Boolean operator when specified as a measure, it needs to be a real measure, so that it can be displayed as a picklist if desired.

Syntax Conventions

The table below displays the syntax conventions used in this document.

Indicator	Definition
[...]	All options listed in brackets are optional.
{...|...}	Options listed in "{}" with " | " separators are mutually exclusive (either/or).
{...,...}	Options listed in "{}" with "," separators way are a complete set.
Bold	Labels.
Italics	Italics indicate a temporary placeholder for a constant or a measure.
Italics/meas	This indicates that the placeholder can be either a constant or a measure.
BoldItailics	This indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N.
Normal	Normal text signifies required information.
Underlined	This convention is used to identify the function name.

Clone Syntax

The syntax for using the Clone procedure appears below. The input and output parameter tables explain the specific usage of the parameters names use in the procedure.

Generic Example for Cloning Real or Integer Measures:

```
DEST_MEASURE <-clone(SRC:SOURCE_MEASURE,
SKUMAP1: SKUMAPMEAS1, SKURATIO: SKUCONTRIBUTION1, SKUMAP2: SKUMAPMEAS2,
SKURATIO2:SKUCONTRIBUTION2, SKUMAP3:SKUMAPMEAS3, SKURATIO3: SKUCONTRIBUTION3,
SKUADJRATIO: SKUADJUSTMENTRATIOMEAS
STRMAP1:STRMAPMEAS1, STRRATIO1:STRCONTRIBUTION1, STRMAP2:STRMAPMEAS2,
STRRATIO2:STRCONTRIBUTION2, STRMAP3:STRMAPMEAS3, STRRATIO3:STRCONTRIBUTION3,
STRADJRATIO:STRADJUSTMENTRATIOMEAS [, STARTINDEX:STARTDATEINDEX]
[, ENDINDEX:ENDDATEINDEX] [, MASK:MASKMEASURE] [,BOOLOPT:BOOLEANOPERATOR])
```

Generic Example for Cloning Boolean Measures:

```
DEST_MEASURE <-clone(SOURCE_MEASURE,
SKUMAPMEAS1, SKUMAPMEAS2, SKUMAPMEAS3,
STRMAPMEAS1, STRMAPMEAS2, STRMAPMEAS3,
STARTDATEINDEX, ENDDATEINDEX, , MASKMEASURE, BOOLEANOPERATOR)
```

Generic Example for Cloning String Measures:

```
DEST_MEASURE <-clone(SOURCE_MEASURE,
SKUMAPMEAS1, SKUMAPMEAS2, SKUMAPMEAS3,
STRMAPMEAS1, STRMAPMEAS2, STRMAPMEAS3,
STARTDATEINDEX, ENDDATEINDEX, MASKMEASURE)
```

Sample of Clone Function with Real or Integer Measures:

```
DEST:clnSls <- Clone(SRC: promo, SKUMAP1:prodcln101xb, SKURATIO1:prodcnt101xb,
SKUMAP2:prodcln201xb, SKURATIO2:prodcnt201xb, SKUMAP3:prodcln301xb,
SKURATIO3:prodcnt301xb, STRMAP1:loccln101xb, STRRATIO1:loccnt101xb,
STRMAP2:loccln201xb, STRRATIO2:loccnt201xb, STRMAP3:loccln301xb,
STRRATIO3:loccnt301xb, SKUADJRATIO:prodadjpct01xb, STRADJRATIO:locadjpct01xb,
MASK:clnmask01xb)
```

Configuration Parameters and Rules

Input Parameters

The table below provides the input parameters for the Clone procedure and special expressions.

Parameter Name	Description
SOURCE_MEASURE	The source measure used for cloning. The source array and destination array need to be at the same intersection, which will be validated by the special expression. This is the intersection in which cloning is performed. Data Type: Integer, Real, Boolean, or String Required: Yes
SKUMAPMEAS1	The first SKU measure that is mapped to the new SKU/Store. Data Type: String Required: Yes At least one map measure is required when using the clone function. SKUMAPMEAS1 needs to be populated with position ID of clone Item.
SKUCONTRIBUTION1	The percentage of data used when cloning historical data. Used to weigh or assign importance of SKU or Store map. Data Type: Real Required: Yes – for Real or Integer cloning. Ignored for Boolean and String cloning. At least one contribution measure is required Real or Integer cloning.
SKUMAPMEAS2	The second SKU measure that is mapped to the new SKU/Store. Data Type: String Required: Yes
SKUCONTRIBUTION2	The percentage of data used when cloning historical data for SKUMAPMEAS2. Used to weigh or assign importance of SKU or Store map. Data Type: Real Required: Yes Ignored for Boolean and String cloning.
SKUMAPMEAS3	The third SKU measure that is mapped to the new SKU/Store. Data Type: String Required: Yes
SKUCONTRIBUTION3	The percentage of data used when cloning historical data for SKUMAPMEAS3. Used to weigh or assign importance of SKU or Store map. Data Type: Real Required: Yes Ignored for Boolean and String cloning.

Parameter Name	Description
SKUADJUSTMENTRATIOMEAS	A value greater than zero (0) used to adjust the level of history calculated at the SKU level. Data Type: Real Required: Yes
STRMAPMEAS1	The first STR measure that is mapped to the new SKU/Store. Data Type: String Required: Yes STRMAPMEAS1 needs to be populated with position ID of clone store.
STRCONTRIBUTION1	The percentage of data used when cloning historical data for STRMAPMEAS1. Used to weigh or assign importance of SKU or Store map. Data Type: Real Required: Yes – for Real or Integer cloning. Ignored for Boolean and String cloning.
StrMapMeas2	The second STR measure that is mapped to the new SKU/Store. Data Type: String Required: Yes
StrContribution2	The percentage of data used when cloning historical data for STRMAPMEAS2. Used to weigh or assign importance of SKU or Store map. Data Type: Real Required: Yes Ignored for Boolean and String cloning.
StrMapMeas3	The third STR measure that is mapped to the new SKU/Store. Data Type: String Required: Yes
StrContribution3	The percentage of data used when cloning historical data for STRMAPMEAS3. Used to weigh or assign importance of SKU or Store map. Data Type: Real Required: Yes Ignored for Boolean and String cloning.
STRADJUSTMENTRATIOMEAS	A value used to adjust the level of history calculated at the store level. Data Type: Real Required: Yes
STARTDATEINDEX	The index value for the Week dimension to be used as the starting point to clone data. Data Type: Integer Required: No

Parameter Name	Description
ENDDATEINDEX	The index value for the Week dimension to be used as the ending point to clone data. Data Type: Integer Required: No
MASKMEASURE	A TRUE or FALSE value that indicates if cloning should be performed. The special expression only updates the DEST_MEASURE for SKU/Stores with MASKMEASURE set to TRUE, if this measure is passed. If this measure is not specified, it updates it for all SKU/Stores. Data Type: Boolean Required: No
BOOLEANOPERATOR	An optional parameter that can be specified when cloning Boolean measures. This parameter indicate whether to use an AND or an OR operator for combining multiple measures Data Type: Boolean Required: No

Output Parameters

The table below provides the output parameters for the Clone procedures.

Parameter Name	Description
DEST_MEASURE	An array containing the item or location data is being cloned. Data Type: Integer, Real, Boolean, or String Required: Yes Data type and Base Intersection of this measure need to be the same as that of the SOURCE_MEASURE.

B

Appendix: Configuring the Forecast Procedure

Using the RPAS Configuration Tools, a time-series demand forecast may be configured as part of a planning workflow or business process. The Forecast procedure provides only a small subset of the functionality that is available through RDF. The differences between these solution extensions are as follows:

- The forecast produced by the Forecast procedure is a single-level forecast.
- RDF allows for forecasts to be generated at aggregate levels in the data (to remove sparsity), and then this forecast is spread down to the execution level by using a profile.
- The Forecast procedure allows for a single forecasting method to be specified in the calculation of the forecast.
- RDF allows for forecasting methods and forecasting parameters to be modified as needed at all levels in your data.
- No standard approval process of the resulting forecasts are included as part of the Forecast procedure.
- RDF allows for forecast adjustments and approvals to be made at the lowest level necessary in your data.

The "Forecast Procedure Syntax" section contains the specifications and syntax for configuring the Forecast procedure.

Note: The syntax is slightly different than the standard RPAS functions and procedures that are described in the "Rule Functions Reference Guide" section of the *RPAS Configuration Tools User Guide*.

Forecast Requirements

The following libraries must be registered in any domain(s) that will use the Forecast solution extension:

- AppFunctions
- RdfFunctions

Forecast Parameter/Model Dependencies

The model listed below required the stated measure to be provided.

- Bayesian model – Plan measure required.
- Profile model – Profile measure required.

Using the Forecast Procedure

The following notes are intended to serve as a guide for configuring the Forecast procedure within the RPAS Configuration Tools.

