

**Oracle[®] Retail Predictive Application
Server
Solution Extension Configuration
Guide
Release 11.2.2
January 2006**

Copyright © 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, ProfitLogic and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	v
Customer Support	v
1 Introduction	1
Overview.....	1
2 Configuring the Forecast Procedure	3
Overview.....	3
Syntax	4
Examples	4
Forecast Parameter Table.....	6
Parameter/Model Dependencies	12
Forecast Procedure Configuration Notes	12
Function Registration Requirements.....	13
3 Configuring the LostSale Function	15
Overview.....	15
Syntax	16
Parameter Table Legend	17
LostSale Filtering Method Descriptions	25
Standard Median.....	25
Retek Median.....	26
Standard Exponential Smoothing	27
Lost Sales – Standard Exponential Smoothing	28
Forecast Sigma.....	29
Forecast Sigma Event	30
Override.....	32
Increment.....	33
Clear	35
LostSale Function Configuration Notes.....	36
Configuration Restriction.....	37
Function Registration Requirements.....	37
4 Configuring the Cluster Procedure	39
Syntax for Cluster()	39
Syntax for CalculateClusterStatistics()	39
Syntax for Break-Point Cluster SpecialExpression: bpcluster()	40
Syntax for BreakPoint Statistics SpecialExpression: bpstatistics().....	40
Examples:	40
Cluster Example (minimal info):	40
Cluster Statistics Example (minimal info):	40
BreakPoint Cluster Example (minimal info):	40
BreakPoint Cluster Statistics Example (minimal info):	40
Cluster Parameter Table.....	41
Parameter Legend	41
BPCluster and BPStatistics Parameters	44
Cluster Procedure Configuration Notes	45
Function Registration Requirements.....	45

- 5 Configuring the ASOSpace Function 47**
 - Syntax Legend47
 - Syntax47
 - Configuration Parameters and Rules.....49
 - Input Parameters49
 - Output Parameters49
 - Function Registration Requirements.....49

- 6 Configuring the ASOAssort Function..... 51**
 - Syntax Legend51
 - Syntax52
 - Example Usage53
 - Configuration Parameters and Rules.....54
 - Input Parameters54
 - Output Parameters56
 - Function Registration Requirements.....56

Customer Support

- <https://metalink.oracle.com>

When contacting Customer Support, please provide:

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

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. The following Solution Extensions are currently available for configuration within the RPAS 11.2 platform:

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

The following chapters provide instructions on the necessary steps to configure each of these Solution Extensions.

Configuring the Forecast Procedure

Overview

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

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

The following section contains the specifications and syntax for configuring the Forecast procedure.

Note: The syntax is slightly different than the standard RPAS functions and procedures that are described in the RPAS 11.2 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 are a complete set.
Bold	Labels.
<i>Italics</i>	Italics indicate a temporary placeholder for a constant or a measure (will end with ‘meas’).
<i>Italics/meas</i>	This indicates that the placeholder can be either a constant or a measure.

Indicator	Definition
<i>BoldItalics</i>	This indicates a numeric placeholder for the dynamic portion of a label, which is usually a number from 1 to N.
Normal	Normal text that has to be there.
<u>Underlined And</u> <u>Large font</u>	This is 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*, TRENDAMP: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*, TRENDAMP:0.5, HISTSTART:H_STARTDATE1,  
FRCSTSTART:F_STARTDATE1)
```

Forecast Parameter Table

Left Hand Side – The label identifies a Left Hand Side (LHS) Parameter. 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 – This label identifies required parameters.

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

Allowed to be Multiple – This indicates that measures are 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
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

Label	Parameter/ Result	Required	Measure	Allowed to be Multiple	Type	Description
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, which is used only for customers that require special processing
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

Label	Parameter/ Result	Required	Measure	Allowed to be Multiple	Type	Description
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 clamped or values are dropped and regression is re-run
MASK	PARAMETER	Y	Y	N	bool	Array that identifies what forecast method is used for each time series; 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

Label	Parameter/ Result	Required	Measure	Allowed to be Multiple	Type	Description
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
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.

Label	Parameter/ Result	Required	Measure	Allowed to be Multiple	Type	Description
PROFILE	PARAMETER	N	Y	N	double	Seasonal profile measure
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
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

Label	Parameter/ Result	Required	Measure	Allowed to be Multiple	Type	Description
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
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; Useful for debugging

Forecast Method / Model List

Model	Numeric Value
AUTO ES	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(...)
```
- Refer to the Forecast documentation for the appropriate input parameters and output measures.
- The resultant measure (that is, frctest) 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,startdatemeas: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)
 1. range value – see Forecast Method list
 2. range syntax: 0(No Forecast), 1(Simple), 2(Holt), 3(Croston)
 3. 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.
- 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. Within the client, it then automatically defaults to that value (such as Holt). If you specify the naval outside of the range, 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.

