Retail passion. Innovative solutions.

Retek® Predictive Application Server™ 11.1

Solution Extension Configuration Guide

ORACLE® | Retek

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA

888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

Sales Enquiries:
+44 (0)20 7563 46 46

Fax:
+44 (0)20 7563 46 10

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method	Contact Information
E-mail	support@retek.com
Internet (ROCS)	rocs.retek.com Retek's secure client Web site to update and view issues
Phone	+1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835
Mail	Retek Customer Support Retek on the Mall 950 Nicollet Mall Minneapolis, MN 55403

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.

Contents

Chapter 1 – Introduction

Overview

Beyond the base functionality provided by RPAS, Solution Extensions offer special functionality that may be configured in its own solution or embedded within any RPAS solution. There are several solution extensions delivered in RPAS 11.1 that require a special interface within the RPAS Configuration Tools to provide the necessary components to the configuration. This approach lessens the tedious tasks of configuring hierarchies, measures, rules and workbooks that are required to support the solution extension. The special interfaces supporting the solution extensions are referred to as Plug-Ins. Each of the following solution extensions requires configuration via a Plug-In in the RPAS Configuration Tools:

- Retek Demand Forecasting
- Curve
- Promote
- Grade

Also available are solution extensions that are fully configurable and do not require support via a Plug-In:

- Forecast Procedure
- LostSale Function
- Cluster Procedure
- ASO Space Function
- ASO Assort Function

Note: Configuration of any of the above solution extensions may be limited based on your specific licensing agreement with Retek. Check with your system administrator for more information.

The following chapters provide instructions on the necessary steps to configure each of these solution extensions.

Chapter 2 – Configuring the Curve solution extension

Overview

Curve is an RPAS solution extension used to generate ratios from historical data at user-specified intersections. The profiles generated by Curve can be used for various purposes: to convert the organization-level assortment plans into base level weekly sales forecasts, for generating seasonal forecasts, daily forecasts, or new product forecasting using lifecycle profiles.

RDF requires profiles (created by Curve) to determine how a source level forecast (ex. Item/Chain/Week) is spread down to the execution or final level (ex. Item/Store/Day). Profiles are generated using historical data and phase definitions based on the system configuration. Using the Curve Plug-In, profiles are defined to support the Curve solution.

Curve hierarchy configuration requirements

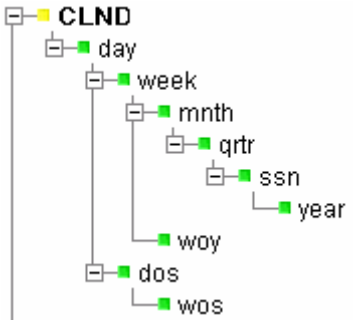
The following dimensions are required to support different seasonal profiles. If the types of profiles defined below are not required, then these hierarchy dimensions may not be necessary:

dos (day of season) - A dimension off day, dos is used to support seasonal profiles normalized to day. This profile should use the 'Daily Seasonal' profile type.

wos (week of season) - A dimension off day or dos, wos is used to support seasonal profiles normalized to week. This profile can use the 'Store Contribution', 'Product Profile' or 'User Defined' profile types.

woy (week of year): A dimension off week or year, woy is used to support weekly seasonal profiles normalized to year. This profile can use the 'Store Contribution', 'Product Profile' or 'User Defined' profile types.

The following illustrates these additional hierarchy requirements that may be configured by the user:

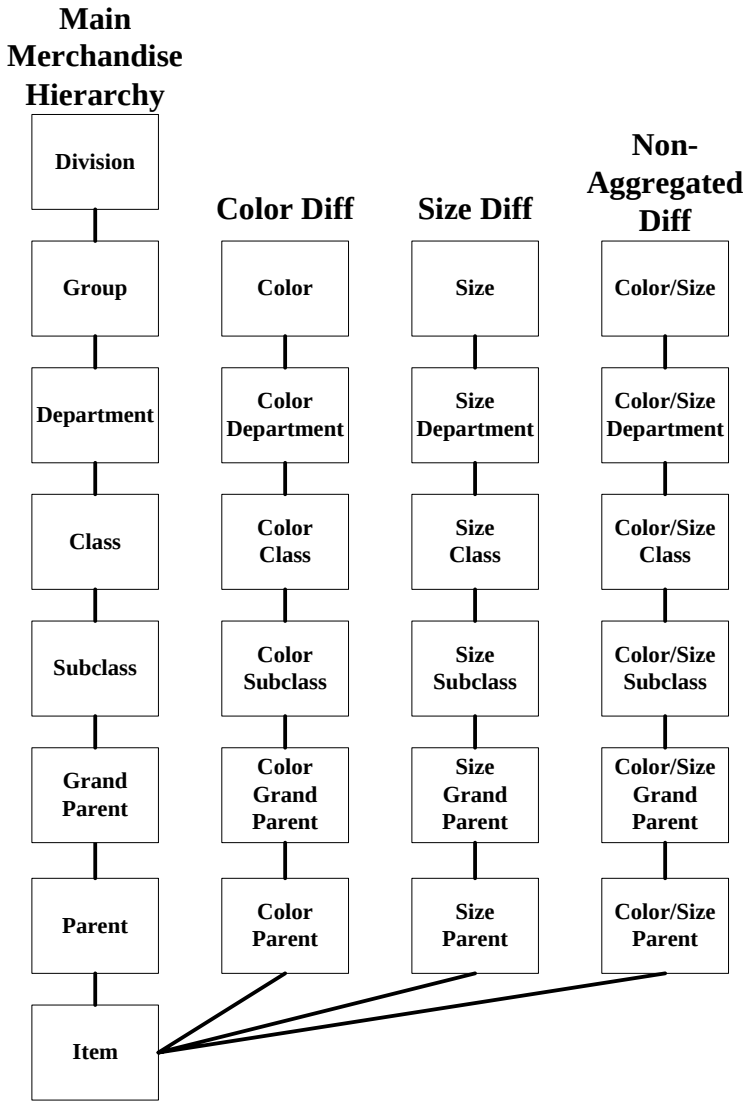


Example: Additional hierarchy requirements for Season Profiles

Configuring Curve differentiator dimensions (optional) and merchandise hierarchy requirements to support RMS

Configuring Differentiator dimensions (also referred to as Diff dimensions) within the merchandise hierarchy is optional. Differentiator dimensions allows the merchandise dimensions to be distinguished based upon an alternative attribute property such as Color, Size, Flavor (or other attributes properties required to support your merchandising needs).

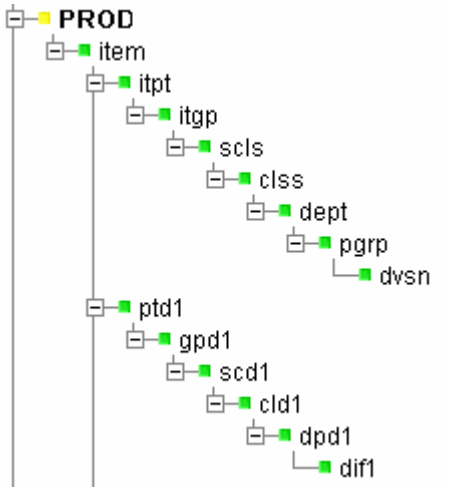
Differentiator dimensions are the combination of each Differentiator and a dimension created off of the lowest dimension in the merchandise hierarchy (item) and only goes up as high as Department. The mock install configuration released with Curve (LabsGA2 and LabsGA3) is configured with an example of a Differentiator branch along the merchandise hierarchy however; up to 10 Differentiator branches may be configured. Below is the example of how a Differentiator branch may be configured:



Example: Differentiator branch

If Curve is to be integrated with RMS Allocation, the Diff dimensions configured in the RPAS Configuration Tools must map to the same Diff dimensions that are or will be configured in the RMS Allocation hierarchies. As well, Allocation also requires Non-Aggregated Differentiator dimensions. These dimensions allow for Diff Dimensions to be combined (as shown in the diagram above) and allow for Curve to generate profiles to support Allocation. Within Allocation, these Non-Aggregated Differentiators are represented by Diffs with Aggregation Indicators set to 'No'.

Within the merchandise hierarchy, also required to support RMS Allocation is the ‘itpt’ (Item Parent) dimension off the ‘item’ (Item) dimension. And off of the ‘itpt’ dimension, add ‘itgp’ (Item Grandparent) dimension. The other aggregate dimensions above ‘item’, should be dimensions beginning off of ‘itgp’. The following illustrates the GA configuration of the merchandise hierarchies configured using Item Parent and Item Grandparent, in addition to a Differentiator branch:



Example: Merchandise hierarchy configuration

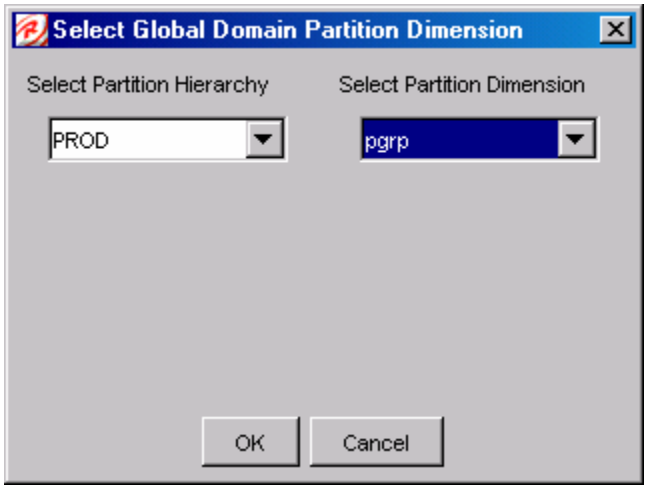
In addition to the above example, profiles 30 through 43 in the mock installations provided in the release packages (LabsGA2 and LabsGA3), are diff profile configurations that may be used to support the generation of spreading ratios for RMS Allocation.

Procedures

Creating a Curve solution extension

To create a Curve solution extension:

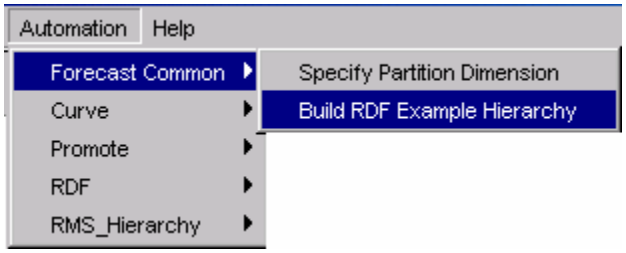
1. Open an existing configuration in which hierarchies (for example, Product, Location and Calendar) have already been defined
2. From the Configuration Tools toolbar, select the 'Automation' menu. If installing a Global Domain environment, go on to step 3. If installing a Simple Domain environment, go on to step 4.
3. Select 'Forecast Common', then the 'Specify Partition Dimension'. You will see the screen in the below example. Select the hierarchy in which the domains will be partitioned, then the Partition Dimension. Select 'OK' and continue with step 4.



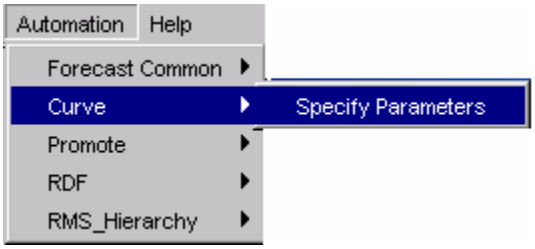
Example: Select Global Domain Partition Dimension

Note: To access this dialog, the configuration must already be flagged as a Global Domain environment. This is done by accessing the Workspace Properties from the File menu, place a checkmark next to 'GlobalDomain'.

4. Optional: the 'Forecast Common' plug-in may also be used to create an example hierarchy configuration. Select 'Forecast Common', then 'Build RDF Example Hierarchy'. The resulting hierarchy configuration is the same hierarchy used for the mock installation configurations (LabsGA2 and LabsGA3) provided in the release packages.



5. From the Curve option, select ‘Specify Parameters’. The following sections outline the process for configuring profiles.



Configuring a final profile

To create a final profile:

1. On the Curve Parameters utility, click the F icon.
2. A new final profile is added and is assigned the next consecutive number starting with ‘01’.
3. Specify the properties for the final profile. See Editing Profile Properties for details.

Configuring source level attributes

To create a source level:

1. On the Profile and Source Level window, highlight the final profile number in which a source will be created.
2. Click the S icon. A new source profile is added and is assigned the next consecutive number.
3. Specify the properties for the Source Level. See Editing Profile Properties for details.

Editing profile properties

To edit profile properties:

Profile Name – The Profile Name is the system-assigned level number when a Final Profile or Source Level is created. This is a read-only field.