- Refer to the appropriate input parameters and output measures when using the Forecast procedure.
- The resultant measure (that is, frcstout) should be at the same intersection as your history measure (that is, pos). This will be the base intersection of the final level.
- The Forecast procedure is a multi-result procedure, meaning that it can return multiple results with one procedure call within a rule. In order to get multiple results, the resultant measures must be configured in the Measure Tool and the specific measure label must be used on the left-hand-side (LHS) of the procedure call. The resultant measure parameters must be comma-separated in the procedural call.
- The startdatemeas that specifies the forecast start date needs to be periodically updated (every week or so) by configuring rules.
- The forecast methods are specified via the mask measure. This is an int measure. Please refer to the Forecast Model/method list table, for the expected values of this measure for each forecast method.

Syntax Conventions

The table below displays the syntax conventions used in this document.

Indicator	Definition
[...]	All options listed in brackets are optional.
{...|...}	Options listed in “{}” with “ | ” separators are mutually exclusive (either/or).
{...,...}	Options listed in “{}” with “,” separators way are a complete set.
Bold	Labels.
Italics	Italics indicate a temporary placeholder for a constant or a measure.
Italics/meas	This indicates that the placeholder can be either a constant or a measure.
BoldItailics	This indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N.
Normal	Normal text signifies required information.
Underlined	This convention is used to identify the function name.

Forecast Procedure Syntax

The syntax for using the Forecast procedure appears below. The input and output parameter tables explain the specific usage of the parameters names used in the procedure.

Generic Example:

```
FORECAST: FORMEAS [, INT: INTMEAS, CUMINT:CUMINTMEAS, PEAKS:PEAKSMEAS,
CHMETHOD:METHMEAS, CHLEVEL:LVLMEAS, CHTREND:TRENDMEAS, ALERTS:ALERTSMEAS] <-
FORECAST(MASK:MEASKMEAS, {STARTDATE:STARTDATE | STARTDATEMEAS:STARTDATEMEAS},
FORECASTLENGTH:FORECASTLENGTH, HISTORY:HISTORYMEAS, PERIOD:PERIOD [
[,{PROMO_0:PROMO0, PROMOEFF_0:PROMOEFF0, PROMOOVER_0:PROMOOVER0,
PROMOTYPE_0:PROMOTYPE0} … {,PROMO_N:PROMON,
PROMOEFF_N:PROMOEFFN,PROMOOVER_N:PROMOOVERN, PROMOTYPE_N:PROMOTYPEN} ],
HISTSTART: HISTSTARTMEAS, {FRCSTSTARTMEAS:FRCSTSTARTMEAS | FRCSTSTART:FRCSTSTART},
MINWINTERS:MINWINTERSMEAS, MINHOLT: MINHOLTMEAS, MINCROSTON:MINCROSTON,
MAXALPHA:MAXALPHA, MAXWINTERSALPHA:MAXWINALPHA, MAXPROFILEALPHA:MAXPROFILEALPHA,
BAYESALPHA:BAYESALPHA, TRENDDAMP:TRENDDAMP, {VALID_DD:VALID_DD,
DDPROFILE:DDPROFILE }, PROMO_IN_BASELINE:PROMO_IN_BASELINE, PLAN:PLAN,
PROFILE:PROFILE, VERBOSE:VERBOSE, AGGPROF:AGGPROF, SPREADPROF:SPREADPROF,
READMODE:READMODE, BAYESIAN_HORIZ,BAYESIAN_HORIZ, MINB:MINB, MAXB:MAXB,
KEEPCLAMPEDMAXB:KEEPCLAMPEDMAXB, SMOOTHBASELINE:SMOOTHBASELINE,
CAUSALMERGE:CAUSALMERGE, CAPS:CAPSMEAS, CAPRATIOS:CAPRATIOSMEAS,
USECAPPING:USECAPPING, MINCAPHIST:MINCAPHIST, PLANINT:PLANINTMEAS,
PLANCUMINT:PLANCUMINTMEAS, CAPINTERVALS:CAPINTERVALS]
```

Sample 1 – startdate as String:

```
FORECASTOUT <-FORECAST(BAYESALPHA:0.15, FORECASTLENGTH:12, HISTORY:RSAL,
MASK:METHMASK1, MAXALPHA:0.99, MAXB:50, MAXWINTERSALPHA:0.99, MINCROSTON:5,
MINHOLT:13, MINWINTERS:104, PERIOD:52, PLAN:BAYES_PLAN1, SMOOTHBASELINE:TRUE,
STARTDATE:"D19980505", TRENDDAMP:0.5, HISTSTART:H_STARTDATE1,
FRCSTSTART:F_STARTDATE1)
```

Sample 2 – startdate as Parameter Measure:

```
FORECAST:FORECASTOUT <-FORECAST(BAYESALPHA:0.15, FORECASTLENGTH:12, HISTORY:RSAL,
MASK:METHMASK1, MAXALPHA:0.99, MAXB:50, MAXWINTERSALPHA:0.99, MINCROSTON:5,
MINHOLT:13, MINWINTERS:104, PERIOD:52, PLAN:BAYES_PLAN1, SMOOTHBASELINE:TRUE,
STARTDATEMEAS:TODAY*, TRENDDAMP:0.5, HISTSTART:H_STARTDATE1,
FRCSTSTART:F_STARTDATE1)
```

Sample 3 – Getting Multiple Results

```
FORECAST: FORECASTMEAS,  INT:INTMEAS,  CUMINT:CUMINTMEAS,  PEAKS: PEAKSMEAS <-
FORECAST(BAYESALPHA:0.15, FORECASTLENGTH:12, HISTORY:RSAL, MASK:METHMASK1,
MAXALPHA:0.99, MAXB:50,  MAXWINTERSALPHA:0.99, MINCROSTON:5, MINHOLT:13,
MINWINTERS:104, PERIOD:52, PLAN:BAYES_PLAN1, SMOOTHBASELINE:TRUE,
STARTDATEMEAS:TODAY*, TRENDDAMP:0.5, HISTSTART:H_STARTDATE1,
FRCSTSTART:F_STARTDATE1)
```

Configuration Parameters and Rules

Input Parameters

The table below provides the input parameters for the Forecast procedure.

Parameter Name	Description
BAYESALPHA	The maximum Bayesian alpha value. Data Type: Real Multiple Allowed: No Required: No
BAYESIAN_HORIZ	The horizon to which the Bayesian adjust is applied. Data Type: Integer Multiple Allowed: No Required: No
CAPRATIOS	Cap ratio for each time series. Data Type: Real Multiple Allowed: No Required: No
CAPS	Caps for each time series. Data Type: Real Multiple Allowed: No Required: No
DDPROFILE	De-seasonalized demand measure used only for profile-based forecasting. Data Type: Real Multiple Allowed: No Required: No
FORECASTLENGTH	The length of the forecast. Data Type: Integer Multiple Allowed: No Required: Yes
FRCSTSTART	The forecast start date. Data Type: Datetime Multiple Allowed: No Required: No
FRCSTSTARTMEAS	The measure of the forecast start dates. Data Type: Datetime Multiple Allowed: No Required: No
HISTORY	The input measure the forecast is based on. Data Type: Real Multiple Allowed: No Required: Yes

Parameter Name	Description
HISTSTART	The historical start date. Data Type: Real Multiple Allowed: No Required: No
KEEPCLAMOEDMAXB	Determines whether variables exceeding maxb are clamped or values are dropped and regression is re-run. Data Type: Real Multiple Allowed: No Required: No
KEEPCLAMPEDMINB	Determines whether variables exceeding minb are clamped or values are dropped and regression is re-run. Data Type: Real Multiple Allowed: No Required: No
MASK	Array that identifies what forecast method is used for each time series. See the Method list; Refer to Forecast Model/Model List table. Data Type: Boolean Multiple Allowed: No Required: Yes
MAXALPHA	The maximum alpha value. Data Type: Real Multiple Allowed: No Required: No
MAXB	The maximum ratio between beta and baseline. Data Type: Real Multiple Allowed: No Required: No
MAXPROFILEALPHA	The maximum alpha for the Profile method. Data Type: Real Multiple Allowed: No Required: No
MAXWINTERSALPHA	The maximum alpha in the Winters method. Data Type: Real Multiple Allowed: No Required: No
MINB	The minimum ratio between beta and baseline. Data Type: Real Multiple Allowed: No Required: No