Function Registration Requirements

The following functions are required to be registered in any domain(s) that will use the Forecast Solution Extension:

- AppFunctions
- RdfFunctions

Configuring the LostSale Function

Overview

The purpose of Oracle Retail's LostSale module, which may also be referred to as "Preprocessing," is to correct past data points that represent unusual sales values that are not representative of a general demand pattern. Such corrections may be necessary when an item is out of stock and cannot be sold, which usually results in low sales. 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 that are 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 that special expressions use "<->" instead of "=" in the expression.

Note: The syntax is slightly different than the standard RPAS functions and procedures that are described in the RPAS 11.2 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 are a complete set.
Bold	Labels.
<i>Italics</i>	Italics indicate a temporary placeholder for a constant or a measure (will end with 'meas').
<i>Italics/meas</i>	This indicates that the placeholder can be either a constant or a measure.
<i>BoldItalics</i>	This indicates a numeric placeholder for the dynamic portion of a label. It is usually a number from 1 to N.
Normal	Normal text that has to be there.
<u>Underlined And Large font</u>	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 it 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

Group	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
	NPTS	Input	Y	Y	N	Double	N/A
	MIN_TSALERT	Input	N	Y	N	Double	N/A

Configuring the LostSale Function

Group	Label	Input Or Output	Required By All Methods	Measure	Has calendar dimension	Type	Default Value
	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
	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	

Group	Label	Input Or Output	Required By All Methods	Measure	Has calendar dimension	Type	Default Value
	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	REFERENCE	Reference will be used for source data substitution
	DEVIATION	Standard deviation for confidence interval calculation by Forecast Sigma filters

Group	Label	Description
Parameters	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	Required method-specific input parameters	Optional method-specific input parameters
MEDIAN5	Retek Median	0		WINDOW1, 2, 3, 4, 5
MEDIAN1	Std Median	1		WINDOW
OVERRIDE	Override	2	REFERENCE	DELTA
INCREMENT	Increment	3	REFERENCE	DELTA
ES_LT	Std ES	4	OUTAGE	ALPHA, NPAS, NFU
LS_ES_LT	Lost Sales -- Std ES	9	OUTAGE	ALPHA, NPAS, NFU
FRCST_SIGMA	Forecast and stddev algo	14	REFERENCE, DEVIATION	NSIGMA_MAX, NSIGMA_MIN, FRCST_MIN, HIST_MIN_FS
FRCST_SIGMA_EVENT	Forecast and stddev algo with event	15	OUTAGE, REFERENCE, DEVIATION	NSIGMAOUT_MAX, NSIGMAOUT_MIN, NSIGMAADJ_MAX, NSIGMAADJ_MIN, FRCST_MIN, HIST_MIN_FS

Model	Comment	Numeric Value	Required method-specific input parameters	Optional method-specific input parameters
MARK_DOWN	Markdown removal -- interpolation on Mdarea	16		PRICE, INVENTORY, HIST_MIN_MD,
CLEAR	Clear (clear specified result measures)	17		TSMASK_DENSE, LSOVER_REF
CLEAR_ALERT	Clear Alert measure	18		
NO_FILT	No Filtering (does not do anything)	19		
FLP_CALC	First and last populated locations calculation	20		
LS_MEDIAN5_EVENT	Lost Sales - Retek Median with Event	21	OUTAGE	WINDOW1, 2, 3, 4, 5

Note: LS_MEDIAN5_EVENT is a combination method. Retek Median is run first, and 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.