Profile Label – The Profile Label is the profile 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 Profile Labels include the Profile Name (which is the system-assigned profile number). There are two reasons for this:
    1. The Profile Name is referenced in the RDF configuration to specify Spreading Profiles. Curve requires that profiles 1 through 9, be referenced as 01, 02,…, 09 when being specified as a Spreading Profile in the RDF configuration.
    2. The 'Default Source Profile' parameter in the Profile Administration workbook is a pick-list populated with the Profile Name of each source level configured for the final profile being viewed in the workbook. If the Default Source Profile set within this configuration, it not expected to change within the domain(s), this recommendation may not be necessary for consideration.
- RPAS will automatically put () around Profile Labels. The configuration specialist should not include these in their level label configuration or the installer will fail. An example of a Profile Label that would violate this requirement is (01 – chn->str-Final). It is acceptable as 01 – chn->str-Final.
- '-' should not be used before or after the Profile Label. An example of a Profile Label that would violate this requirement is –01-chn->str-Final-. It is acceptable as 01 – chn->str-Final.
- ':' should not be used at all in the Profile Label. An example of a Profile Label that would violate this requirement is 01: chn->str Final.

Profile Type – Assigned on the final profile, the Profile Type is a pick-list of profile types used to determine the profile algorithm and validation required by the profile level. Profile Types are represented with pre-defined configuration information.

The following Profile Types share the same profile algorithm. The rationale for providing different types that have the same behavior is strictly to remind the user of the intent of the profile while using the Profile Administration workbook:

- **Store Contribution Profile** - The Store Contribution Profile is used to determine the data relationship between stores to aggregate dimensions in the location hierarchy.
- **Hourly Profile** - The Hourly Profile is used to determine the data relationship between hour to an aggregate dimension in the calendar hierarchy (for example, hour to week).
- **Daily Profile** - The Daily Profile is used to determine the data relationship between a given day to the week in which it belongs.
- **Product Profile** - The Product Profile is used to determine the data relationship between any two dimensions along the product hierarchy.

- **Size Profile** - The Size Profile is used to determine the data relationship between any dimension in the size hierarchy and any dimension in the product hierarchy. A size hierarchy must be defined to use this profile type.
- **User Defined Profile** - The User Defined Profile may be used to support any profile configuration.

The following Profile Types have unique behavior:

- **Diff Profile** – Diff Profiles are used to determine spreading ratios from aggregate dimensions in the Product hierarchy to diff dimensions. Used to support the spreading of data in RMAS Allocation, Diff Profiles exhibit the same behavior as the previous profile types. However, unique to Diff Profiles is special validation of the relationship between the defined diff dimensions to dimensions along the main branch of the Product hierarchy (see the RDF Administrator's Guide for more information on validation criteria).
- **Daily Seasonal Profile** - The Daily Seasonal Profile is used to determine the data relationship between a given day of the week to aggregate dimensions in the calendar hierarchy. This profile type uses training window data to compute the profile. The resulting profile is then clipped to fit within the defined phase window.
- **Life Cycle Profile** - The Life Cycle Profile uses data along a user-defined training window, then stretches or shrinks data to fit a user-defined phase window.

Profile Intersection – The Profile Intersection is the intersection at which an intermediate profile is calculated. This intermediate profile is then replicated down or aggregated up to the Stored Intersection. If the Store Intersection is the same as the Profile Intersection, the values in intermediate profile are copied to the Stored Intersection. The Profile Intersection must be lower than the Aggregation Intersection. If the profile is being used as the Spreading Profile in RDF, this Profile Intersection should be the same as the Final Forecast Level.

- Once the Profile Intersection is entered at the Final Profile level, the Stored Intersection for both the Final and Source (if created) will populate with the same value. These may be overwritten if necessary.

Note: If installing a Global Domain environment, all intersections configured to support a profile MUST include a dimension at or below the partition dimension.

Aggregation Intersection – The Aggregation Intersection is the intersection at which the profile will sum to one (or 100%). If the profile is being used as the Spreading Profile in RDF, this Aggregation Intersection should be the same as the Source Forecast Level.

- Once the Aggregating Intersection is entered, the Approval Intersection will populate with the same value for both the Final and Source (if created) will populate with the same value. This may be overwritten if necessary.

Note: If installing a Global Domain environment, all intersections configured to support a profile MUST include a dimension at or below the partition dimension.

Approval Intersection – Assigned only at the Final Profile, the Approval Intersection is the intersection at which the profile is approved. Approval Intersection should be above or equal to the Aggregation Intersection. If the profile is being used as the Spreading Profile in RDF, this Approval Intersection should be the same as the Aggregation Intersection.

- The Approval Intersection may be pre-populated with the value set for the Aggregation Intersection. This may be overwritten if necessary.

Note: If installing a Global Domain environment, all intersections configured to support a profile MUST include a dimension at or below the partition dimension.

Stored Intersection – The Stored Intersection is the destination intersection of the profile. The intermediate profile produce at the Profile Intersection is either replicated down to or aggregated up to the Stored Intersection. If the Store Intersection is the same as the Profile Intersection, the values in intermediate profile are copied to the Stored Intersection. The Stored Intersection not should be greater than the Aggregation Intersection. If the profile is being used as the Spreading Profile in RDF, this Stored Intersection should be the same as the Profile Intersection.

- The Stored Intersection may be pre-populated with the value set for the Profile Intersection. This may be overwritten if necessary.

Note: If installing a Global Domain environment, all intersections configured to support a profile MUST include a dimension at or below the partition dimension.

Default Source – Assigned only at the Final Profile, the Default Source is the primary Source Level that will be used in the calculation of the Final Profile. The desired Source Level must be created before it is an option in this pick-list.

Source Data – Assigned only at the Final Profile, the Source Data is the measure to be used as the input data (for example, POS) for the generation of profiles. The values in this pick-list are populated with all measures configured external to the RDF, Curve and Promote solution extensions.

- If the profile is to be used to support the dynamic generation of spreading ratios (Spreading Profile) in the RDF batch forecast process, no value in Source Data should be specified.

Editing the Curve GA configuration

The autogeneration process creates hierarchies, measures, rules and workbook templates required to support the essential Curve 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:

'Curve' Solution Extension Name – The name assigned to the resulting Curve solution after autogeneration occurs **cannot** be edited.

Major and Minor Classes – Additional Major and Minor classes may be added to the Curve GA Configuration. The Major and Minor classes 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 Curve 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 Curve GA Configuration. As well, new Measures and Rules may be added to the GA Configuration Workbook Templates. This is done by adding new Major and Minor classes, then adding new Rules to existing Rule Groups in the GA Configuration.

Deleting a Profile Level

Deletion of a profile level will cause the system-assigned enumerated values in the Profile Name to renumber such that levels are in consecutive order starting with profile level 01. Deleting a profile level may impact any solution configuration that utilizes a specific profile level. For example, the following parameters within an RDF Solution configuration may be affected if profile levels are deleted or renumbered:

- Seasonal Profile
- Spreading Profile

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.

Procedures

Deleting a profile level

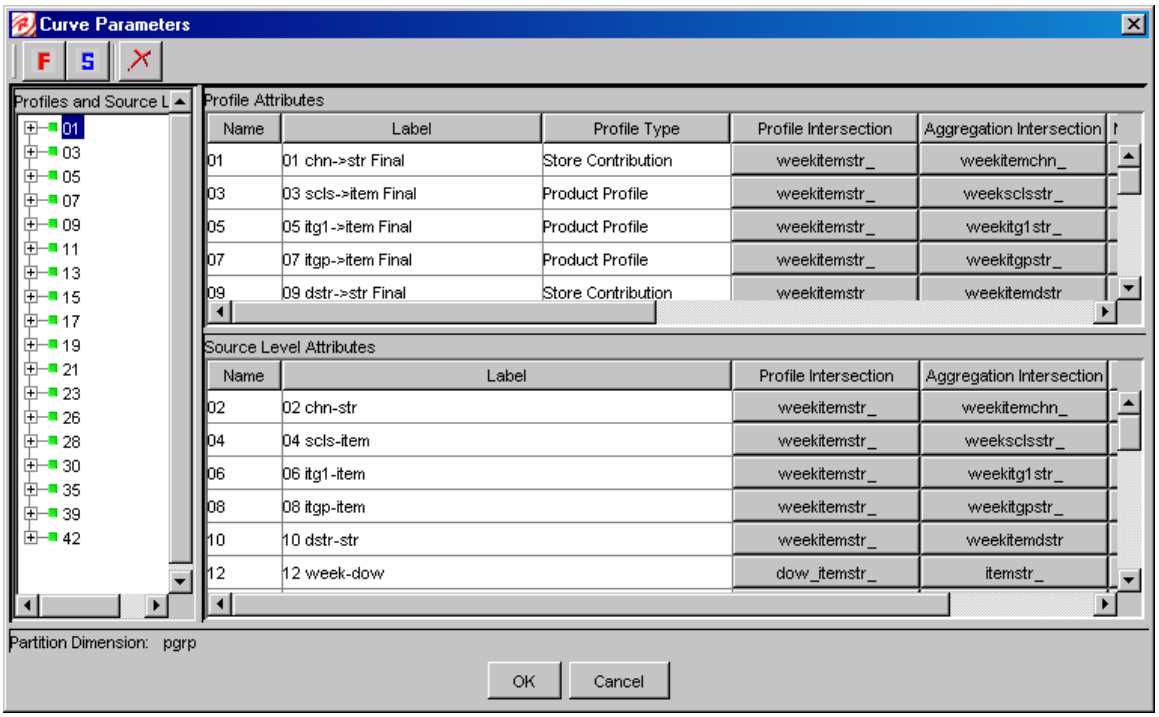
To delete a level:

1. On the Profiles and Source Level window, highlight the number of the profile you want to delete.
2. Click the X icon.
3. The profile is deleted. If you delete a final profile, any source profiles that are associated with it will also be deleted.
4. Select the ‘OK’ icon to regenerate the solution with the changes to the cluster configuration.

Note: See the RDF 11.1 Administration Guide for more information on patchable changes to the configuration.

Curve Plug-In example

The following example shows the Curve Parameters plug-in.



Curve Parameters window

Chapter 3 – Configuring the Retek Demand Forecasting solution extension

Overview

Retek Demand Forecasting™ is a statistical forecasting solution. It 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, enabling a profitable and customer-oriented approach to predicting and meeting product demand.

Often forecast information is required for items at the lowest levels in a hierarchy. Problems can arise when historic sales data for these items is too sparse and 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, 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 generated at an aggregate level and then spread down to lower levels can be compared to forecasts 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: Ability to configure the RDF solution may be limited 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 pre-configuration data requirements

There are several parameters within the RDF configuration that may reference other measures configured external to the solution. Specifically: Source Data, Plan Data, Spreading Profile and Seasonal Profile. It is required that prior to configuring an RDF solution that these measures already exist within the Project.

Source data

The RDF plug-in populates a pick-list with all measures that have been created external to an RDF, Curve or Promote solution extension. If the sales data measure that will be used for Forecasting has not been created, it cannot be selected from the Source Data pick-list. If however an Approved Forecast will be specified as the Source Data for a Final Level, a duplicate measure can be created external to a solution extension. For example, appf0lXB may be configured in the same solution as the sales measures.

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

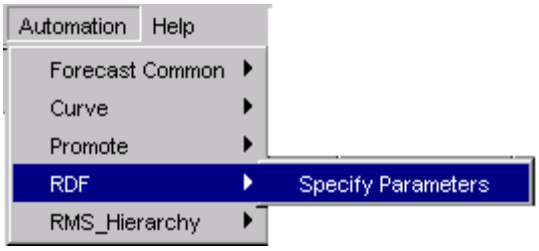
If the ‘Plan Data’ that will be used to support Bayesian forecasting is being defined within another solution, this measure should already exist. However, the entry of this parameter is not required within the configuration and can be entered in the resulting domain(s).

Procedures

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.



Configuring a final forecast level

To create a final level:

1. On the Forecasting Parameters utility, click the F icon.
2. A new final level is added and is assigned the next available level number.
3. 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 is assigned the next available number.
3. Specify the properties for the source level. See Editing Forecast Level Properties for details.

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 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 domain(s).
- 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, see the RDF 11.1 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 measures configured external to the RDF, Curve and Promote solution extensions.

Periodicity – Periodicity is the number of periods within the Calendar dimension defined in the forecast level Intersection. For example, if Intersection is defined at Week/Item/Store the Periodicity value will be 52 (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. It is important to note that the 'Causal' method should be selected as a valid method ONLY for levels in which causal forecasting will be utilized. 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 utilize Causal Forecasting. If Causal is selected and Promote is not licensed or configured, the RDF batch forecast will not generate.

- 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 being 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 being 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).

For more information on Source Level Forecasting, see the RDF 11.1 User Guide.

Autogenerating hierarchies, measures, rules and workbook templates

The following is the process to autogenerate the hierarchies, measures, rules and workbook templates 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:

  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.

  Note: Each time autogenerate is initiated, all measures, rules and workbook template information is overwritten using the latest profile information. See xxx for more information on acceptable configuration changes to the Curve solution.

Deleting a forecast level

Deletion of 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 utilizes a specific level.

As well, 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.

Procedures

Delete a forecast level

To delete a level:

1. On the Forecasting Parameters utility highlight the number of the level you want to delete from the Level window.
2. Click the X icon.
3. The level is deleted. If you delete a final level, any source levels associated with it will also be deleted.
4. Select the ‘OK’ icon to regenerate the solution with the changes to the cluster configuration.

Note: See the RDF 11.1 Administration Guide for more information on patchable changes to the configuration.