Parameter Name	Description
MINCAPHIST	The minimum number of weeks before capping can be used. Data Type: Real Multiple Allowed: No Required: No
MINCROSTON	The minimum Croston history. Data Type: Integer Multiple Allowed: No Required: No
MINHOLT	The minimum Holt history. Data Type: Integer Multiple Allowed: No Required: No
MINWINTERS	The minimum Winters history. Data Type: Integer Multiple Allowed: No Required: No
PERIOD	The forecasting period for calculating seasonal coefficients. Data Type: Integer Multiple Allowed: No Required: Yes
PLAN	The Plan measure. Data Type: Real Multiple Allowed: No Required: No
PLANCUMINT	The cumulative Interval of the plan associated with the plan (PARAMETER forecast); Bayesian only. Data Type: Real Multiple Allowed: No Required: No
PLANINT	The interval of the plan associated with the plan (PARAMETER forecast); Bayesian only. Data Type: Real Multiple Allowed: No Required: No
PROFILE	The Seasonal Profile measure. Data Type: Real Multiple Allowed: No Required: No
PROMO	The Promo variable measure (one for each promotion). Data Type: Integer Multiple Allowed: Yes Required: No

Parameter Name	Description
PROMO_IN_BASELINE	An indicator used to identify if the promotion is incorporated in the baseline. Data Type: Boolean Multiple Allowed: No Required: No
PROMOEFF	The calculated promotional effects (one per promotion). Data Type: Real Multiple Allowed: Yes Required: No
PROMOOVER	The promo effect override measure (one for each promotion). Data Type: Boolean Multiple Allowed: Yes Required: No
PROMOTYPE	The promo type measure (one for each promotion). Data Type: Integer Multiple Allowed: Yes Required: No
READMODE	Indicates whether mode is Random or Sequential. Data Type: Integer/enum Multiple Allowed: No Required: No
SMOOTHBASELINE	When value is TRUE, historical baseline is smoothed prior to future baseline forecast. Defaults to TRUE in casual method. Data Type: Boolean Multiple Allowed: No Required: No
SPREADPROF	The profile to spread to final forecast level. Data Type: Real Multiple Allowed: No Required: No
STARTDATE/ STARTDATEMEAS	The forecast start date. Either STARTDATE or STARTDATEMEAS is required. Data Type: STARTDATE - Date as a string. Data Type: STARTDATEMEAS – Date as measure. Multiple Allowed: No Required: Yes
TRENDDAMP	The trend damping parameter. Data Type: Real Multiple Allowed: No Required: No

Parameter Name	Description
USECAPPING	A Boolean measure that indicates whether capping is applied. Data Type: Boolean Multiple Allowed: No Required: No
VALID_DD	The maximum non-zero history to use de-seasonalized demand value for seasonal profile based forecasting. Data Type: Integer Multiple Allowed: No Required: No
CAPINTERVALS	When set to TRUE, interval and cumint are capped. Data Type: Boolean Multiple Allowed: No Required: No
VERBOSE	When set to TRUE, detail information is added to log file. Useful for debugging. Data Type: Boolean Multiple Allowed: No Required: No

Output Parameters

The table below provides the output parameters for the Forecast procedure.

Parameter Name	Description
ALERT	A high-level forecast alert generated by the forecast engine. Data Type: Boolean Multiple Allowed: No Required: No
CHALPHA	ES alpha. Data Type: Real Multiple Allowed: No Required: No
CHLEVEL	ES level. Data Type: Integer Multiple Allowed: No Required: No
CHMETHOD	Selected method. Refer to Forecast Model/Model List table. Data Type: Integer Multiple Allowed: No Required: No

Parameter Name	Description
CHTREND	ES trend. Data Type: Real Multiple Allowed: No Required: No
CUMINT	Cumulative interval forecast. Data Type: Real Multiple Allowed: No Required: No
FORECAST	Forecast output. Data Type: Real Multiple Allowed: No Required: Yes
INT	Interval forecast for Standard Deviation. Data Type: Real Multiple Allowed: No Required: No
PEAKS	Peaks, which are used for calculating baseline of the forecast. Data Type: Real Multiple Allowed: No Required: No

Forecast Method/Model List

The table below provides the numeric value assigned to the forecast model/model list.

Model	Numeric Value
AUTO ES	1
SIMPLE	2
HOLT	3
WINTERS	4
CASUAL	5
AVERAGE	6
NO FORECAST	7
COPY	8
CROSTON	9
M. WINTERS	10
A. WINTERS	11
SIMPLE CROSTON	12
BAYESIAN	13
LOADPLAN	14
PROFILE	15

C

Appendix: Configuring the LostSale Function/Procedure

Overview

The purpose of Oracle Retail's LostSale module, which may also be referred to as "Preprocessing," is to correct past data points that represent unusual sales values that are not representative of a general demand pattern. Such corrections may be necessary when an item is out of stock and cannot be sold, which usually results in low sales. Data Correction may also be necessary in a period when demand is unusually high. The LostSale module allows you to automatically make adjustments to the raw POS (Point of Sales) data so that subsequent demand forecasts do not replicate undesired patterns that are caused by lost sales or unusually high demand.

The "LostSale Syntax" section contains the specifications and syntax for configuring the LostSale function in the RPAS Configuration Tools. There is an RPAS multi-return function named preprocess and one RPAS special expression named preprocess. The special expression provides better performance; however, it only works in the batch mode. The multiple return function preprocess works in both batch mode and workbook mode. The syntax is exactly the same in both modes, except that procedures use "<-" instead of "=" in the expression.

> **Note:** The syntax is slightly different than the standard RPAS functions and procedures that are described in the "Rule Functions Reference Guide" section of the *RPAS Configuration Tools User Guide*.

LostSale Requirements

The following librariesmust be registered in any domain(s) that will use the LostSale solution extension:

- AppFunctions
- LostSalesFunctions

Configuration Restrictions

The following restrictions apply to use the LostSale function/procedure:

- An underscore (_) character may not be used in any measure names and rules unless the measures and rules are to be expanded using the RDF or Curve solution's classification scheme.

  The classifications apply the AppFunctions and are as follows:

  - _F: Expand measures and rules across final levels
  - _S: Expand measures and rules across source levels
  - _B: Expand measures and rules across birth dates

LostSale Parameter/Model Dependencies

The model listed below required the stated measure to be provided.

- Bayesian model – Plan measure required.
- Profile model – Profile measure required.

Using the LostSale Function

The following notes are intended to serve as a guide for configuring the LostSale function within the RPAS Configuration Tools:

1. The LostSale function is a multi-result function, meaning that it can return multiple results within one function call within a rule. In order to get multiple results, the resultant measures must be configured in the Measure Tool, and the specific measure label must be used on the left-hand side (LHS) of the function call. The resultant measure parameters must be comma-separated in the function call as in the example.
2. Because different filtering methods require different input parameters, it is necessary that every input parameter (measure or constant) must be accompanied by the corresponding label. All of the input measure parameters must be configured and registered before the function call. The input parameters must be comma-separated in the function call as in the example.
3. The LostSale function library must be registered after the domain build by using the regfunction RPAS utility.
4. The LostSale function required all the input and output measures using the same intersections. Mixed input/output measure intersections should be aligned to the same calculation intersection with other RPAS function/procedure before calling the LostSale function. The same procedure can be carried out to the resultant measures to spread or aggregate them to the designated intersections.
5. Because of the limitation that the same measure cannot simultaneously appear on both left-hand side and right-hand side, the implementation of the CLEAR filter requires the user to provide a LSOVER_REF measure (a duplication of the previously calculated LSOVER measure) when you try to retain the results on certain time series but clear the others by providing a mask measure (TSMASK_DENSE). The LSOVER_REF is not required when the results for all the time series need to be cleared.
6. The LSTODAY measure is used to specify the end date for the filter processing. It only accepts the index number for the end date along the calendar dimension as valid input. If it is desired that the string position name to be used for the end date specification, the available RPAS time dimension translation function “index” can be used to do the name-index conversion before calling the LostSale function.
7. The LSTODAY input parameter is designed to be a measure rather than a constant to provide more flexibility. Current implementation only allows one global LSTODAY index value to be used in processing all the time series. To specify the end date, you just need to populate its value for the first time series, and this index will be applied to all the other time series.
8. The index value in LSTODAY measure started from 1 rather than 0.
9. FLP_FIRST and FLP_LAST are the resultant measures to be used for the First-Last-Populated Location calculation. They do not have the calendar dimension, and each of their cell values represent the indices for the first and last populated locations along the calendar dimension from the first time series up to the current time series, respectively.

10. TSMASK_DENSE is a Boolean input measure without calendar dimension to specify which time series is going to be processed and which is not. For filtering methods other than the CLEAR method, the true value means that it will be processed if the popcount for the current time series is larger than the hard-coded threshold value. Otherwise, it will not be processed. The false value means that the current time series will not be processed. If the TSMASK_DENSE measure is not specified, all the time series will be processed and the internal hard-coded threshold value will not be considered. For the CLEAR filtering method, the true value means that the previously calculated results for the current time series will be cleared and the false value means the results will be retained. If the TSMASK_DENSE measure is not specified, all the results will be cleared.
11. For all the input measures that do not have the calendar dimension, such as UP_ADJ_RATIO and DELTA, you can use a constant as input. In this case, the constant value will be applied to all the time series.