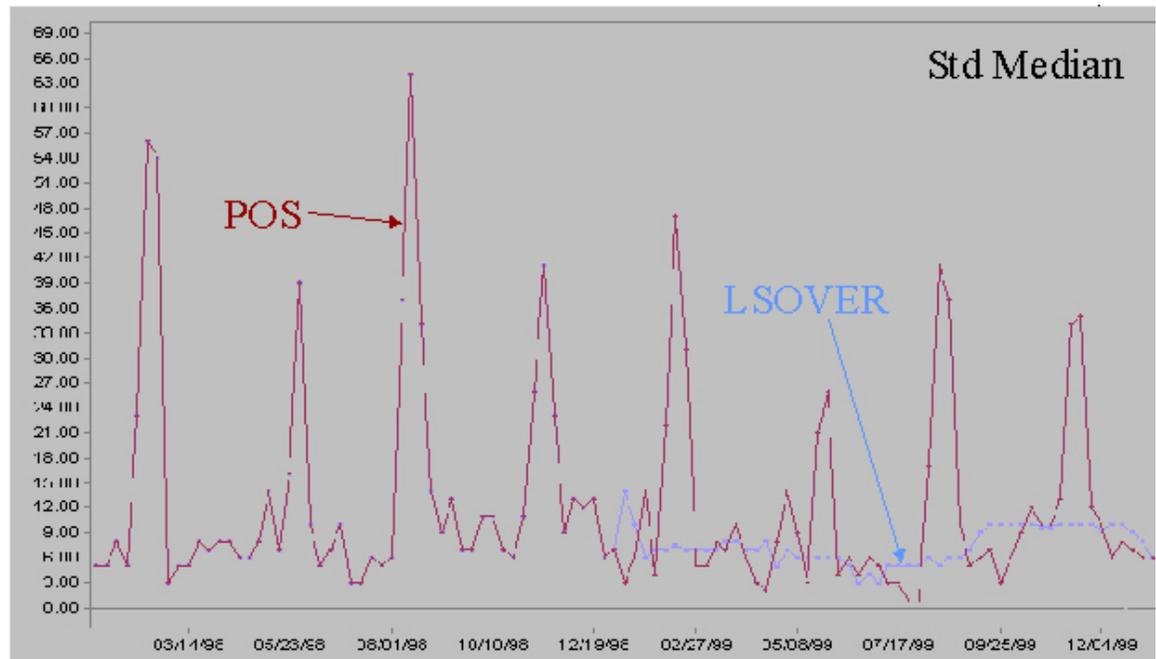
- A standard median filter implementation
- Does not take outage info as an input
- One optional parameter: window length

Mathematical Formulation

$LSOVER(t) = \text{median value of SRC over } [t-\text{window}/2, t+\text{window}/2]$,

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

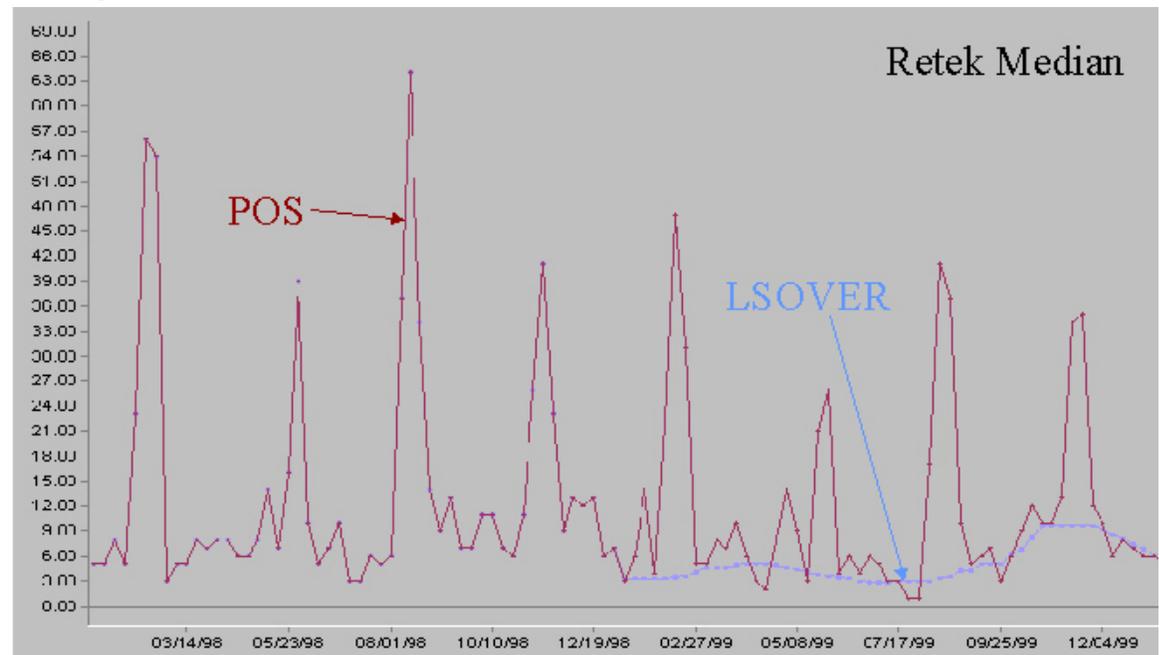
Retek Median is recommended for getting data baselines on long time ranges when promo indicators are not available.