Editing the RDF GA configuration

The autogeneration process creates hierarchies, measures, rules and workbook templates 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.

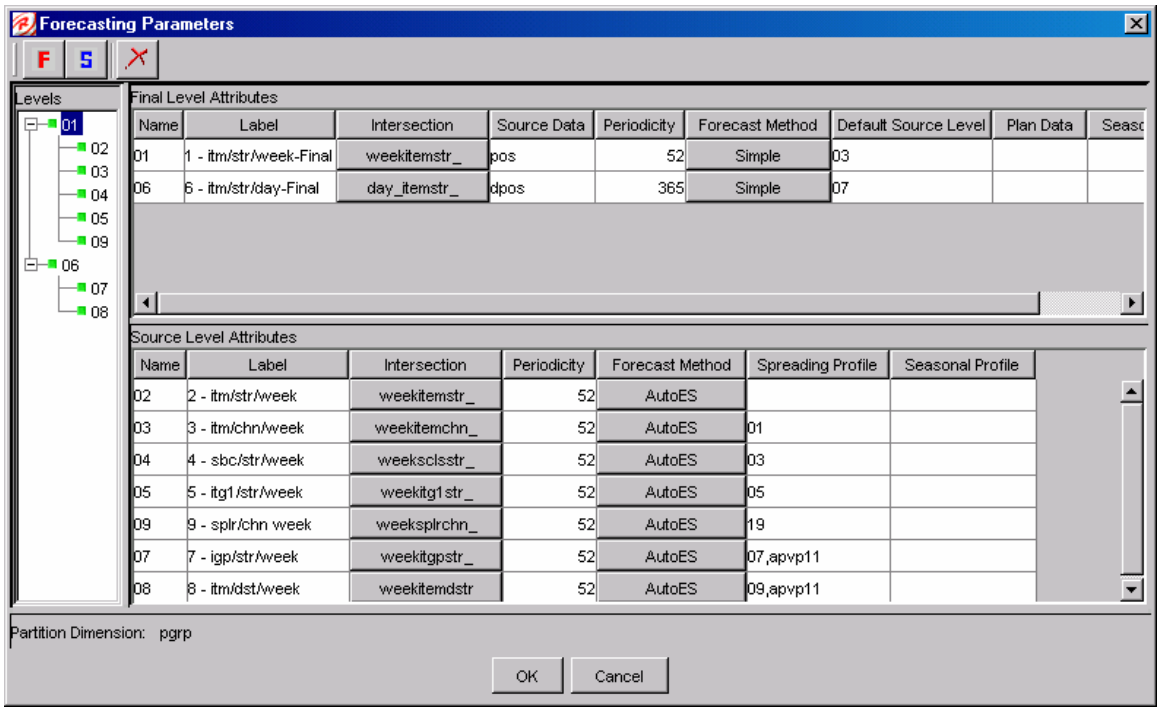
Major and Minor Classes – Additional Major and Minor classes may be added to the RDF GA Configuration. The Major and Minor classes that are part of the GA Configuration may not be edited. This restriction also applies to Measure Names and Measure Labels.

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. As well, new Measures and Rules may be added to the GA Configuration Workbook Templates. This is done by adding new Major and Minor classes, then adding new Rules to existing Rule Groups in the GA Configuration.

RDF example

The following is an example of the Forecasting Parameters utility:



Forecasting Parameters window

Chapter 4 – Configuring the Promote Solution Extension

Overview

Promote 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 both past sales data and promotional information to forecast future demand.

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

Procedures

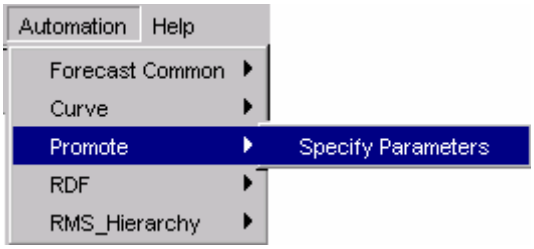
Creating a Promote Solution Extension

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.



Create a Promotion

To create a Promotion:

1. On the Promote Parameters utility, click the P icon.
2. A new promotion is added and is assigned a default promotion number for the Promotion Name (ex. P001).
3. 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 4 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. Special code is used within the domain to create rules as needed for the batch forecast and workbook templates.

  Workbook Templates: All pre-defined workbook templates to support the base Promote solution will be created, however the worksheets are not visible. Special code is used within the domain to create worksheets based on the user's selections in the workbook template wizards.

  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.

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

Note: See the RDF 11.1 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. Special code within the domain handles the creation of rules and workbook template worksheets. It is important to note that 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.

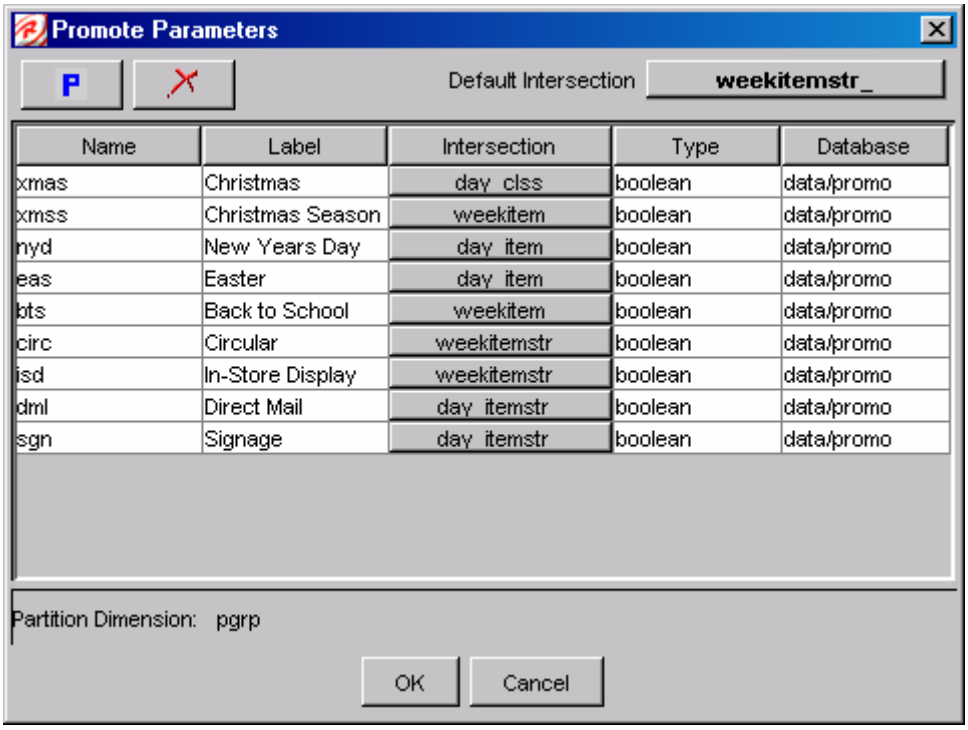
Major and Minor Classes – Additional Major and Minor classes may be added to the Promote GA Configuration. The Major and Minor classes that are part of the GA Configuration may not be edited. This restriction also applies to Measure Names and Measure Labels.

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 since the Promote worksheets are not visible in the configuration.

Promotion example

The following example shows the Promote Parameters utility:



Promote Parameters window

Chapter 5 – Configuring the Grade solution extension

Overview

Grade is a clustering tool that provides insight into how various parts of a retailer's operations can be grouped together. Typically, a retailer may cluster stores over item sales to create logical groupings of stores based upon sales of particular products. This provides increased visibility to where products are selling and allows the retailer to make more accurate decisions in merchandising. Beyond this traditional use of clusters, Grade is flexible enough to cluster any business measure based on products, locations, time, promotions, customers, or any hierarchy configured in the solution.

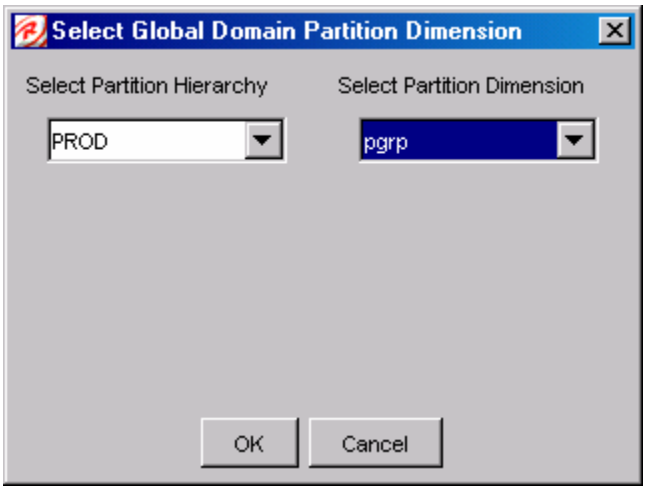
Using the Grade Plug-in, the grades/clusters that will be utilized within the Grade Solution are defined.

Procedures

Creating a Grade solution extension

To create a Grade solution extension:

1. Open an existing configuration in which hierarchies (for example, Product, Location and Calendar) have already been defined.
2. From the Configuration Tools toolbar, select the 'Automation' menu. If installing a Global Domain environment, go on to step 3. If installing a Simple Domain environment, go on to step 4.
3. Select 'Forecast Common', then the 'Specify Partition Dimension'. You will see the screen in the below example. Select the hierarchy in which the domains will be partitioned, then the Partition Dimension. Select 'OK' and continue with step 4.



Select Global Domain Partition Dimension window

Note: To access this dialog, the configuration must already be flagged as a Global Domain environment. This is done by accessing the Workspace Properties from the File menu, place a checkmark next to 'GlobalDomain'.

4. Optional: the 'Forecast Common' plug-in may also be used to create an example hierarchy configuration. Select 'Forecast Common', then 'Build RDF Example Hierarchy'. The resulting hierarchy configuration is the same hierarchy used for the mock installation configurations (LabsGA2 and LabsGA3) provided in the release packages.
5. From the Grade option, select 'Specify Parameters'. The following sections outline the process for configuring profiles.

Create clusters

To create Clusters:

1. On the Grade Parameters utility, enter the Maximum Number of Clusters that will be required to support any Clustering/Grading process.
2. Click on the 'Create Clusters' icon.

   The 'Cluster' and 'Label' parameters will update to reflect the number of clusters specified.
3. Specify the properties for the clusters. See Editing Grade Parameters for details.

Editing Grade parameters

Edit Grade parameters:

Cluster – Cluster is the system assigned Cluster Name. This value cannot be edited.

Label – The Label is the description of the cluster/grade that will be viewed by the user once the domain is created

- Cluster Labels may not exceed forty characters.

The following characters may not precede or follow the label that is entered in this field:

- ‘( )’ Example: (cluster01)
- ‘-‘ Example: -cluster01-

The following may not be use at all in the Cluster Label:

- ‘:’ Example: cluster01:

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 Grade Parameters utility, click the **OK** icon.

  The system automatically generates:

  Hierarchies: The CLSH hierarchy will be created with a clst dimension. The GRCH hierarchy will be created with the grcd dimension.

  Measures: All measures necessary to support the base Grade solution will be created.

  Rules: Only the rules and rule groups necessary to support the installation of the Grade solution are visible in the configuration. Special code is used within the domain to create rules as needed for cluster generation and workbook templates.

  Workbook Templates: All pre-defined workbook templates to support the base Grade solution will be created, however only the worksheets necessary to support the domain installation are visible. Special code is used within the domain to create additional worksheets based on the user's selections in the workbook template wizards.

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

Adding or deleting clusters in the configuration

If it is necessary to remove or add clusters to the configuration the following process may be followed:

1. On the Grade Parameters utility, enter the new Maximum Number of Clusters that will be required to support any Clustering/Grading process.
2. Click on the 'Create Clusters' icon.
3. The 'Cluster' and 'Label' parameters will update to reflect the number of clusters specified.
4. Select the 'OK' icon to regenerate the solution with the changes to the cluster configuration.

Editing the Grade GA configuration

The Grade autogeneration process creates all hierarchy dimensions and measures to support the essential Grade functionality, however only the minimum rules, workbook templates and worksheets required to support the domain installation are visible in the configuration. Special code within the domain handles the creation of rules and workbook template worksheets. It is important to note that this limitation allows for fewer options than in RDF and Curve for edits to the GA Configuration.

The following outlines acceptable changes and restrictions:

'Grade' Solution Extension Name – The name assigned to the resulting Grade solution after autogeneration occurs **cannot** be edited.