Syntax Conventions

The table below displays the syntax conventions used in this document.

Indicator	Definition
[...]	All options listed in brackets are optional.
{...|...}	Options listed in "{}" with " | " separators are mutually exclusive (either/or).
{...,...}	Options listed in "{}" with "," separators way are a complete set.
Bold	Labels.
Italics	Italics indicate a temporary placeholder for a constant or a measure.
Italics/meas	This indicates that the placeholder can be either a constant or a measure.
BoldItailics	This indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N.
Normal	Normal text signifies required information.
Underlined	This convention is used to identify the function name.

LostSale Syntax

The syntax for using the LostSale appears below. The input and output parameter tables explain the specific usage of the parameters names use in the function/procedure.

Generic Example 1:

```
LSOVER: LSOVERMEAS, LS: LSMEAS, [, TSALERT: TSALERTMEAS, SERVICE_LEVEL:
SERVICELEVELMEAS, STOCK_LEVEL: STOCKLEVELMEAS, FLP_FIRST: FLPFIRSTMEAS, FLP_LAST:
FLPLASTMEAS] = preprocess(SRC: SRCMEAS, LSTODAY: LSTODAYMEAS, NPTS: NPTSMEAS [,
MIN_TSALERT: MINTSALERTMEAS, OUTAGE: OUTAGEMEAS, TSMASK_DENSE: TSMASKMEAS,
UP_ADJ_RATIO: UPADJMEAS, DOWN_ADJ_RATIO: DOWNADJMEAS, REFERENCE: REFMEAS,
DEVIATION: DEVMEAS {, WINDOW: WINDOWMEAS | , WINDOW1: WINDOW1MEAS, WINDOW2:
WINDOW2MEAS, WINDOW3: WINDOW3MEAS, WINDOW4: WINDOW4MEAS, WINDOW5: WINDOW5MEAS} {,
ALPHA: ALPHAMEAS, NPAS: NPASMEAS, NFUT: NFUTMEAS} {, NSIGMA_MIN: NSIGMA_MINMEAS,
NSIGMA_MAX: NSIGMA_MAXMEAS | , NSIGMAOUT_MIN: NSIGMAOUT_MINMEAS, NSIGMAOUT_MAX:
NSIGMAOUT_MAXMEAS, NSIGMAADJ_MIN: NSIGMAADJ_MINMEAS, NSIGMAADJ_MAX:
NSIGMAADJ_MAXMEAS} {, FRCST_MIN: FRCST_MINMEAS, HIST_MIN_FS: HIST_MIN_FSMEAS} {,
PRICE: PRICEMEAS, INVENTORY: INVENTORYMEAS, HIST_MIN_MD: HISTMINMDMEAS} , DELTA:
DELTAMEAS, LSOVER_REF: LSOVERREFMEAS]
```

Generic Example 2:

```
LSOVER: LSOVERMEAS, LS: LSMEAS, [, TSALERT: TSALERTMEAS, SERVICE_LEVEL:
SERVICELEVELMEAS, STOCK_LEVEL: STOCKLEVELMEAS, FLP_FIRST: FLPFIRSTMEAS, FLP_LAST:
FLPLASTMEAS] <-preprocess(SRC: SRCMEAS, LSTODAY: LSTODAYMEAS, NPTS: NPTSMEAS [,
MIN_TSALERT: MINTSALERTMEAS, OUTAGE: OUTAGEMEAS, TSMASK_DENSE: TSMASKMEAS,
UP_ADJ_RATIO: UPADJMEAS, DOWN_ADJ_RATIO: DOWNADJMEAS, REFERENCE: REFMEAS,
DEVIATION: DEVMEAS {, WINDOW: WINDOWMEAS | , WINDOW1: WINDOW1MEAS, WINDOW2:
WINDOW2MEAS, WINDOW3: WINDOW3MEAS, WINDOW4: WINDOW4MEAS, WINDOW5: WINDOW5MEAS} {,
ALPHA: ALPHAMEAS, NPAS: NPASMEAS, NFUT: NFUTMEAS} {, NSIGMA_MIN: NSIGMA_MINMEAS,
NSIGMA_MAX: NSIGMA_MAXMEAS | , NSIGMAOUT_MIN: NSIGMAOUT_MINMEAS, NSIGMAOUT_MAX:
NSIGMAOUT_MAXMEAS, NSIGMAADJ_MIN: NSIGMAADJ_MINMEAS, NSIGMAADJ_MAX:
NSIGMAADJ_MAXMEAS} {, FRCST_MIN: FRCST_MINMEAS, HIST_MIN_FS: HIST_MIN_FSMEAS} {,
PRICE: PRICEMEAS, INVENTORY: INVENTORYMEAS, HIST_MIN_MD: HISTMINMDMEAS} , DELTA:
DELTAMEAS]
```

Sample 1:

```
LSOVER:LSOVER1, LS:LS1, TSALERT:TSALERT1 = preprocess(SRC:POS, METHODID:MTHID,
LSTODAY:TODAY1, NPTS:NPTS, WINDOW:WIN)
```

Sample 2:

```
LSOVER:LSOVER1, LS:LS1, TSALERT:TSALERT1 <- preprocess(SRC:POS, METHODID:MTHID,
LSTODAY:TODAY1, NPTS:NPTS, WINDOW:WIN)
```

Configuration Parameters and Rules

Input Parameters

The table below provides the input parameters for the LostSale procedure.

Parameter Name	Description
SRC	The source data. Data Type: Real Required: Yes
METHODID	The filtering method ID. Data Type: Real Required: Yes
LSTODAY	The end date for filter processing. Data Type: Real Required: Yes
NPTS	The number of points into history that will be filtered. Data Type: Real Intersection: Required: Yes
MIN_TSALERT	The threshold value used to set off TSALERT. Data Type: Real Required: No
OUTAGE	The outage indicator. Data Type: Boolean Required: No
TSMASK_DENSE	A Boolean value to specify which time series will be processed. Data Type: Boolean Required: No
UP_ADJ_RATIO	The upward adjustment ratio that will be applied on LS. Data Type: Real Required: No Default value: 1.0*
DOWN_ADJ_RATIO	The downward adjustment ratio that will be applied on LS. Data Type: Real Required: No Default value: 1.0*
REFERENCE	Reference will be used for source data substitution. Data Type: Real Required: No

Parameter Name	Description
DEVIATION	The standard deviation for confidence interval calculation by Forecast Sigma filters. Data Type: Real Required: No
WINDOW	Filter window length for Standard Median filter. Data Type: Real Required: No Default value: 13
WINDOW1	First round filter window length for Retek Median filter. Data Type: Real Required: No Default value: 13
WINDOW2	Second round filter window length for Retek Median filter. Data Type: Real Required: No Default value: 19
WINDOW3	Third round filter window length for Retek Median filter. Data Type: Real Required: No Default value: 7
WINDOW4	Forth round filter window length for Retek Median filter. Data Type: Real Required: No Default value: 5
WINDOW5	Fifth round filter window length for Retek Median filter. Data Type: Real Required: No Default value: 11
ALPHA	The exponential coefficient used to evaluate past and future velocities. Data Type: Real Required: No Default value: 0.2
NPAST	The maximum number of historical points to calculate past velocity. Data Type: Real Required: No Default value: 5

Parameter Name	Description
NFUT	The maximum number of historical points to calculate future velocity. Data Type: Real Required: No Default value: 5
NSIGMA_MIN	The number of standard deviations for lower bound calculation. Data Type: Real Required: No Default value: 3.0
NSIGMA_MAX	The number of standard deviations for upper bound calculation. Data Type: Real Required: No Default value: 3.0
FRCST_MIN	The forecast lower bound for Forecast Sigma filters. Data Type: Real Required: No Default value: 0.1
HIST_MIN_FS	The minimum number of historical points required for Forecast Sigma filters. Data Type: Real Required: No Default value: 5
NSIGMAOUT_MIN	The number of standard deviations for lower outlier calculation. Data Type: Real Required: No Default value: 3.0
NSIGMAOUT_MAX	The number of standard deviations for upper outlier calculation. Data Type: Real Required: No Default value: 3.0
NSIGMAADJ_MIN	The number of standard deviations for lower bound calculation. Data Type: Real Required: No Default value: 1.5

Parameter Name	Description
NSIGMAADJ_MAX	The number of standard deviations for upper bound calculation. Data Type: Real Required: No Default value: 1.5
PRICE	The historical price data. Used by Mark Down filter only. Data Type: Real Required: No
INVENTORY	The historical inventory data. Used by Mark Down filter only. Data Type: Real Required: No
HIST_MIN_MD	The minimum number of historical points. Used by Mark Down filter only. Data Type: Real Required: No Default value: 5
DELTA	Ratio of reference will be used to copy or increase for OVERRIDE and INCREMENT filters. Data Type: Real Required: No Default value: 1.0*
LSOVER_REF	Data will be used to override SRC. Used by CLEAR filter only. Data Type: Real Required: No

* If the measure is not specified, the default value will be applied to each of the time series to be processed.