- A sophisticated median filter that takes trends into consideration and improves side effects over the standard median filter. It makes five 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. Otherwise, $SMOOTH_1(t) = SRC(t)$.
3. To prepare for the fourth pass, the one-step difference of $SMOOTH_1(t)$ is calculated. That is, $DIFF_2(t) = SMOOTH_1(t) - SMOOTH_1(t-1)$. An average version of $DIFF_2(t)$ is calculated using the standard median filter. The result is denoted by $AVG_DIFF_2(t)$. The result of the fourth pass is $SMOOTH_2(t) = SMOOTH_2(t-1) + AVG_DIFF_2(t)$.
4. Finally, $LSOVER(t)$ is the result of applying the standard median filter to $SMOOTH_2(t)$.

Example Chart



Retek Median with default parameters

Example Usage

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

Mathematical Formulation

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

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

$$Future_Velocity = \frac{\sum_{i=1}^{nf} (1-\alpha)^{i-1} * SRC(t_i + i)}{\sum_{i=1}^{nf} (1-\alpha)^{i-1}}$$

$$LISOVER(t) = Past_Velocity + \frac{Future_Velocity - Past_Velocity}{t_i - t_f + 2} * (t - t_f + 1), \forall t \in [t_f, t_i]$$

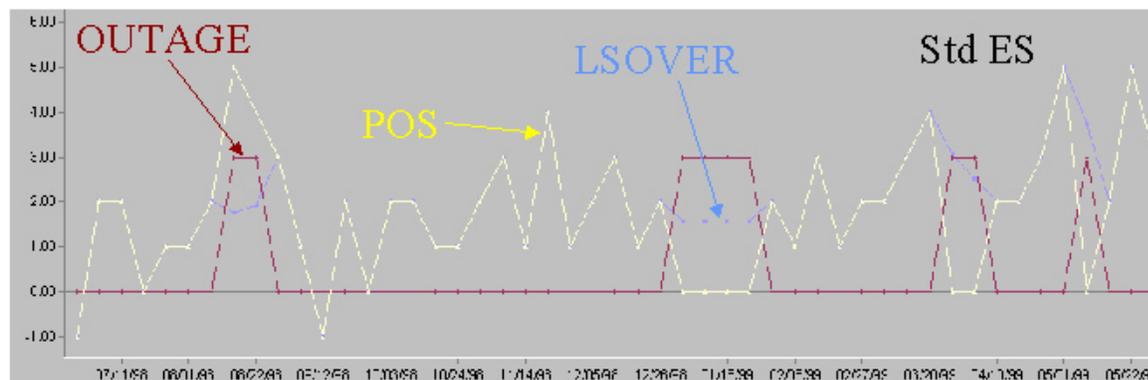
Where:

“ 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

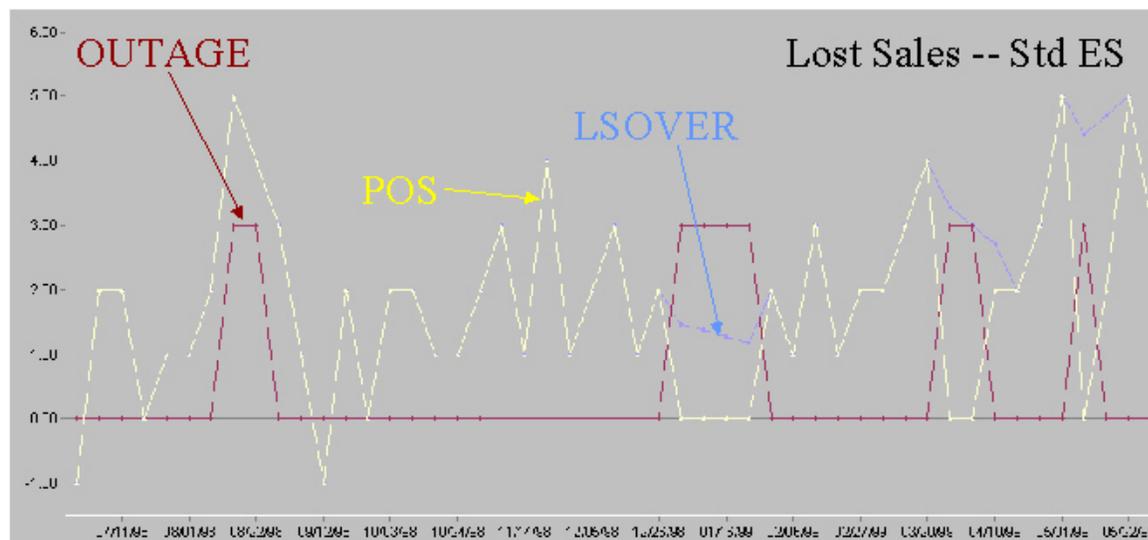
Example Usage

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

IF # historical points < MinHist THEN

$LSOVER(t) = SRC(t)$

ELSE IF forecast(t) < MinFrcst THEN

$forecast(t) = MinFrcst$ **AND** $\sigma = MinFrcst$

ELSE IF $\sigma = 0$ THEN

IF forecast(t) < 1.0 THEN

$\sigma = forecast(t)$

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

IF SRC(t) > forecast(t) + nsu* σ THEN

$LSOVER(t) = forecast(t) + nsu*\sigma$

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

$LSOVER(t) = forecast(t) - nsl*\sigma$

ELSE $LSOVER(t) = SRC(t)$

Where:

“nsu” is the number of std. deviations for upper bound.

“nsl” is the number of std. deviations for lower bound.

“*MinFrcst*” is the forecast lower bound.

“*MinHist*” is the minimum item history (# points) required for filtering.

Example Chart

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

Example Usage

```
LSOVER:LSOVER1, LS:LS1, TSALEERT:TSALEERT1 =  
preprocess(SRC:POS, METHODID:mthid, LSTODAY:TODAY1,  
NPTS:npts, REFERENCE:forecast1, DEVIATION:dev1,  
NSIGMA_MIN:nsigma_min, NSIGMA_MAX:nsigma_max,  
FRCST_MIN:0.1, HIST_MIN_FS:hist_min_fs)
```

Forecast Sigma Event

This is similar to ‘Frcst Sigma,’ and it takes an outage (for instance, event) indicator to further process.