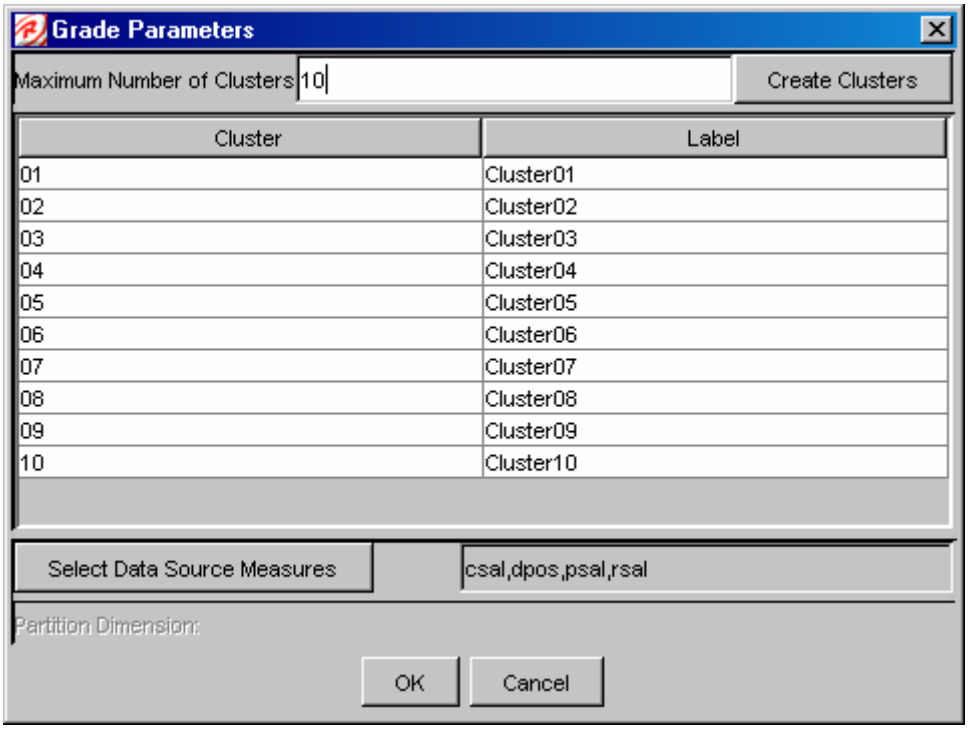
Major and Minor Classes – Additional Major and Minor classes may be added to the Grade GA Configuration. The Major and Minor classes that are part of the GA Configuration may not be edited. This restriction also applies to Measure Names and Measure Labels.

Rules – Additional Rule Sets, Rule Groups and Rules may be added to the Grade 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 Grade GA Configuration. However, new Measures and Rules can only be added to the GA Configuration Workbook Template worksheets visible in the configuration.

Grade example

The following is an example of the Grade Parameters utility:



Grade Parameters window

Chapter 6 – Configuring the Forecast procedure

Overview

RPAS may provide a time-series demand forecast to any solution configured via the RPAS Configuration Tools, also allowing for this forecast to be exported to other systems outside RPAS. The Forecast procedure provides only a small subset of the functionality available through RDF. The following highlights the differences between these solution extensions:

- 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) then this forecast is spread down to the execution level using a profile.
- The Forecast procedure allows for a single forecasting method to be used for a forecast produced only on aggregated data in a domain.
- RDF allows for forecasting methods and forecasting parameters to be modified as needed down to the lowest level in your data.
- Using the Forecast procedure, all forecasted time-series are automatically approved.
- RDF allows for forecast adjustments and approvals to be made at the lowest level necessary in your data.

The following section contains the specifications and syntax for configuring the Forecast procedure. Note that the syntax is slightly different than the standard RPAS functions and procedures described in the RPAS 11.1 Rule Functions Reference Guide.

Syntax legend

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	indicates a temporary placeholder for a constant or a measure (will end with 'meas')
Italics/meas	indicates that the placeholder can be either a constant or a measure.
BoldItalics	indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N
Normal	normal text that has to be there.
Underlined And Large font	Used to identify the function name.

Syntax

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*, **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*, **CUSTOMERID**:*CUSTOMERID*, **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*]

Examples

Example 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)
```

Example 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)
```

Example 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)
```

Forecast Parameter Table

Left Hand Side – Label identifies a Left Hand Side(LHS) Parameter. In other words, these are the labels that identify output parameters that will appear on the left hand side of the equal sign. Otherwise the parameter is Right Hand Side(RHS) or input parameter.

Required – Label identifies required parameters.

Measure – Indicates whether the parameter is expected to be a measure or not.

Allowed to be Multiple – Allowed to be multiple measures.

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
ALERTS	RESULT	N	Y	N	bool	A high-level forecast alert generated by the forecast engine
CHALPHA	RESULT	N	Y	N	double	ES alpha
CHLEVEL	RESULT	N	Y	N	int	ES level
CHMETHOD	RESULT	N	Y	N	int	Chosen method- Refer to Forecast Model/Model List table.
CHTREND	RESULT	N	Y	N	double	ES trend
CUMINT	RESULT	N	Y	N	double	Cumulative interval forecast
FORECAST	RESULT	Y	Y	N	double	forecast output
INT	RESULT	N	Y	N	double	Interval forecast

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
PEAKS	RESULT	N	Y	N	double	Peaks… for calculating baseline of forecast
AGGPROF	PARAMETER	N	Y	N	double	Profile to aggregate promotions to source level
BAYESALPHA	PARAMETER	N	N	N	double	Max Bayesian alpha
BAYESIAN_HORIZ	PARAMETER	N	Y	N	int	Horizon to which the Bayesian adjustment is applied
CAPRATIOS	PARAMETER	N	Y	N	double	Cap ratio for each time series
CAPS	PARAMETER	N	Y	N	double	Caps for each time series
CUSTOMERID	PARAMETER	N	N	N	string	A string ID assigned to a customer. Used only customers that require special processing

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
DDPROFILE	PARAMETER	N	Y	N	double	De-seasonalized demand measure for profile-based forecasting
FORECASTLENGTH	PARAMETER	Y	N	N	int	Length of forecast
FRCSTSTART	PARAMETER	N	N	N	datetime	Forecast start date
FRCSTSTARTMEAS	PARAMETER	N	Y	N	datetime	Measure of forecast start dates
HISTORY	PARAMETER	Y	Y	N	double	Input measure forecast is based on
HISTSTART	PARAMETER	N	Y	N	double	Historical startdate
KEEPCLAMPEDMAXB	PARAMETER	N	N	N	double	Whether variables exceeding maxb are clamped or values are dropped and regression is re-run
KEEPCLAMPEDMINB	PARAMETER	N	N	N	double	Whether variables exceeding minb are clampled or values are dropped and regression is re-run

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
MASK	PARAMETER	Y	Y	N	bool	Array identifying what forecast method is used for each time series. Please see the Method list. Refer to Forecast Model/Model List table.
MAXALPHA	PARAMETER	N	N	N	double	Max alpha parameter
MAXB	PARAMETER	N	N	N	double	Maximum ratio between beta and baseline
MAXPROFILEALPHA	PARAMETER	N	N	N	double	Max alpha for profile method
MAXWINTERSALPHA	PARAMETER	N	N	N	double	Max alpha in Winters method
MINB	PARAMETER	N	N	N	double	Minimum ratio between beta and baseline
MINCAPHIST	PARAMETER	N	N	N	double	Minimum number of weeks before capping can be used.

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
MINCROSTON	PARAMETER	N	N	N	int	Min Croston history
MINHOLT	PARAMETER	N	N	N	int	Min Holt history
MINWINTERS	PARAMETER	N	N	N	int	Min winters history
PERIOD	PARAMETER	Y	N	N	int	Forecasting period for calculating seasonal coefficients
PLAN	PARAMETER	N	Y	N	double	Plan measure
PLANCUMINT	PARAMETER	N	Y	N	double	Cumulative Interval of the plan associated with the plan(PARAMETER forecast). Bayesian only.
PLANINT	PARAMETER	N	Y	N	double	Interval of the plan associated with the plan(PARAMETER forecast). Bayesian only.
PROFILE	PARAMETER	N	Y	N	double	Seasonal profile measure

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
PROMO	PARAMETER	N	Y	Y	int	Promo variable measure(one for each promotion)
PROMO_IN_BASELINE	PARAMETER	N	Y	N	bool	Indicator that says if the promotion is incorporated in the baseline
PROMOEFF	PARAMETER	N	Y	Y	double	Calculated promotional effects (one per promotion)
PROMOOVER	PARAMETER	N	Y	Y	bool	Promo effect override measure (one for each promotion)
PROMOTYPE	PARAMETER	N	Y	Y	int	Promo type measure (one for each promotion)
READMODE	PARAMETER	N	N	N	int/enum	Whether mode is Random or Sequential.

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
SMOOTHBASELINE	PARAMETER	N	N	N	bool	Default to true in the causal method. When it is true, historical baseline is smoothed prior to future baseline forecast
SPREADPROF	PARAMETER	N	Y	N	double	Profile to spread forecast to final level
STARTDATE/ STARTDATEMEAS:	PARAMETER	Y	N	N	date as string/ date as measure	Forecast start date . This or startdatemeas is required
TRENDDAMP	PARAMETER	N	N	N	double	Trend damping parameter
USECAPPING	PARAMETER	N	N	N	bool	Boolean measure that indicates whether capping is used or not.
VALID_DD	PARAMETER	N	N	N	int	Max non-zero history to use de-seasonalized demand value for seasonal profile-based forecasting

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
CAPINTERVALS	PARAMETER	N	N	N	bool	When set to true, interval and cumint is capped.
VERBOSE	PARAMETER	N	N	N	bool	When verbose is set to true, there is much more information print out in the log file. It is useful for debugging

Forecast Method/Model List

Model	Numeric Value
FORECAST	1
SIMPLE	2
HOLT	3
WINTERS	4
CAUSAL	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

Parameter/Model Dependencies

The following is a list of models and the labels of the parameters that are required for the model to work.

1. Bayesian – plan measure
2. Profile – profile measure

Forecast Procedure Configuration Notes

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

- The following syntax must be used within the rule:

  ```
  forecast <- Forecast(…)
  ```

- Please refer to the Forecast documentation for the appropriate input parameters and output measures.
- 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 as in the example below:

  ```
  forecast:frcstout,cumint:cumintout,int:intout<-
  Forecast(forecastlength:12,history:pos,mask:frcstmask,period:26,star
  tdatemeas:todaymeas)
  ```

- For standard deviations, specify the int measure on the LHS of the Forecast procedure call in addition to the forecast measure.
- Startdate (scalar, date type)
- Mask (pick list)
  - range value – see Forecast Method list
  - range syntax: 0(No Forecast), 1(Simple), 2(Holt), 3(Croston)
  - naval – for Planning, set to 8 for Bayesian for all products/locations
- History start date (by prod/loc)
- Forecast start date (by prod/loc)
- Forecast mask (by prod/loc)
- Need to determine the rule that will automatically update the startdatemeas each week with the current week’s date.

Forecast Method Configuration Notes

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

- The syntax for the range is # (value), # (value). For Example:
- 0(No Frcst), 1(Simple), 2(Holt), 3(Croston)
- Spaces are allowed within the label inside the parentheses.
- Define your pick list measure with basestate and aggstate = write. If you define it as read, only the first value in the pick list range will show up.
- Set the measure type to int.
- For the Forecast Method measure, set the baseint to item/str (no week).
- You can specify the naval to a numeric integer within the range, such as 2 in the example above. Then, within the client, it automatically defaults to that value, such as Holt. If you specify the naval outside the range, then the pick list is displayed with NA. Given this ability, the planning teams can then specify 8 as the naval such that Bayesian is used for all item/stores. In this case, it may be desirable to configure the pick list measure as a scalar such that it cannot be changed on an item/store basis.

Chapter 7 – Configuring the LostSale Function

Overview

The purpose of Retek's LostSale module, which may also be referred to as "Preprocessing," is to correct past data points that represent unusual sales values not representative of a general demand pattern. Such correction may be necessary either when an item is out of stock and cannot be sold, resulting in usually low sales. Conversely, correction of data 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 caused by lost sales or unusually high demand.

The following section contains the specifications and syntax for configuring the LostSale function/special expression 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 multi-return function preprocess works in both batch mode and workbook mode. Their syntax are exactly the same except special expressions use "<-" instead of "=" in the expression. Note that the syntax is slightly different than the standard RPAS functions and procedures described in the RPAS 11.1Rule Functions Reference Guide.

Syntax Legend

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	indicates a temporary placeholder for a constant or a measure (will end with 'meas')
Italics/meas	indicates that the placeholder can be either a constant or a measure.
BoldItalics	indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N
Normal	normal text that has to be there.
Underlined And Large font	Used to identify the function name.

Syntax

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*]

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*]

Example 1

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

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

Parameter Table Legend

Group – Whether the label is general enough to be used with all the filtering methods, or can only be used for specific filtering methods.

Input or Output – Whether the label is used as input or output label. In other words, the Label should appear on the left hand side or the right hand side of the function.

Required by All Methods – Whether the label is required for all filtering methods.

Measure – Whether the parameter is expected to be a measure or not.

Has Calendar Dimension – Whether the parameter measure has the calendar dimension or not.

Type – input/output parameter's data type.

LostSale Parameter Table

a	Label	Input Or Output	Required By All Methods	Measure	Has calendar dimension	Type	Default Value
General Parameters	LSOVER	Output	Y	Y	Y	Double	N/A
	LS	Output	Y	Y	Y	Double	N/A
	TSALERT	Output	N	Y	N	Boolean	N/A
	SERVICE_LEVEL	Output	N	Y	Y	Double	N/A
	STOCK_LEVEL	Output	N	Y	N	Double	N/A
	FLP_FIRST	Output	N	Y	N	Double	N/A
	FLP_LAST	Output	N	Y	N	Double	N/A
	SRC	Input	Y	Y	Y	Double	N/A
	METHODID	Input	Y	Y	N	Double	N/A
	LSTODAY	Input	Y	Y	N	Double	N/A

a	Label	Input Or Output	Required By All Methods	Measure	Has calendar dimension	Type	Default Value
	NPTS	Input	Y	Y	N	Double	N/A
	MIN_TSALERT	Input	N	Y	N	Double	N/A
	OUTAGE	Input	N	Y	Y	Boolean	N/A
	TSMASK_DENSE	Input	N	Y	N	Boolean	N/A
	UP_ADJ_RATIO	Input	N	Y	N	Double	1.0*
	DOWN_ADJ_RATIO	Input	N	Y	N	Double	1.0*
Method Specific Parameters	REFERENCE	Input	N	Y	Y	Double	N/A
	DEVIATION	Input	N	Y	Y	Double	N/A
	WINDOW	Input	N	Y	N	Double	13
	WINDOW1	Input	N	Y	N	Double	13
	WINDOW2	Input	N	Y	N	Double	19
	WINDOW3	Input	N	Y	N	Double	7
	WINDOW4	Input	N	Y	N	Double	5
	WINDOW5	Input	N	Y	N	Double	11
	ALPHA	Input	N	Y	N	Double	0.2
	NPAST	Input	N	Y	N	Double	5
	NFUT	Input	N	Y	N	Double	5
	NSIGMA_MIN	Input	N	Y	N	Double	3.0
	NSIGMA_MAX	Input	N	Y	N	Double	3.0
	FRCST_MIN	Input	N	Y	N	Double	0.1

a	Label	Input Or Output	Required By All Methods	Measure	Has calendar dimension	Type	Default Value
	HIST_MIN_FS	Input	N	Y	N	Double	5
	NSIGMAOUT_MIN	Input	N	Y	N	Double	3.0
	NSIGMAOUT_MAX	Input	N	Y	N	Double	3.0
	NSIGMAADJ_MIN	Input	N	Y	N	Double	1.5
	NSIGMAADJ_MAX	Input	N	Y	N	Double	1.5
	PRICE	Input	N	Y	Y	Double	N/A
	INVENTORY	Input	N	Y	Y	Double	N/A
	HIST_MIN_MD	Input	N	Y	N	Double	5
	DELTA	Input	N	Y	N	Double	1.0*
	LSOVER_REF	Input	N	Y	Y	Double	N/A

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

LostSale Parameter Description

Group	Label	Description
General Parameters	LSOVER	Adjusted source data. It is the Primary Result. LSOVER = SRC + LS
	LS	Adjustment on the source data.
	TSALERT	Boolean flag set to true when more than MIN_TSALERT number of data points have been modified
	SERVICE_LEVEL	SERVICE_LEVEL = SRC / LSOVER
	STOCK_LEVEL	Used by Mark Down filter only
	FLP_FIRST	First populated position. Used by FLP filter only
	FLP_LAST	Last populated position. Used by FLP filter only
	SRC	Source data
	METHODID	Filtering method ID
	LSTODAY	End date for filter processing
	NPTS	Number of points into history will be filtered
	MIN_TSALERT	Threshold value used to set off TSALERT
	OUTAGE	Outage indicator
	TSMASK_DENSE	Boolean to specify which time series will be processed
	UP_ADJ_RATIO	Upward adjustment ratio will be applied on LS
	DOWN_ADJ_RATIO	Downward adjustment ratio will be applied on LS
Method Specific Parameters	REFERENCE	Reference will be used for source data substitution
	DEVIATION	Standard deviation for confidence interval calculation by Forecast Sigma filters

Group	Label	Description
	WINDOW	Filter window length for Standard Median filter
	WINDOW1	First round filter window length for Retek Median filter
	WINDOW2	Second round filter window length for Retek Median filter
	WINDOW3	Third round filter window length for Retek Median filter
	WINDOW4	Forth round filter window length for Retek Median filter
	WINDOW5	Fifth round filter window length for Retek Median filter
	ALPHA	Exponential coefficient used to evaluate past and future velocities
	NPAST	Maximum number of historical points to calculate past velocity
	NFUT	Maximum number of historical points to calculate future velocity
	NSIGMA_MIN	Number of std. Deviations for lower bound calculation
	NSIGMA_MAX	Number of std. Deviations for upper bound calculation
	FRCST_MIN	Forecast lower bound for Forecast Sigma filters
	HIST_MIN_FS	Minimum number of historical points required for Forecast Sigma filters
	NSIGMAOUT_MIN	Number of std. Deviations for lower outlier calculation
	NSIGMAOUT_MAX	Number of std. Deviations for upper outlier calculation
	NSIGMAADJ_MIN	Number of std. Deviations for lower bound calculation
	NSIGMAADJ_MAX	Number of std. Deviations for upper bound calculation
	PRICE	Historical price data. Used by Mark Down filter only
	INVENTORY	Historical inventory data. Used by Mark Down filter only
	HIST_MIN_MD	Minimum number of historical points. Used by Mark Down filter only

Group	Label	Description
	DELTA	Ratio of reference will be used to copy or increase for OVERRIDE and INCREMENT filters
	LSOVER_REF	Data will be used to override SRC. Used by CLEAR filter only

LostSale Filtering Method List

Model	Comment	Numeric Value	Implemented	Required method-specific input parameters	Optional method-specific input parameters
MEDIAN5	Retek Median	0	X		WINDOW1, 2, 3, 4, 5
MEDIAN1	Std Median	1	X		WINDOW
OVERRIDE	Override	2	X	REFERENCE	DELTA
INCREMENT	Increment	3	X	REFERENCE	DELTA
ES_LT	Std ES	4	X	OUTAGE	ALPHA, NPAS, NFU
LS_ES_LT	Lost Sales -- Std ES	9	X	OUTAGE	ALPHA, NPAS, NFU
FRCST_SIGMA	Forecast and stddev algo	14	X	REFERENCE, DEVIATION	NSIGMA_MAX, NSIGMA_MIN, FRCST_MIN, HIST_MIN_FS
FRCST_SIGMA_EVENT	Forecast and stddev algo with event	15	X	OUTAGE, REFERENCE, DEVIATION	NSIGMAOUT_MAX, NSIGMAOUT_MIN, NSIGMAADJ_MAX, NSIGMAADJ_MIN, FRCST_MIN, HIST_MIN_FS
MARK_DOWN	Markdown removal -- interpolation on Mdarea	16			PRICE, INVENTORY, HIST_MIN_MD,

Model	Comment	Numeric Value	Implemented	Required method-specific input parameters	Optional method-specific input parameters
CLEAR	Clear (clear specified result measures)	17	X		TSMASK_DENSE, LSOVER_REF
CLEAR_ALERT	Clear Alert measure	18			
NO_FILT	No Filtering (does not do anything)	19	X		
FLP_CALC	First and last populated locations calculation	20	X		
LS_MEDIAN5_EVENT	Lost Sales - Retek Median with Event	21	X	OUTAGE	WINDOW1, 2, 3, 4, 5

Note: LS_MEDIAN5_EVENT is a combination method. Retek median is run first the LS_ES_LT is run on the result from Retek Median.

LostSale Filtering Method Descriptions

Standard Median

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

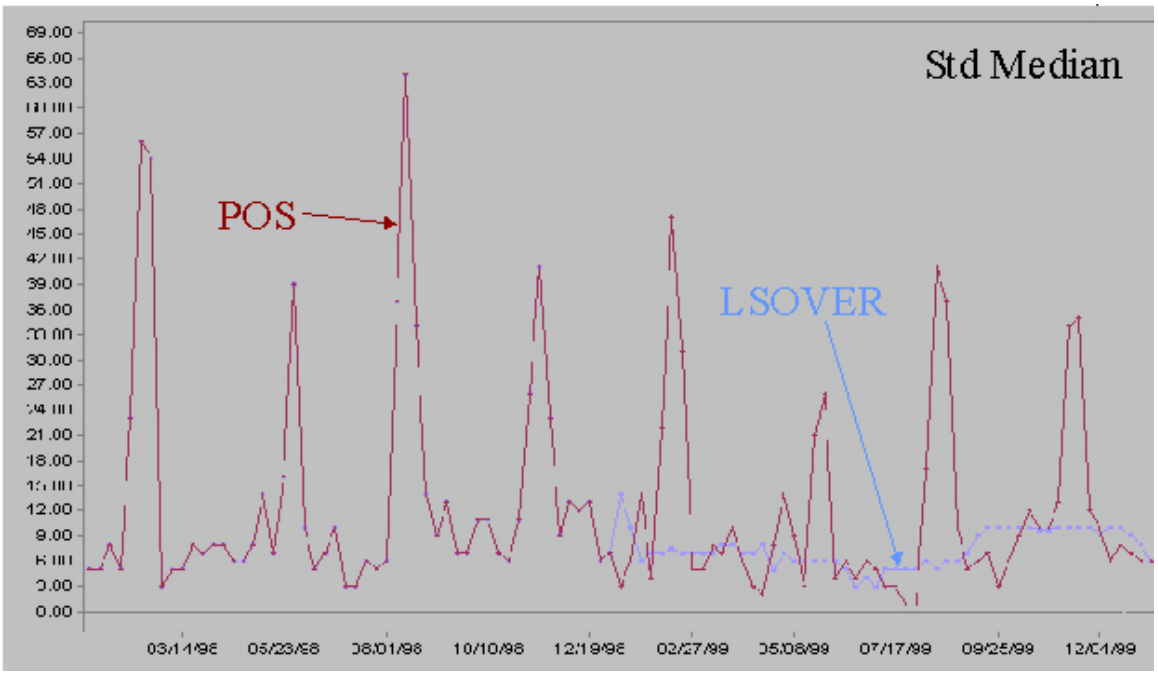
- A standard median filter implementation.
- Does not take outage info as an input.
- One optional parameter: window length.

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 Usage

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

Retek Median

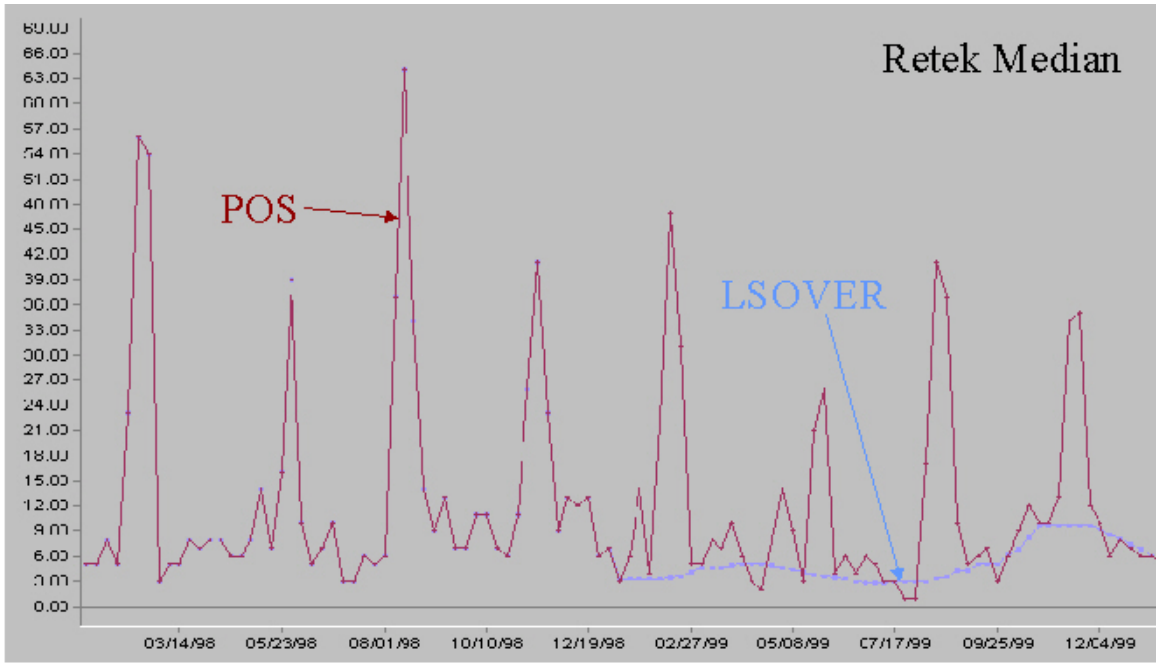
Retek Median is recommended for getting data baselines on long time ranges when promo indicators are not available.

- A sophisticated median filter that takes into consideration trends and improves side effects over the standard median filter. It makes 5 standard median filter passes.
- Does not take outage info as an input.
- 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, SMOOTH_1(t) = SRC(t) otherwise.
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 Usage

```
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.

- 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.
- 3 optional parameters:
  1. Alpha (ES parameter used to evaluate past and future velocities).
  2. Maximum number of historical points to calc past velocity.
  3. Maximum number of future points to calc 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.