Output Parameters

The table below provides the output parameters for the LostSale function/procedure.

Parameter Name	Description
LSOVER	Adjusted source data. It is the Primary Result LSOVER = SRC + LS Data Type: Real Required: Yes
LS	The adjustment on the source data. Data Type: Real Required: Yes
TSALERT	Boolean flag set to TRUE when more than MIN_TSALERT number of data points have been modified. Data Type: Boolean Required: No
SERVICE_LEVEL	SERVICE_LEVEL = SRC / LSOVER Data Type: Real Required: No
STOCK_LEVEL	Used by Mark Down filter only. Data Type: Real Required: No
FLP_FIRST	First populated position. Used by FLP filter only. Data Type: Real Required: No
FLP_LAST	Last populated position. Used by FLP filter only. Data Type: Real Required: No

Forecast Method/Model List

The table below provides the numeric value assigned to the forecast model/model list.

Model	Numeric Value	Comments
MEDIAN5	0	Retek Median Required input parameters: None Optional input parameters: ▪ WINDOW1 ▪ WINDOW2 ▪ WINDOW3 ▪ WINDOW4 ▪ WINDOW5
MEDIAN1	1	Standard Median Required input parameters: None Optional input parameters: WINDOW

Model	Numeric Value	Comments
OVERRIDE	2	Override Required input parameters: REFERENCE Optional input parameters: DELTA
INCREMENT	3	Increment Required input parameters: REFERENCE Optional input parameters: DELTA
ES_LT	4	Standard ES Required input parameters: OUTAGE Optional input parameters: ▪ ALPHA ▪ NPAS ▪ NFU
LS_ES_LT	9	Lost Sales – Standard ES Required input parameters: OUTAGE Optional input parameters: ▪ ALPHA ▪ NPAS ▪ NFU
FRCST_SIGMA	14	Forecast and standard deviation algorithm. Required input parameters: ▪ REFERENCE ▪ DEVIATION Optional input parameters: ▪ NSIGMA_MAX ▪ NSIGMA_MIN ▪ FRCST_MIN ▪ HIST_MIN_FS
FRCST_SIGMA_EVENT	15	Forecast and standard deviation algorithm with Event. Required input parameters: ▪ OUTAGE ▪ REFERENCE ▪ DEVIATION Optional input parameters: ▪ NSIGMAOUT_MAX ▪ NSIGMAOUT_MIN ▪ NSIGMAADJ_MAX ▪ NSIGMAADJ_MIN ▪ FRCST_MIN ▪ HIST_MIN_FS

Model	Numeric Value	Comments
MARK_DOWN	16	Markdown removal – interpolation on Mdarea. Required input parameters: None Optional input parameters: ▪ PRICE ▪ INVENTORY ▪ HIST_MIN_MD
CLEAR	17	Clear – clears specified result measures. Required input parameters: None Optional input parameters: ▪ TSMASK_DENSE ▪ LSOVER_REF
CLEAR_ALERT	18	Clear Alert measure. Required input parameters: None Optional input parameters: None
NO_FILT	19	No filtering. Required input parameters: None Optional input parameters: None
FLP_CALC	20	First and last populated location calculation. Required input parameters: None Optional input parameters: None
LS_MEDIAN5_EVENT	21	Lost Sales – Retek Median with Event. Required input parameters: OUTAGE Optional input parameters: ▪ WINDOW1 ▪ WINDOW2 ▪ WINDOW3 ▪ WINDOW4 ▪ WINDOW5

LostSale Filtering Methods

Standard Median

Standard Median is recommended for getting data baselines on long time ranges when promo indicators are not available.

A standard median filter implementation

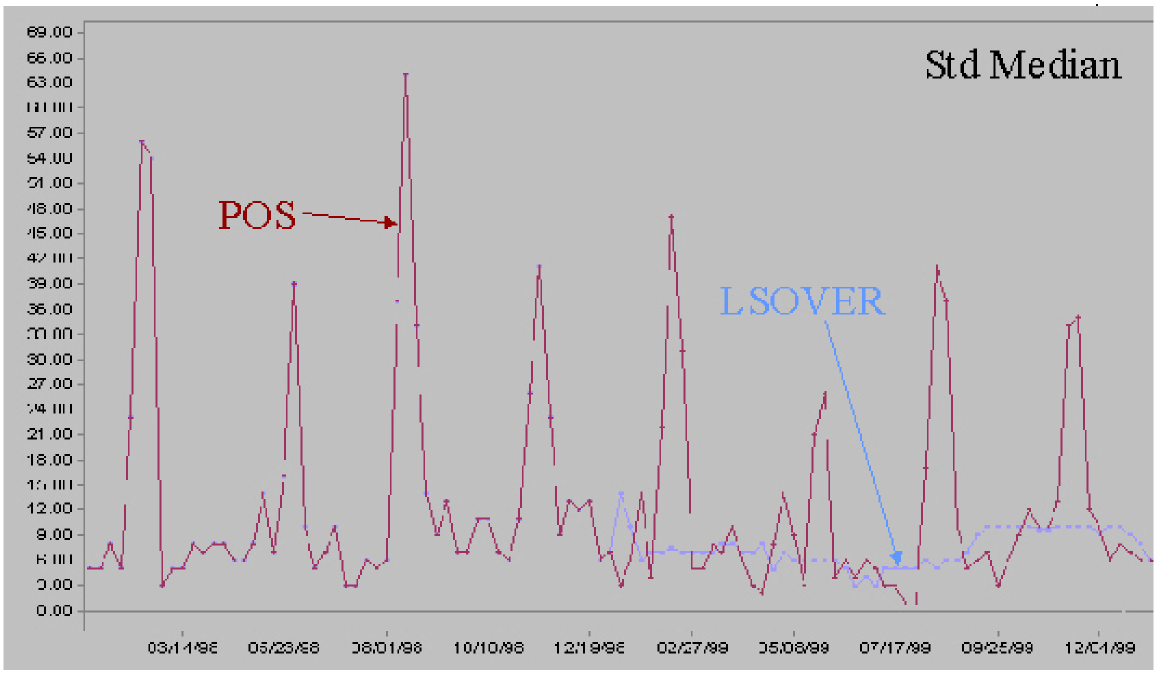
- Does not take outage information as an input.
- Can use one optional parameter: window length.

Mathematical Formulation

LSOVER(t) = median value of SRC over [t-window/2, t+window/2],

Where: “window” is the parameter window length of the filter.

Example Chart



Std Median with “window” = 13 points

Example:

```
LSOVER:lsover1, LS:ls1, TSALERT:tsalert1 = preprocess(SRC:pos, METHODID:mthid,
LSTODAY:today1, NPTS:npts, WINDOW:win)
```

Retek Median

Retek Median is recommended for getting data baselines on long time ranges when promo indicators are not available.

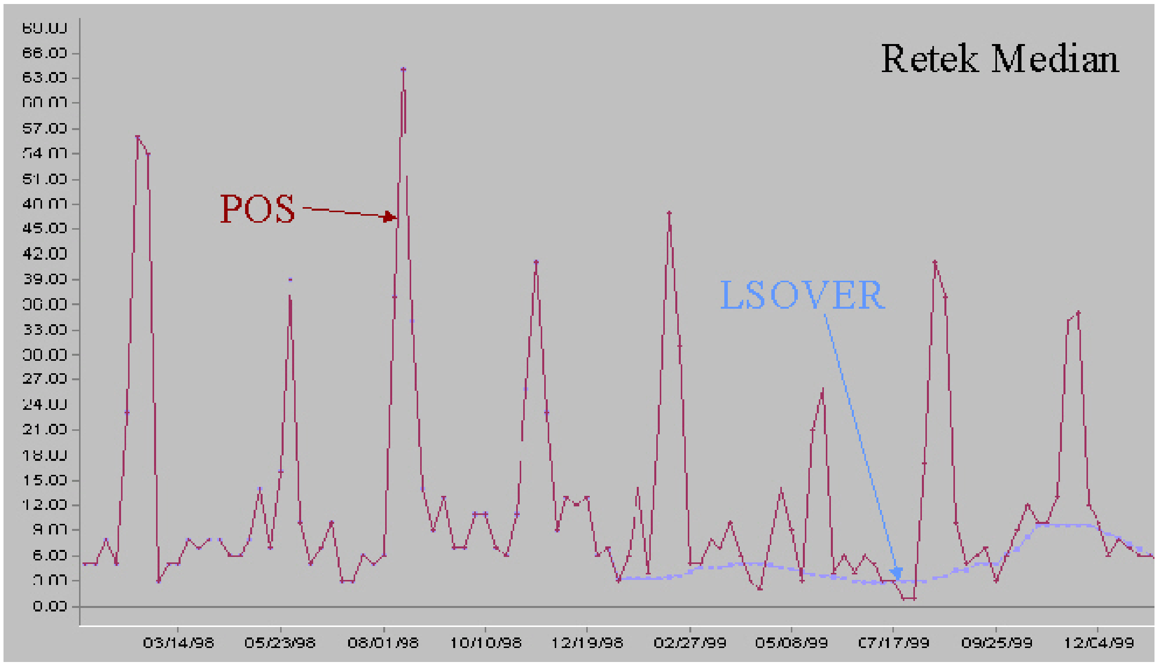
Retek Median provides the following features:

- A sophisticated median filter that takes trends into consideration and improves side effects over the standard median filter. It makes five standard median filter passes.
- Does not take outage information as an input.
- Can accept five optional parameters: window length for each pass.