Mathematical Formulation

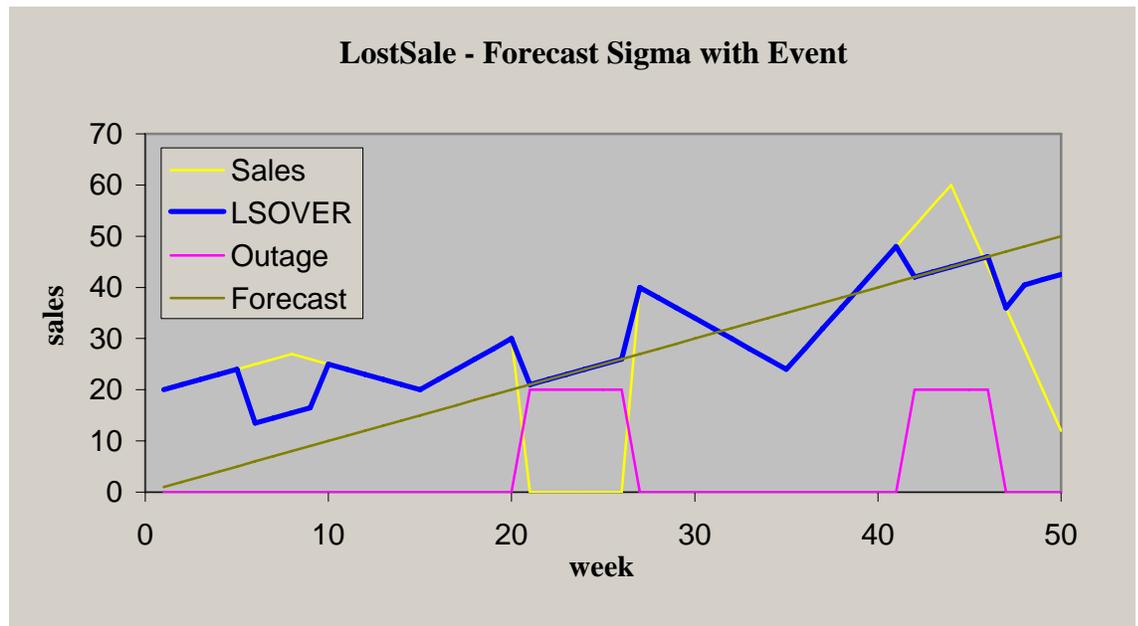
When the outage/event mask is ON:

$LSOVER(t) = \text{forecast}(t)$

When the outage/event mask is OFF:

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

Example Chart



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

`minFrcst = 0.1` and `minHist = 5` weeks

Example 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 that is adjusted by the adjustment percentage according to the mask. It is recommended for filling data gaps when an existing reference measure exists as a default value.

- It is a simple data copy of a given percentage of the reference data to copy from.
- This may or may not take outage (for instance, event) info as an input to mask the operation.
- 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