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

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

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

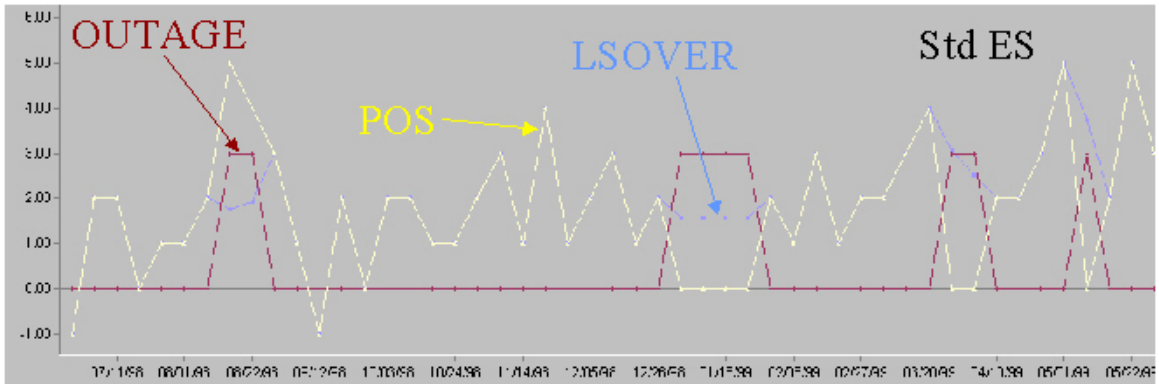
Where:

"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

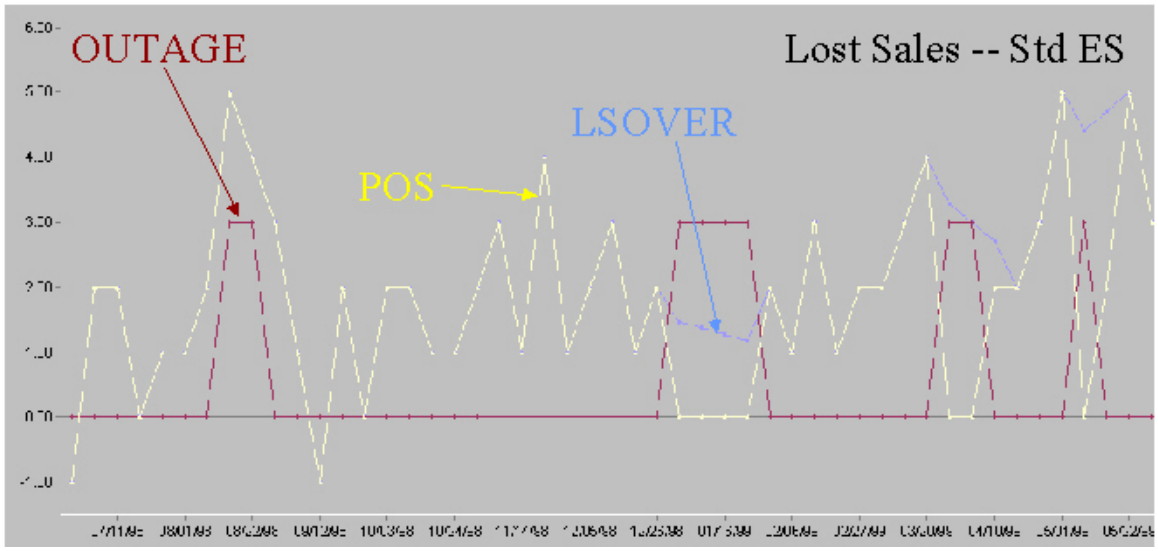
Example Usage

```
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 Usage

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, then 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.

- Does not take outage info as an input.
- Two required parameters:
  1. Approved forecast array.
  2 Approved standard deviation array of forecast.
- Four 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 already 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** σ = *MinFrcst*

ELSE IF $\sigma = 0$ **THEN**

IF *forecast(t)* < 1.0 **THEN**

σ = *forecast(t)*

ELSE $\sigma = \sqrt{\text{forecast(t)}}$

IF *SRC(t)* > *forecast(t)* + nsu*σ **THEN**

LSOVER(t) = *forecast(t)* + nsu*σ

ELSE IF *SRC(t)* < *forecast(t)* – nsl*σ **THEN**

LSOVER(t) = *forecast(t)* - nsl*σ

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

Error! Objects cannot be created from editing field codes.

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

Example Usage

```
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

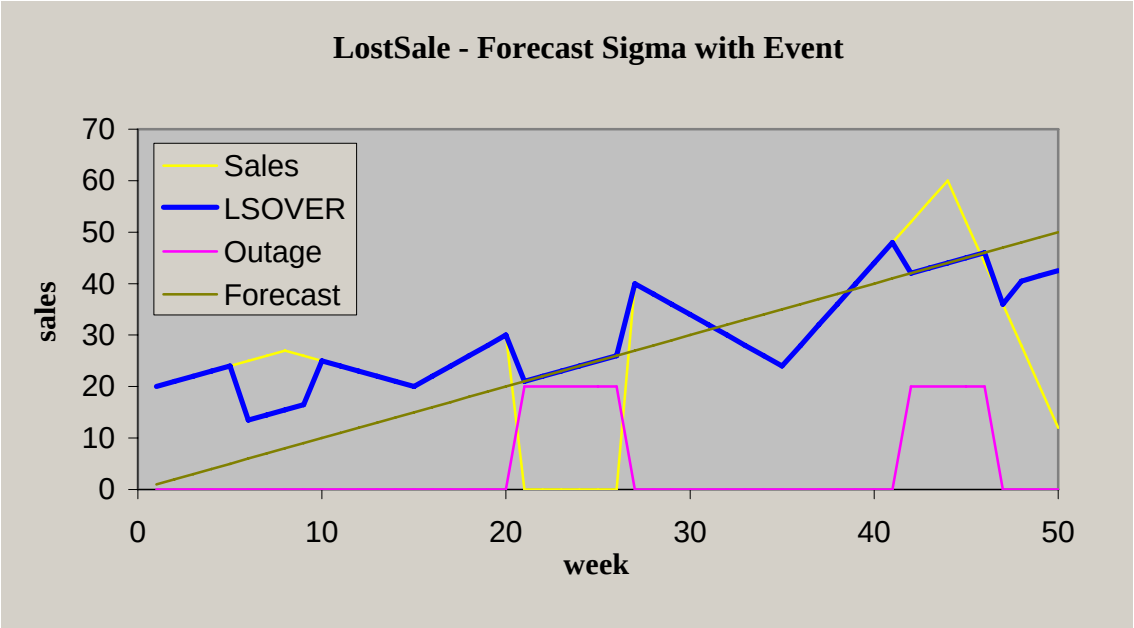
Similar to 'Frcst Sigma' and it takes an outage (i.e., event) indicator to further process.

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 the outliers calculated through NSIGMAOUT_MIN and NSIGMAOUT_MAX, they will be brought into the confidence interval bounds 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 Usage

```
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 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.

- It is a simple data copy of a given percentage of the reference data to copy from.
- May or may not take outage (i.e., event) info as an input to mask the operation.
- Two required parameters:
    1. Reference measure to copy data from.
    2. Source measure for the original data
- One 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

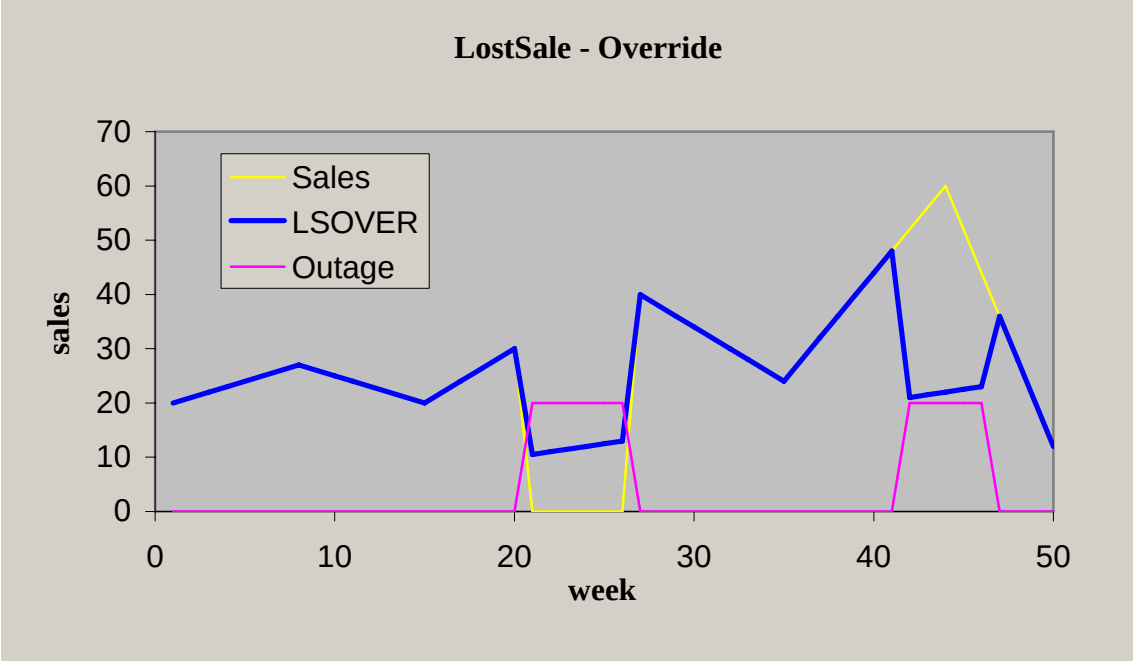
a is an adjustment percentage

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 Usage

```
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 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.

- It is a simple data increment of a given percentage of the reference data to copy from.
- May or may not take outage (i.e., event) info as an input to mask the operation.
- One required parameter:
  - Reference measure to increment by.
- One 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 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

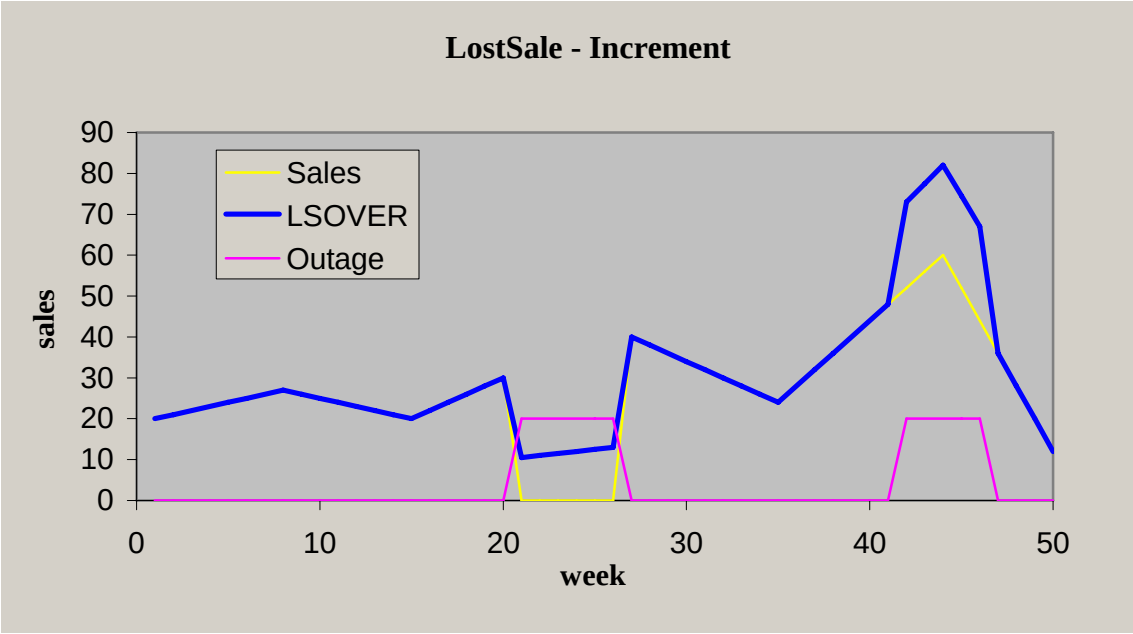
a is an adjustment percentage (can be between (–100%) and (+100%))

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

Used for canceling the effect of some former preprocessing adjustments.