Mathematical Formulation

1. The first two passes recursively apply the standard median filter. The result is denoted by MEDIAN_2(t). The one-step difference of MEDIAN_2(t) is calculated. That is, DIFF_1(t) = MEDIAN_2(t) – MEDIAN_2(t-1). Then, the standard median filter is applied to DIFF_1(t). The result is denoted by MEDIAN_DIFF_1(t).
2. Using MEDIAN_DIFF_1(t), a first smoothed version (that is, baseline) of the source data is calculated at the third step: SMOOTH_1(t) = SMOOTH_1(t-1) + MEDIAN_DIFF_1(t) on points where the absolute deviation of SRC(t) over its mean is larger than half of the global absolute standard deviation. Otherwise, SMOOTH_1(t) = SRC(t).
3. To prepare for the fourth pass, the one-step difference of SMOOTH_1(t) is calculated. That is, DIFF_2(t) = SMOOTH_1(t) - SMOOTH_1(t-1). An average version of DIFF_2(t) is calculated using the standard median filter. The result is denoted by AVG_DIFF_2(t). The result of the fourth pass is SMOOTH_2(t) = SMOOTH_2(t-1) + AVG_DIFF_2(t).
4. Finally, LSOVER(t) is the result of applying the standard median filter to SMOOTH_2(t).

Example Chart



Retek Median with Default Parameters

Example:

```
LSOVER:lsover1, LS:ls1, TSALERT:tsalert1 = preprocess(SRC:pos, METHODID:mthid,
LSTODAY:today1, NPTS:npts, WINDOW1:win, WINDOW2:win2, WINDOW3:win3, WINDOW4:win4,
WINDOW5:win5)
```

Standard Exponential Smoothing

Standard Exponential Smoothing is recommended for removing specific spikes of low or non-seasonal data when spike indicators are available.

Standard Exponential Smoothing provides the following features:

- Based on standard Exponential Smoothing calculations of past and future sales velocities.
- Must have “unusual event” measure info as input (that is, also referred to as outage).
- Does not take into consideration seasonal components.
- Three optional parameters:
  1. Alpha (ES parameter used to evaluate past and future velocities)
  2. Maximum number of historical points to calculate past velocity
  3. Maximum number of future points to calculate future velocity

Mathematical Formulation

Std ES is the standard Exponential Smoothing filter. It preprocesses a subset of points as predetermined by an input measure. For every contiguous sequence of points to adjust, say between t_f and t_l, a past velocity and a future velocity are calculated using an exponentially weighted average. For the points between t_f and t_l, the adjustment is calculated as a linear interpolation of the past and future velocities.

$$Past\_Velocity = \frac{\sum_{i=1}^{np}(1-\alpha)^{i-1} * SRC(t_f - i)}{\sum_{i=1}^{np}(1-\alpha)^{i-1}}$$

$$Future\_Velocity = \frac{\sum_{i=1}^{nf}(1-\alpha)^{i-1} * SRC(t_l + i)}{\sum_{i=1}^{nf}(1-\alpha)^{i-1}}$$

$$LSOVER(t) = Past\_Velocity + \frac{Future\_Velocity - Past\_Velocity}{t_l - t_f + 2} * (t - t_f + 1), \forall t \in [t_f, t_l]$$

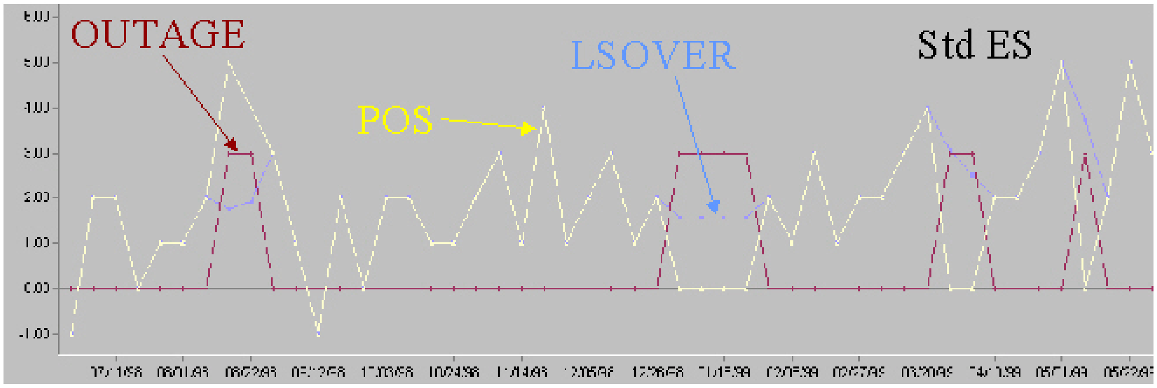
Where:

“ α ” is the exponential coefficient used to evaluate past and future velocities.

„ np ” is the maximum number of historical points to calc past velocity.

„ nf ” is the maximum number of future points to calc future velocity.

Example Chart



Std ES with “α ” = 0.2, “np ” = 2 weeks, and “nf ”= 2 weeks

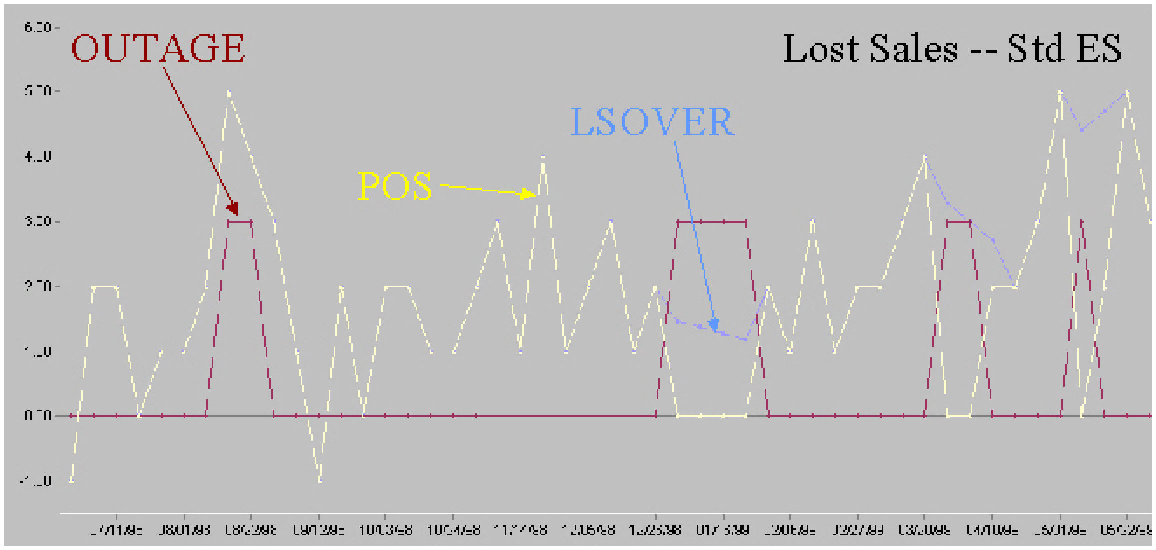
Example:

```
LSOVER:lsover1, LS:ls1, TSALERT:tsalert1 = preprocess(SRC:pos, METHODID:mthid,
LSTODAY:today1, NPTS:npts, OUTAGE:outage1, ALPHA:alpha, NPAST:npast, NFUT:nfut)
```

Lost Sales – Standard Exponential Smoothing

Lost Sales – Standard Exponential Smoothing Functions the same as Std ES except that it only adjusts lost sales (that is, negative spikes).