- Does not take outage info 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)

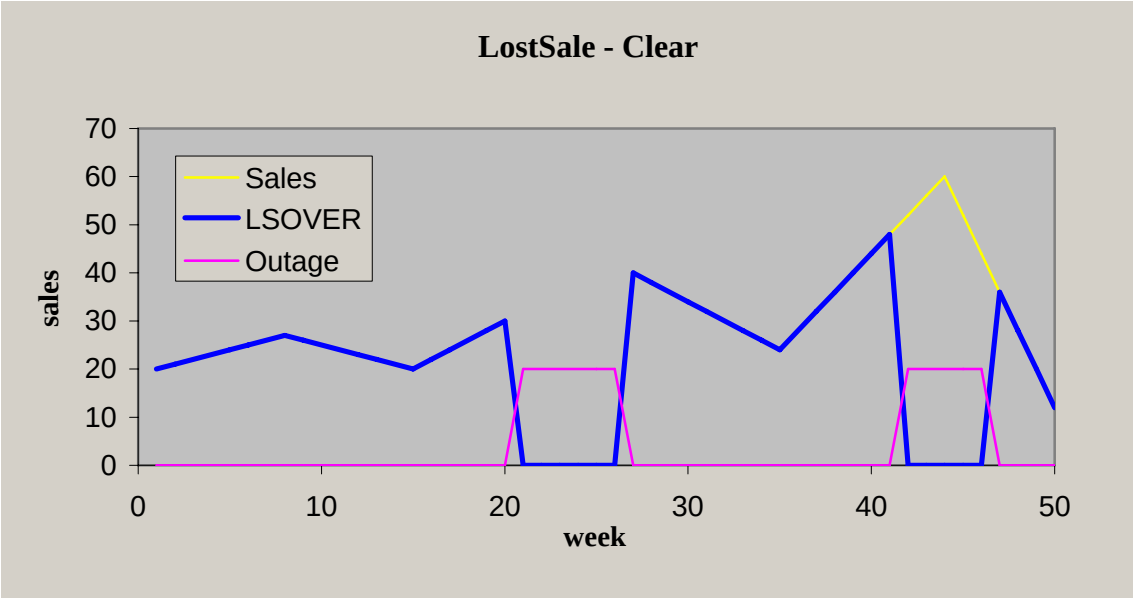
LS(t) = LSOVER_REF(t) – SRC(t)

ELSE

LSOVER(t) = 0

LS(t) = 0

Example Chart



Lost Sales – Clear with TS_Mask

Example Usage

1. Clear all:

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

2. 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)
```

LostSale Function Configuration Notes

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 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 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 can not appear on both left-hand-side and right-hand-side simultaneously, the implementation of the CLEAR filter requires the user to provide a LSOVER_REF measure, which is a duplication of the previously calculated LSOVER measure, when the user 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, user just needs 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 represents 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 if the popcount for the current time series is larger than the hard-coded threshold value, it will be processed. Otherwise, it will not be processed. The false value means the current time series will not be processed. If the TSMASK_DENSE measure is not specified, then 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, user can use a constant as input. In this case, the constant value will be applied to all the time series.

Configuration Restriction

- '_' (underscore) may not be used in any measure names and rules. The exception to this is measures and rules that are to be expanded using the RDF or Curve solution's classification scheme. The classifications in question 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

Chapter 8 – Configuring the Cluster Procedure

Clustering may be used to provide insight into how various parts of a retailer's operations can be grouped together. Typically a retailer may cluster stores over item sales to create logical groupings of stores based upon sales of particular products. This provides increased visibility to where products are selling and allows the retailer to make more accurate decisions in merchandising. Beyond this traditional use of clusters, the Cluster is flexible enough to cluster any business measure based on products, locations, time, promotions, customers, or any hierarchy configured in the solution.

Syntax Legend

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	indicates a temporary placeholder for a constant or a measure (will end with 'meas')
Italics/meas	indicates that the placeholder can be either a constant or a measure.
BoldItailics	indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N
Normal	normal text that has to be there.
Underlined And Large font	Used to identify the function name.

Syntax for Cluster()

POINTMEMBERSHIP: *MEMBERMEAS*, **CENTROID:** *CENTROIDMEAS* [, **DISTFROMCENTROID:** *DISTCENTDMEAS*, **COHESION:** *COHESIONMEAS*, **CLUSTERPORTION:** *CLPORTMEAS*, **CENTROIDTOAVG:** *C2AVGMEAS*, **CLOSESTCLUSTER:** *CLOSCLUSTMEAS*, **CLOSESTCLUSTERDIST:** *CLOSCLUSTDISTMEAS*] <-Cluster(**MEASURE:** *MEASMEAS*, **METHOD:** *METHOD*, **NUMCLUSTERS:** *NUMCLUST*, **CLUSTERHIER:** *CLUSTHIER*, **CLUSTEROVERHIER:** *CLUSTOVERHIER* [, **BYGROUPDIMS:** *BYGROUPDIM*, **AGGMETHOD:** *AGGTYPES*])

Syntax for CalculateClusterStatistics()

CENTROID: *CENTROIDMEAS*, [**DISTFROMCENTROID:** *DISTCENTDMEAS*, **COHESION:** *COHESIONMEAS*, **CLUSTERPORTION:** *CLPORTMEAS*, **CENTROIDTOAVG:** *C2AVGMEAS*, **CLOSESTCLUSTER:** *CLOSCLUSTMEAS*, **CLOSESTCLUSTERDIST:** *CLOSCLUSTDISTMEAS*] <- CalculateClusterStatistics(**MEASURE:** *MEASMEAS*, **POINTMEMBERSHIP:** *MEMBERMEAS*, **CLUSTERHIER:** *CLUSTHIER*, **CLUSTEROVERHIER:** *CLUSTOVERHIER* [, **BYGROUPDIMS:** *BYGROUPDIM*, **AGGMETHOD:** *AGGTYPE*])

Syntax for Break-Point Cluster SpecialExpression: bpcluster()

POINTMEMBERSHIP <- bpcluster(SOURCEMEASNAME, CONFIGURATIONMEASNAME, CONFIGNAME [, GROUPBYINT])

Syntax for BreakPoint Statistics SpecialExpression: bpstatistics()

CENTROID, DISTANCE <- bpstatistics(POINTMEMBERSHIP, SOURCEMEASNAME [, GROUPBYINT])

Examples

Cluster Example (minimal info):

POINTMEMBERSHIP:MEMB, CENTROID:CENT<-Cluster(MEASURE:RSAL, METHOD:”BANG”, NUMCLUSTERS:5, CLUSTERHIER:”PROD”, CLUSTEROVERHIER:”LOC”)

Cluster Statistics Example (minimal info):

CENTROID:CENT<-CalculateClusterStatistics(MEASURE:RSAL, POINTMEMBERSHIP:MEMB, CLUSTERHIER:”PROD”, CLUSTEROVERHIER:”LOC”)

BreakPoint Cluster Example (minimal info):

MEMB<-bpcluster(RSAL, GCFG, ”GCFG01”, ”CHN_PGRP”)

BreakPoint Cluster Statistics Example (minimal info):

CENTROID, DISTANCE <- bpstatistics(MEMB, RSAL, ”CHN_PGRP”)

Cluster Parameter Table

Parameter Legend

Left Hand Side – Label identifies a Left Hand Side (LHS) Parameter. In other words these are the labels identify output parameters that will appear on the left hand side of the equal sign. Otherwise the parameter is Right Hand Side (RHS) or input parameter.

Required – Label identifies required parameters.

Measure – Whether the parameter is expected to be a measure or not.

Allowed to be Multiple - Allowed to be multiple measures

Cluster and Cluster Statistics Parameters

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
POINTMEMBERSHIP	Cluster – RESULT Cluster Statistics - PARAMETER	Y	Y	N	Integer	Its intersection should be the dimension being clustered and all by group dimensions from other hierarchies. The values state which positions are assigned to which cluster index.
CENTROID	RESULT	Y	Y	N	Double	Its intersection should be the cluster dimension, the dimension being clustered over and all by group dimensions from other hierarchies. The values are the average of all points in the cluster.
DISTFROMCENTROID	RESULT	N	Y	N	Double	Its intersection should be the dimension being clustered and all by group dimensions from other hierarchies. The values are the squared Euclidean distance from that point to its centroid.

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
COHESION	RESULT	N	Y	N	Double	Its intersection should be the cluster dimension and all by group dimensions. It is a measure of how "tight" a cluster is. It will be the average of the squared Euclidean distance of each point in the cluster to the centroid.
CLUSTERPORTION	RESULT	N	Y	N	Double	Its intersection should be the cluster dimension and all by group dimensions. The value is the ratio of points in this cluster versus all clusters.
CENTROIDTOAVG	RESULT	N	Y	N	Double	Its intersection should be the cluster dimension, the dimension being clustered over and all by group dimensions from other hierarchies. The values are the ratio of the centroid to the average of all points.
CLOSESTCLUSTER	RESULT	N	Y	N	Integer	Its intersection should be the cluster dimension and all by group dimensions. The value is the nearest cluster index.
CLOSESTCLUSTERDIST	RESULT	N	Y	N	Double	Its intersection should be the cluster dimension and all by group dimensions. The values are the squared Euclidean distance from the centroid of the cluster to the centroid of the closest cluster.
MEASURE	PARAMETER	Y	Y	N	Double	The measure you are trying to cluster. It must have at least two dimensions.
METHOD	PARAMETER	Cluster – Y Cluster Statistics - N	N	N	String	Determines which clustering algorithm to use. Valid values are BANG (preferred) or KMEANS.

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
NUMCLUSTERS	PARAMETER	Cluster – Y Cluster Statistics - N	N	N	Integer	For each by group partition, what is the maximum number of clusters?
CLUSTERHIER	PARAMETER	Y	N	N	String	The hierarchy that contains the dimension to cluster. The results will give you clusters of positions in this dimension.
CLUSTEROVERHIER	PARAMETER	Y	N	N	String	The hierarchy that contains the dimension to cluster over. The algorithm uses the positions in this dimension as the co-ordinates when clustering.
BYGROUPDIMS	PARAMETER	N	N	N	String	The algorithm generates clusters one by group combination at a time. Pass the by group intersection here.
AGGMETHOD	PARAMETER	N	N	N	String	The algorithm aggregates the measure data up to the appropriate level. If AGGMETHOD is specified, it will use it. Otherwise it will use whatever is defined on the measure.

BPCluster and BPStatistics Parameters

Label	Parameter/ Result	Required	Measure	Allowed to be multiple	Type	Description
POINTMEMBERSHIP	BPCluster – RESULT BPStatistics - PARAMETER	Y	Y	N	Integer	Its intersection should be the dimension being clustered and all by group dimensions from other hierarchies. The values state which positions are assigned to which cluster index.
CENTROID	RESULT	Y	Y	N	Double	Its intersection should be the cluster dimension, the dimension being clustered over and all by group dimensions from other hierarchies. The values are the average of all points in the cluster.
DISTFROMCENTROID	RESULT	N	Y	N	Double	Its intersection should be the dimension being clustered and all by group dimensions from other hierarchies. The values are the squared Euclidean distance from that point to its centroid.
MEASURE	PARAMETER	Y	Y	N	Double	The measure you are trying to cluster. It must have at least two dimensions.
BYGROUPDIMS	PARAMETER	N	N	N	String	The algorithm generates clusters one by group combination at a time. Pass the by group intersection here.
CONFIGMEASNAME	PARAMETER	BPCluster-Y	Y	N	Double	Measure defined at Cluster/Configuration intersection. It contains the percentages for the breakpoint calculation.

Cluster Procedure Configuration Notes

The following notes are intended to serve as a guide for configuring the Cluster procedure within the configuration tools.

1. See section Syntax for the appropriate syntax for calling this procedure. Parameter labels must always be used.
2. This rule will remain red indicating that it is invalid since the RPAS JNI cannot validate it at this point in time. Therefore, there is no validation for this rule. Please refer to the Grade documentation for the appropriate input parameters and output measures.
3. Make sure the resultant measures are at the right intersection levels using the information in section 3.2 above.
4. The Cluster 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 exist 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 as in section 1.2 above.
5. You must configure/register all required input measures.
6. Be sure to create load and commit rules for the input measures. Since the RPAS JNI cannot validate the Cluster procedure call, all input measures must exist within other rules in the rule set in order for them to be available for selection in the Workbook Tool.
7. The domain must be built using the latest release of RPAS. You will get the following message in the log since the Cluster function is not validated. This is ok.

   ```
   Warning: unable to parse new expression (Unknown special expression: Cluster)
   ```

8. register Grade library after the domain build using the regfunction RPAS utility. The library must be located in the $RPAS_HOME/applib directory. The library is libClusterEngine.so. Do not specify the *lib* or *.so* part of the function name with the regfunction utility. Example:

   ```
   regfunction -d /domains/D01 -l ClusterEngine
   ```

9. Use the Mace command to execute the Cluster rule with the rule group (i.e. grade_batch).

   ```
   mace -d /domains/D01 -run -group grade_batch
   ```

Cluster Statistics Procedure Configuration Notes

The following notes are intended to serve as a guide for configuring the CalculateClusterStatistics procedure within the configuration tools.

1. See Syntax for the appropriate syntax for calling this procedure. Parameter labels must always be used.
2. This rule will remain red indicating that it is invalid since the RPAS JNI cannot validate it at this point in time. Therefore, there is no validation for this rule. Please refer to the Grade documentation for the appropriate input parameters and output measures.
3. Make sure the resultant measures are at the right intersection levels using the information in section 3.2 above.
4. The CalculateClusterStatistics 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 exist 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 as in section 1.3 above.
5. You must configure/register all required input measures.
6. Be sure to create load and commit rules for the input measures. Since the RPAS JNI cannot validate the Cluster procedure call, all input measures must exist within other rules in the rule set in order for them to be available for selection in the Workbook Tool.
7. The domain must be built using the latest release of RPAS. You will get the following message in the log since the Cluster function is not validated. This is ok.

   ```
   Warning: unable to parse new expression (Unknown special expression:
   CalculateClusterStatistics)
   ```

8. Must register Grade library after the domain build using the regfunction RPAS utility, if it hasn't been done already. The library must be located in the $RPAS_HOME/applib directory. The library is libClusterEngine.so. Do not specify the *lib* or *.so* part of the function name with the regfunction utility. Example:

   ```
   regfunction -d /domains/D01 -l ClusterEngine
   ```

9. Use the Mace command to execute the Cluster rule with the rule group (i.e. grade_batch).

   ```
   mace -d /domains/D01 -run -group grade_batch
   ```

Chapter 9 – Configuring the ASOSpace Function

ASOSpace function utilizes a specialized Dynamic Programming method (Karigma) to generate optimal Assortment plan that maximize total profit constraint by total store space, based on each subcategory's space – profit histogram.

Syntax Legend

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	indicates a temporary placeholder for a constant or a measure (will end with 'meas')
Italics/meas	indicates that the placeholder can be either a constant or a measure.
BoldItailics	indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N
Normal	normal text that has to be there.
Underlined And Large font	Used to identify the function name.

Syntax

ProfitSolutionMeas,
PlanoSolutionMeas,
HistProfitMeas,
HistCostMeas
<- ASOSpace(
MaxCapacityMeas,
ConstraintTypeMeas,
IncludeMeas,
MandatoryMeas,
MinSpaceMeas,
MaxSpaceMeas,
ProfitVectorMeas,
PogLengthVectorMeas)

Example Usage