Example Chart



Lost Sales -- Std ES with “α ” = 0.2, “np ” = 2 weeks, and “nf ”= 2 weeks

Example:

```
LSOVER:lsover1, LS:ls1, TSALERT:tsalert1 = preprocess(SRC:pos, METHODID:9,
LSTODAY:today1, NPTS:30, OUTAGE:outage1, ALPHA:0.2, NPAST:5, NFUT:5)
```

Forecast Sigma

Forecast Sigma is recommended for removing recent spiky data points when approved forecasts and approved confidence intervals are available on the filtering window, but spike indicators are not available. This method is based on the principle that if a data point significantly deviates from an approved forecast, this data point is likely to be an unusual event that should be overridden in the source measure (POSOVER) used by the forecasting engine. It is adjusted by bringing the override value within some bounds of the approved forecast as defined by a proportional coefficient scalar of the forecasts' standard deviation.

Forecast Sigma provides the following features:

- Does not take outage information as an input
- Requires 2 parameters:
  1. Approved forecast array
  2. Approved standard deviation array of forecast
- Can accept 4 optional parameters:
  1. Number of std. deviations for upper bound
  2. Number of std. deviations for lower bound.
  3. Forecast lower bound
  4. Minimum item history (# points) required for filtering

Mathematical Formulation

This method relies on approved forecasts with their corresponding confidence intervals. It adjusts the points that are far (as defined by a multiple of the forecast standard deviation) from their corresponding previously approved forecasts by bringing the override values to their closest confidence interval bounds.

```
IF # historical points < MinHist THEN
    LSOVER(t) = SRC(t)
ELSE IF forecast(t) < MinFrcst THEN
    forecast(t) = MinFrcst AND $\sigma$ = MinFrcst
ELSE IF $\sigma = 0$ THEN
    IF forecast(t) < 1.0 THEN
        $\sigma$ = forecast(t)
    ELSE $\sigma = \sqrt{forecast(t)}$
IF SRC(t) > forecast(t) + nsu*$\sigma$ THEN
    LSOVER(t) = forecast(t) + nsu*$\sigma$
ELSE IF SRC(t) < forecast(t) – nsl*$\sigma$ THEN
    LSOVER(t) = forecast(t) - nsl*$\sigma$
ELSE LSOVER(t) = SRC(t)
```

Where:

"nsu" is the number of std. deviations for upper bound.

"nsl" is the number of std. deviations for lower bound.

"MinFrcst" is the forecast lower bound.

"MinHist" is the minimum item history (# points) required for filtering.

Example Chart

Lost Sales – Forecast Sigma with nsu = 3, nsl = 3, minFrcst = 0.1 and minHist = 5 weeks

Example:

```
LSOVER:LSOVER1, LS:LS1, TSALERT:TSALERT1 = preprocess(SRC:POS, METHODID:mthid,
LSTODAY:TODAY1, NPTS:npts, REFERENCE:forecast1, DEVIATION:dev1,
NSIGMA_MIN:nsigma_min, NSIGMA_MAX:nsigma_max, FRCST_MIN:0.1,
HIST_MIN_FS:hist_min_fs)
```

Forecast Sigma Event

This is similar to Forecast Sigma. It takes an outage (for instance, event) indicator to further process.

Mathematical Formulation

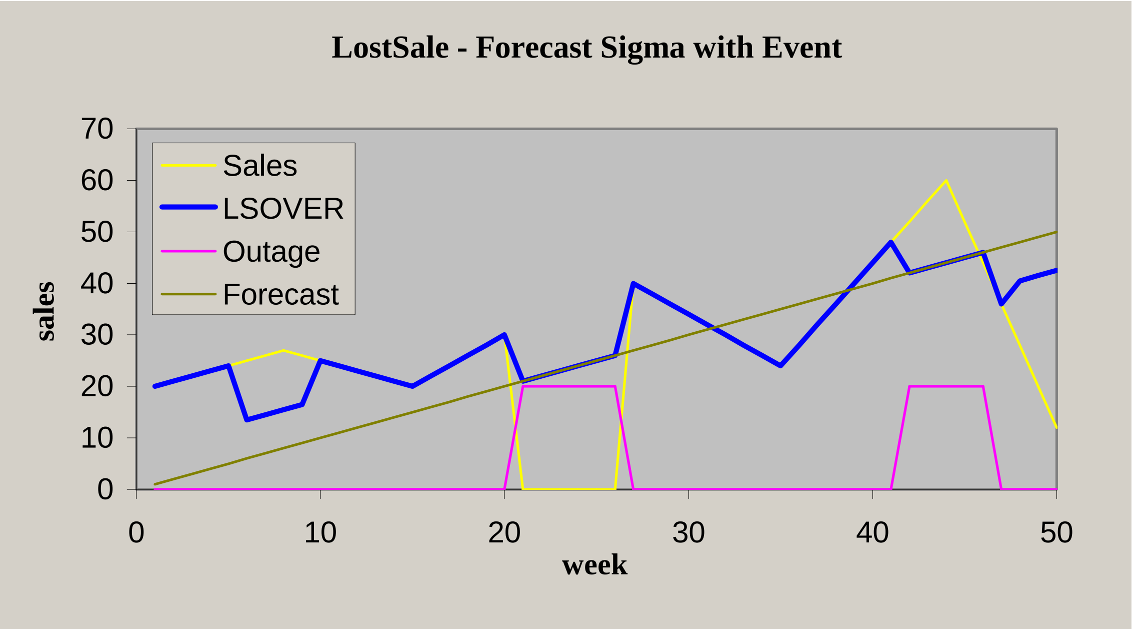
When the outage/event mask is ON:

LSOVER(t) = forecast(t)

When the outage/event mask is OFF:

If the data points that are outside of the outliers calculated through NSIGMAOUT_MIN and NSIGMAOUT_MAX, they will be brought into the confidence interval bounds, which are defined through NSIGMAADJ_MIN and NSIGMAADJ_MAX.

Example Chart



Lost Sales – Forecast Sigma Event with nsigmaout_min = 3, nsigmaout_max = 3, nsigmaadj_min = 1.5, nsigmaadj_max = 1.5,

minFrcst = 0.1 and minHist = 5 weeks

Example:

```
LSOVER:LSOVER1, LS:LS1, TSALERT:TSALERT1 = preprocess(SRC:POS, METHODID:mthid,
LSTODAY:TODAY1, NPTS:npts, OUTAGE:outage1, REFERENCE:forecast1, DEVIATION:dev1,
NSIGMAOUT_MIN:nsigmaout_min, NSIGMAOUT_MAX:nsigmaout_max,
NSIGMAADJ_MIN:nsigmaadj_min, NSIGMAADJ_MAX:nsigmaadj_max, FRCST_MIN:frcst_min,
HIST_MIN_FS:hist_min_fs)
```

Override

This method overrides the destination measure with the source measure that is adjusted by the adjustment percentage according to the mask. It is recommended for filling data gaps when an existing reference measure exists as a default value.

Override provides the following features:

- It is a simple data copy of a given percentage of the reference data to copy from.
- This may or may not take outage (for instance, event) info as an input to mask the operation.
- Requires 2 parameters:
  1. Reference measure to copy data from
  2. Source measure for the original data
- Can accept 1 optional parameter, Ratio of reference to actually copy.

Mathematical Formulation

This method uses the following parameters:

- A source measure that can be any measure in the system as long as it has the same intersection as the destination measure
- A reference measure that can be any measure in the system as long as it has the same intersection as the destination measure
- A destination measure that can be any measure in the system as long as it has the same intersection as the source measure
- A mask that is a Boolean measure that has the same intersection as the source and destination measures
- An adjustment percentage

This method overrides the destination measure with the source measure adjusted by the adjustment percentage according to the mask:

Let:

S(i) is the value in cell (i) of the source measure

R(i) is the value in cell (i) of the reference measure

D(i) is the value in cell (i) of the destination measure

M(i) is the value of cell (i) of the mask

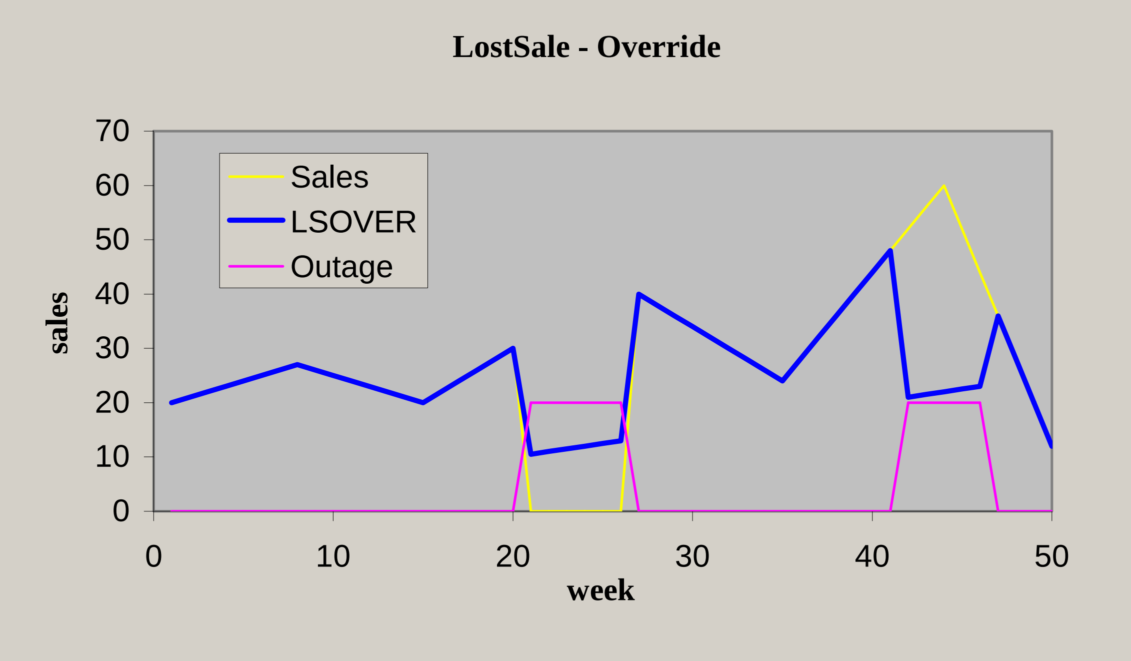
a is an adjustment percentage

The result of the override method is:

D(i) = a * R(i) if M(i) is TRUE

D(i) = S(i) if M(i) is FALSE

Example Chart



Lost Sales – Override with delta = 0.5

Example:

```
LSOVER:lsover1, LS:ls1, TSALERT:tsalert1 = preprocess(SRC:pos, METHODID:mthid,
LSTODAY:today1, NPTS:npts, REFERENCE:ref1, OUTAGE:outage1, DELTA:delta1)
```

Increment

This method increments or decrements the destination measure by the source measure, which is adjusted by the adjustment percentage according to the mask. It is recommended for updating outliers or data gaps when an existing reference measure exists as a default adjustment.

Increment provides the following features:

- It is a simple data increment of a given percentage of the reference data to copy from.
- It may or may not take outage (for instance, event) info as an input to mask the operation.
- Has 1 required parameter, Reference measure to increment by.
- Can accept 1 optional parameter, Ratio of reference to actually increment by.

Mathematical Formulation

This method uses the following inputs:

- A source measure that can be any measure in the system as long as it has the same intersection as the destination measure
- A reference measure that can be any measure in the system as long as it has the same intersection as the destination measure
- A destination measure that can be any measure in the system as long as it has the same intersection as the source measure
- A mask that is a Boolean measure that has the same intersection as the source and destination measures
- An adjustment percentage

This method increments or decrements the destination measure by the source measure, which is adjusted by the adjustment percentage according to the mask.

Let:

S(i) is the value in cell (i) of the source measure

R(i) is the value in cell(i) of the reference measure

D(i) is the value in cell (i) of the destination measure

M(i) is the value of cell (i) of the mask

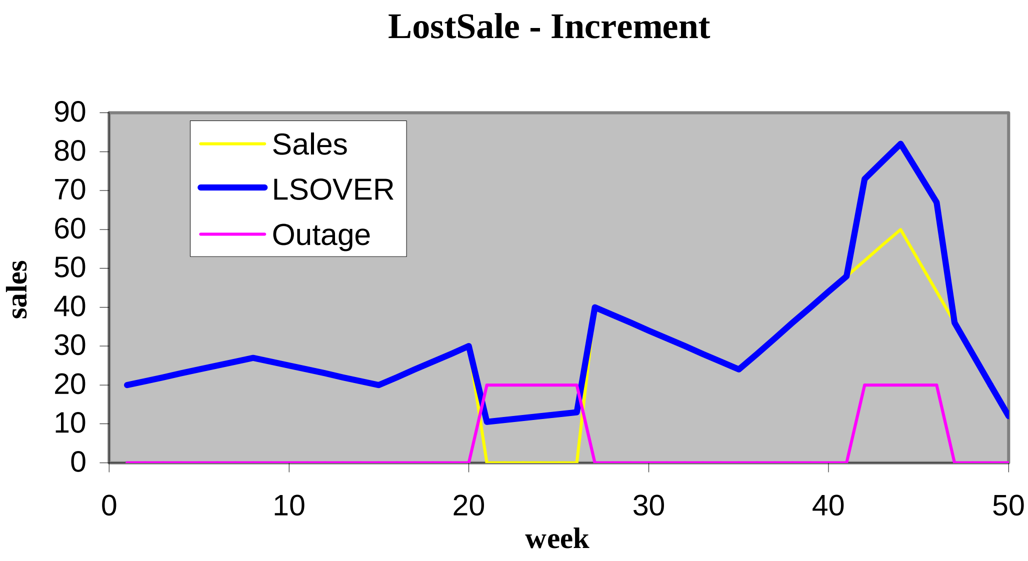
a is an adjustment percentage (can be between (–100%) and (+100%))

The result of the reduction method is:

D(i) = S(i) + a * R(i) if M(i) is TRUE

D(i) = S(i) if M(i) is FALSE

Example Chart



Lost Sales – Increment with delta = 0.5

Example Usage

```
LSOVER:lsover1, LS:ls1, TSALERT:tsalert1 = preprocess(SRC:pos, METHODID:mthid,
LSTODAY:today1, NPTS:npts, REFERENCE:ref1, OUTAGE:outage1, DELTA:delta1)
```

Clear

This is used for canceling the effect of some former preprocessing adjustments.

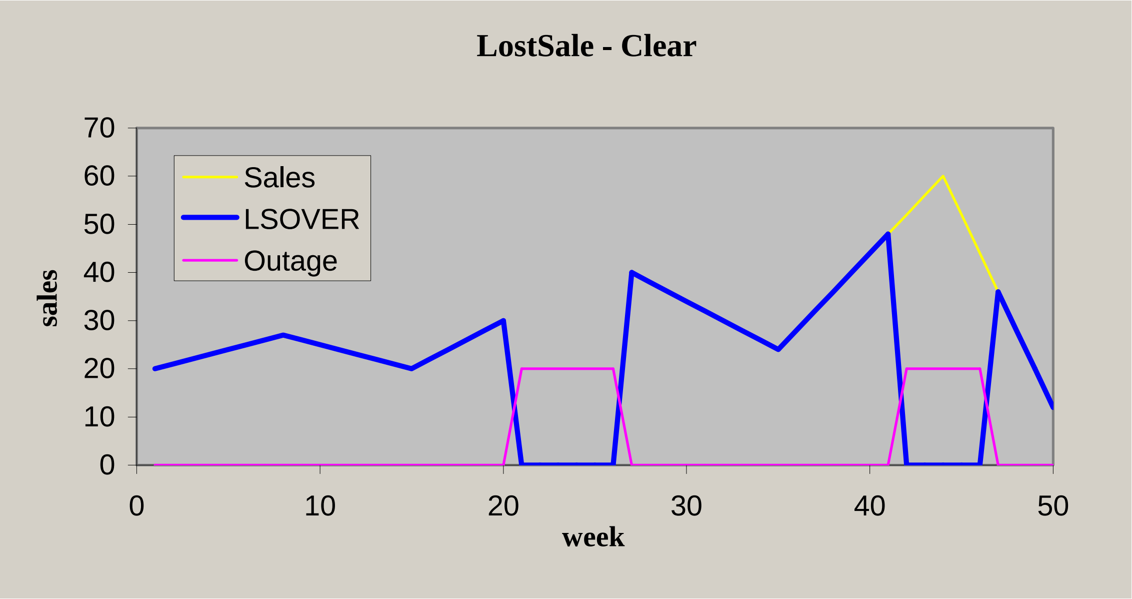
Clear provides the following features:

- Does not take outage information as an input.
- May or may not take time series mask (does not have calendar dimension) input to retain results for certain time series.
- If time series mask is specified, one duplicated LSOVER measure must be provided in addition to the original LSOVER measure.

Mathematical Formulation

```
IF TimeSeriesMask is provided && TimeSeriesMask = false THEN
LSOVER(t) = LSOVER_REF(t)
LS(t) = LSOVER_REF(t) – SRC(t)
ELSE
LSOVER(t) = 0
LS(t) = 0
```

Example Chart



LostSales – Clear with TS_Mask

Example – Clear All

```
LSOVER:LSOVER1, LS:LS1, TSALERT:TSALERT1 = preprocess(SRC:POS, METHODID:mthid,
LSTODAY:TODAY1, NPTS:npts)
```

Example - Partial Clear with Mask Input:

```
LSOVER:LSOVER1, LS:LS1, TSALERT:TSALERT1 = preprocess(SRC:POS, METHODID:mthid,
LSTODAY:TODAY1, NPTS:npts, TSMASK_DENSE:tsMask1, LSOVER_REF:lsoverref1)
```