```
rule MOcalc2 { "MgOtExpProfR.level([clnd].[phse]),
MgOtSpaceX.level([clnd].[phse]), MgOtHistProfR.level([clnd].[phse]),
MgOtHistCostX.level([clnd].[phse])
<- ASOSpace(
MgWpMaxCapX.level([clnd].[phse]), MgWpConsTypSl.level([clnd].[phse]),
MgWpIncludeB.level([clnd].[phse]),MgWpManCatB.level([clnd].[phse]),
MgWpMinSpX.level([clnd].[phse]), MgWpMaxSpX.level([clnd].[phse]),
AdWpPogPftR.level([clnd].[phse]), AdWpPogLX.level([clnd].[phse]))" };

rulegroup MOcalc2 {
rule MOcalc2;
};
```

Configuration Parameters and Rules

Input Parameters

Parameter	Description	Intersection	Usage Comment
MaxCapacityMeas	Maximum capacity available	STR/PHSE	
ConstraintTypeMeas	Constraint Mode	STR/PHSE	
IncludeMeas	Include flag	SCAT/STR/PHSE	
ManadatoryMeas	Mandatory flag	SCAT/STR/PHASE	
MinSpaceMeas	Minimum space required for each Subcategory	SCAT/STR/PHSE	
MaxSpaceMeas	Maximum space allowed for each Subcategory	SCAT/STR/PHSE	
ProfitVectorMeas	Per SCAT/STR/PHSE, this is a vector of profit indexed by planogram.	SCAT/STR/PHSE/PLNG	
PogLengthVectorMeas	Per SCAT/STR/PHSE, this is a vector of size indexed by planogram	SCAT/STR/PHSE/PLNG	

Output Parameters

Parameter	Description	Intersection	Usage Comment
ProfitSolutionMeas	Profit Solution measure	SCAT/STR/PHSE	
PlanoSolutionMeas	Plano Solution measure	SCAT/STR/PHSE	
HistProfitMeas	Histogram Profit data points	STR/PHSE/LNUM	
HistCostMeas	Histogram Cost data points	STR/PHSE/LNUM	

Chapter 10 – Configuring the ASOAssort Function

ASOAssort function utilizes a specialized Dynamic Programming method (Karigma) to generate optimal Assortment plan that maximize total profit constraint by total shelf space, based on each item's size, cost, retail price, demand, and store's operating time and replenish strategy.

Although the target of the optimization is for each item, the ASOAssort function divide the total problem into two levels and solve it level by level to improve performance. The first level is called 'Collection' in Assortment Planning terminology, which is usually implemented on the 'STYL' dimension. The second level is called 'Subcategory', which is usually implemented on the 'SCAT' dimension. For this reason, the ASOAssort requires some duplicated inputs on multiple levels, like Include Flag and Mandatory Flag. For Calendar, the time period used for Assort Optimization is usually called a 'Phase', which is implemented on 'PHSE' dimension.

For input arguments to the ASOAssort function like Cost, Retail Prices etc. are in the unit of Dollars. Demand is in the unit of Units. For Size arguments, the unit is usually Planogram size, which will be converted to real size metric like Inches or Centimeters based on UOM inputs.

The output of ASOAssort is the optimal planogram found by the Karigma algorithm. Both the solutions at the Subcategory level and Collection level are returned, together with the profit/cost table at each level, and Stock Out unit/dollars table, histogram table at the item level.

Syntax Legend

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	indicates a temporary placeholder for a constant or a measure (will end with 'meas')
Italics/meas	indicates that the placeholder can be either a constant or a measure.
BoldItailics	indicates a numeric placeholder for the dynamic portion of a label. Usually a number from 1 to N
Normal	normal text that has to be there.
Underlined And Large font	Used to identify the function name.

Syntax

SubcategoryExpectedProfileTableMeas,
SubcategoryRecommendedCostTableMeas,
SubcategoryRecommendedActualCostTableMeas,
CollectionRecommendedProfileTableMeas,
ColectionRecommendedCostTableMeas,
ItemRecommendedFacingsTableMeas,
ItemExpectedProfileTableMeas,
ItemExpectedCostTableMeas,
ItemExpectedBuyQuantityTableMeas,
ItemStockOutUnitTableMeas,
ItemStockOutDollarsTableMeas,
HistogramProfitDataPointsMeas,
HistogramCostDataPointsMeas
<- **ASOAssort(**
IncludeMeas,
MininumNumberOfFacingMeas,
MaximumNumberOfFacingMeas,
MinimumShelfPresenceMeas,
FacingCapacityMeas,
RegularDemandMeas,
MaximumDemandMeas,
MaxLeadTimeMeas,
SafetyStockMultiplierMeas,
MandatoryItemMeas,
ItemCostMeas,
OperatingHoursMeas,
StockClerkSuccessRate,
StockClerkCycleTime,
ItemRetailPriceMeas,
ItemWidthMeas,
ConstraintTypeMeas,
NumberOfShelvesMeas,
POGLengthUOM,
ItemWidthUOM,
StepSizeMeas,
ToleranceMeas,
StepSizeToleranceMeas,
StepSizeUOM,
ToleranceUOM,
IncludeCollectionMeas,
MandatoryCollectionMeas,
PlanoSize)

Example Usage

rule AOcalc2 { "PlOtScatProfR.level([clnd].[phse]), PlOtScatCostX.level([clnd].[phse]), PlOtSctActCstX.level([clnd].[phse]), PlOtCollProfR.level([clnd].[phse]), PlOtCollCostX.level([clnd].[phse]), PlOtItmRecFacX.level([clnd].[phse]), PlOtItmProfR.level([clnd].[phse]), PlOtItmSolCstX.level([clnd].[phse]), PlOtItmBuyQtyX.level([clnd].[phse]), PlOtItmStkOutU.level([clnd].[phse]), PlOtItmStkOutR.level([clnd].[phse]), PlOtHistProfR.level([clnd].[phse]), PlOtHistCostX.level([clnd].[phse])

<- ASOAssort(

PlWpIncludeB.levek([clnd].[phse]), PlWpMinFacX.level([clnd].[phse]), PlWpMaxFacX.level([clnd].[phse]), PlWpRstkTrgU.level([clnd].[phse]), PlWpFacCapX.level([clnd].[phse]), PlFcDemandU.level([clnd].[phse]), PlWpDemandMaxU.level([clnd].[phse]), PlWpMaxLTX.level([clnd].[phse]), PlWpSfySkMultX.level([clnd].[phse]), PlWpManItmB.level([clnd].[phse]), PlWpItmCstC.level([clnd].[phse]), AdWpOpHrsX.level([clnd].[phse]), AdWpStkClSRtX, AdWpStkClkCTX, PlWpItmRetR.level([clnd].[phse]), PlWpItmWX.level([clnd].[phse]), PlWpConsTypSl.level([clnd].[phse]), PlWpNmShlvsX.level([clnd].[phse]), PlWpPogUOMSl, PlWpItmUOMSl, PlWpStepSizeX.level([clnd].[phse]), PlWpToleranceX.level([clnd].[phse]), PlWpScatTolX.level([clnd].[phse]), PlWpStpSzUOMSl, PlWpTolUOMSl, PlWpInclCollB.level([clnd].[phse]), PlWpManCollB.level([clnd].[phse]), PlWpPogOptsSl.level([clnd].[phse]))" };

rulegroup AOcalc2 { rule AOcalc2;};

Configuration Parameters and Rules

Input Parameters

Parameter	Description	Intersection	Usage Comment
IncludeMeasMeas	Indicate whether an item should be included in the calculation	SKU/STR/PHSE	
MinimumNumberOfFacingsMeas	Minimum Number of Facings that must be included in the planogram.	SKU/STR/PHSE	
MaximumNumberOfFacingsMeas	Maximum Number of Facings that can be included in the planogram	SKU/STR/PHSE	
MinimumShelfPresenceMeas	Minimum Shelf Presence in Units	SKU/STR/PHSE	
FacingCapacityMeas	The number of units that can sit on a shelf depth wise.	SKU/STR/PHSE	
RegularDemandMeas	Regular Demand Rate, number of units in demand per PHSE	SKU/STR/PHSE	
MaximumDemandMeas	Maximum Demand over multiple periods. It will used to calculate the MaximumLeadTimeDemand (MaximumDemand x MaximumLeadTime)	SKU/STR/PHSE	
SafetyStockMultiplierMeas	Percentage of Safety Stock	SKU/STR/PHSE	
MandatoryMeas	Mandatory Include Flag measure, whether or not the item is mandatory in the final mix.	SKU/STR/PHSE	
ItemCostMeas	The cost of the item to the retailer	SKU/STR/PHSE	
OperatingHoursPerPeriodMeas	The number of Hours that the store is open for business, per optimization period (PHSE).	STR/PHSE	

Parameter	Description	Intersection	Usage Comment
StockClerkSuccessRateMeas	Defined as the probability that the stock clerk will successfully replenish the shelf when the supply falls below the Minimum Shelf Presence quantity.	STR	
StockClerkCycleTimeMeas	Number of hours between the stock clerk's round	STR	
ItemRetailPrice	Item cost to the consumer	SKU/STR/PHSE	
ItemWidthMeas	The width of the item	SKU/STR/PHSE	
ConstraintModeMeas	Constraint Type, L for LESSEQUAL, E for EQUAL, G for GREATEREQUAL, A for APPROXIMATEEQUAL	STR/PHSE	
NumberOfShelvesMeas	Number of Shelves can be used	STR/PHSE	
POGLengthUOM	The unit of measure the Planogram length is in.	SCALAR	
ItemWidthUOM	The unit of measure the ItemWidthMeas is in	SCALAR	
StepSizeMeas	The incremental size that will be considered for each sub-category.	STR/WEEK	Default to 0
ToleranceMeas	Tolerance in Karigma algorithm	STR/PHSE	
StepSizeToleranceMeas	Tolerance for the StepSize in inches.	STR/PHSE	Default to 0
StepSizeUOM	The unit of measure the StepSizeMeas is in	SCALAR	
ToleranceUOM	The unit of measure the ToleranceMeas is in	SCALAR	
IncludeCollectionMeas	Indicate whether a STYL is included in the calculation	STYL/STR/PHSE	
MandatoryCollectionMeas	Indicate whether a STYL is mandatory to show up in the final mix	STYL/STR/PHSE	
PlanoSizeMeas	The planogram length	STR/PHSE	

Output Parameters

Parameter	Description	Intersection	Usage Comment
SubcategoryExpectedProfileTableMeas	Subcategory solution expected profile table	SCAT/STR/PHSE	
SubcategoryRecommendedCostTableMeas	Subcategory solution recommended cost table	SCAT/STR/PHSE	
SubcategoryRecommendedActualCostTableMeas	Subcategory solution recommended actual cost table	SCAT/STR/PHSE	
CollectionRecommendedProfileTableMeas	Collection solution recommended profile table	STYL/STR/PHSE	
CollectionRecommendedCostTableMeas	Collection solution recommended cost table	STYL/STR/PHSE	
ItemRecommendedFacingsTableMeas	Item solution recommended facing table	SKU/STR/PHSE	
ItemExpectedProfileTableMeas	Item solution expected profit table	SKU/STR/PHSE	
ItemExpectedCostTableMeas	Item solution expected cost table	SKU/STR/PHSE	
ItemExpectedBuyQuantityTableMeas	Item solution expected buy quantity table	SKU/STR/PHSE	
ItemStockOutUnitTableMeas	Item solution stock out units table	SKU/STR/PHSE	
ItemStockOutDollarsTableMeas	Item solution stock out dollars table	SKU/STR/PHSE	
HistogramProfitDataPointsMeas	Histogram profit data points	STR/PHSE/LNUM	
HistogramCostDataPointsMeas	Histogram cost data points	STR/PHSE/LNUM	