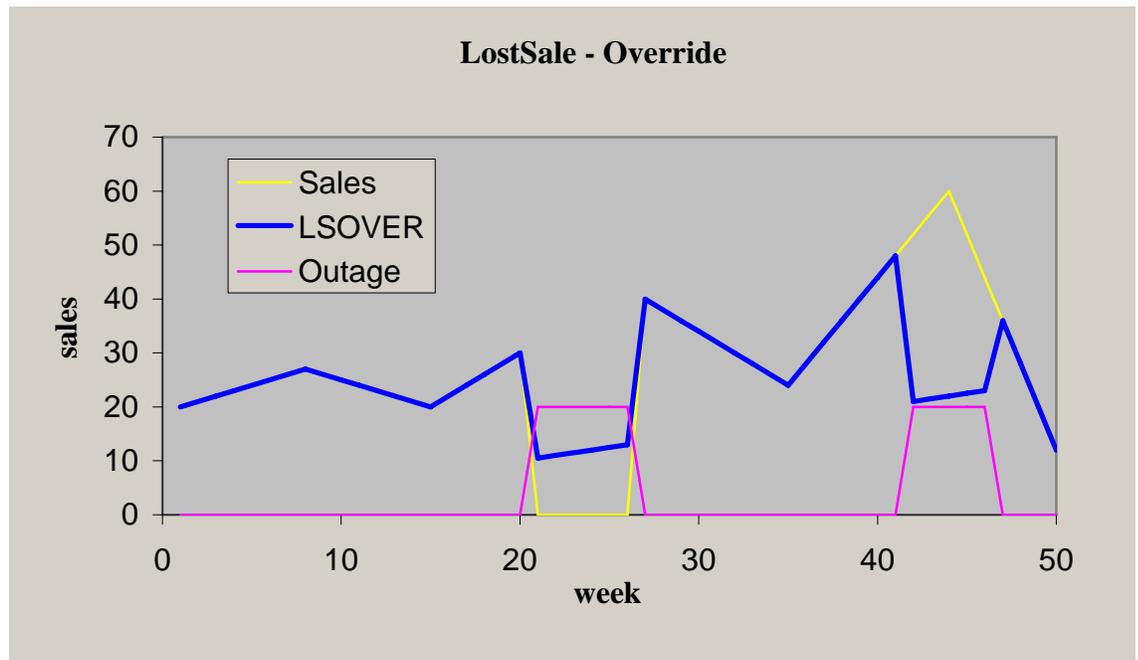
a is an adjustment percentage

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

- It is a simple data increment of a given percentage of the reference data to copy from.
- It may or may not take outage (for instance, event) info as an input to mask the operation.
- 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, which is adjusted by the adjustment percentage according to the mask.

Let:

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

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

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

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

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

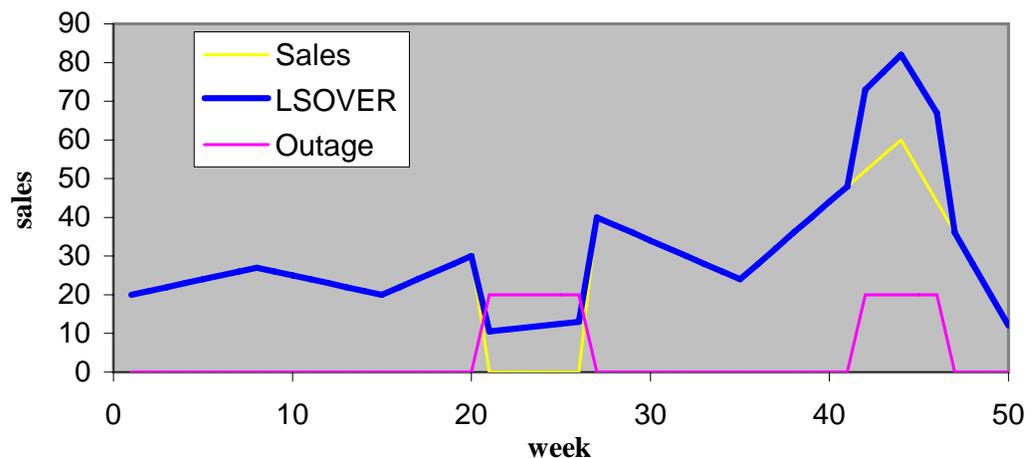
The result of the reduction method is:

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

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

Example Chart

LostSale - Increment



Lost Sales – Increment with delta = 0.5

Example Usage

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

Clear

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

- 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**

$LSOVER(t) = LSOVER_REF(t)$

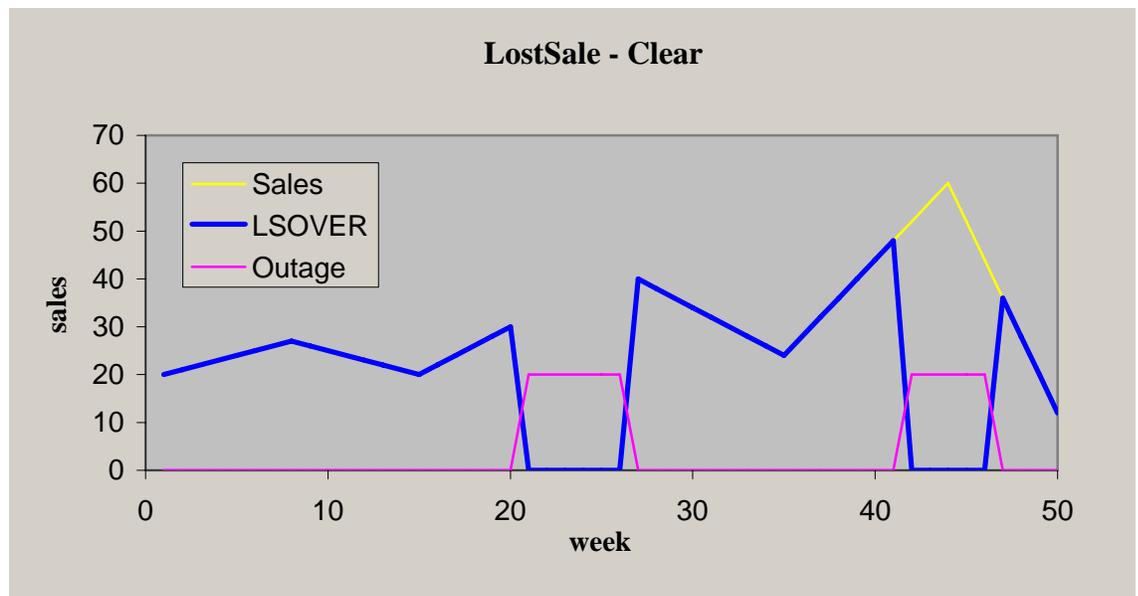
$LS(t) = LSOVER_REF(t) - SRC(t)$

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 of the input measure parameters must be configured and registered before the function call. The input parameters must be comma-separated in the function call as in the example.
3. The LostSale function library must be registered after the domain build by using the regfunction RPAS utility.
4. The LostSale function required all the input and output measures using the same intersections. Mixed input/output measure intersections should be aligned to the same calculation intersection with other RPAS function/procedure before calling the LostSale function. The same procedure can be carried out to the resultant measures to spread or aggregate them to the designated intersections.
5. Because of the limitation that the same measure cannot simultaneously appear on both left-hand-side and right-hand-side, the implementation of the CLEAR filter requires the user to provide a LSOVER_REF measure (a duplication of the previously calculated LSOVER measure) when you try to retain the results on certain time series but clear the others by providing a mask measure (TSMASK_DENSE). The LSOVER_REF is not required when the results for all the time series need to be cleared.
6. The LSTODAY measure is used to specify the end date for the filter processing. It only accepts the index number for the end date along the calendar dimension as valid input. If it is desired that the string position name to be used for the end date specification, the available RPAS time dimension translation function “index” can be used to do the name-index conversion before calling the LostSale function.
7. The LSTODAY input parameter is designed to be a measure rather than a constant to provide more flexibility. Current implementation only allows one global LSTODAY index value to be used in processing all the time series. To specify the end date, you just need to populate its value for the first time series, and this index will be applied to all the other time series.

8. The index value in LSTODAY measure started from 1 rather than 0.
9. FLP_FIRST and FLP_LAST are the resultant measures to be used for the First-Last-Populated Location calculation. They do not have the calendar dimension, and each of their cell values represent the indices for the first and last populated locations along the calendar dimension from the first time series up to the current time series, respectively.
10. TSMASK_DENSE is a Boolean input measure without calendar dimension to specify which time series is going to be processed and which is not. For filtering methods other than the CLEAR method, the true value means that it will be processed if the popcount for the current time series is larger than the hard-coded threshold value. Otherwise, it will not be processed. The false value means that the current time series will not be processed. If the TSMASK_DENSE measure is not specified, all the time series will be processed and the internal hard-coded threshold value will not be considered. For the CLEAR filtering method, the true value means that the previously calculated results for the current time series will be cleared and the false value means the results will be retained. If the TSMASK_DENSE measure is not specified, all the results will be cleared.
11. For all the input measures that do not have the calendar dimension, such as UP_ADJ_RATIO and DELTA, you can use a constant as input. In this case, the constant value will be applied to all the time series.

Configuration Restriction

- ‘_’ (underscore) may not be used in any measure names and rules unless the measures and rules are to be expanded using the RDF or Curve solution’s classification scheme. The classifications in question are as follows:
- AppFunctions
 - _F: Expand measures and rules across final levels
 - _S: Expand measures and rules across source levels
 - _B: Expand measures and rules across birth dates

Function Registration Requirements

The following functions are required to be registered in any domain(s) that will use the LostSale Solution Extension:

- AppFunctions
- LostSaleFunctions

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 it 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 are a complete set.
Bold	Labels.
<i>Italics</i>	Italics indicate a temporary placeholder for a constant or a measure (will end with 'meas').
<i>Italics/meas</i>	This indicates that the placeholder can be either a constant or a measure.
<i>BoldItalics</i>	This 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.
<u>Underlined And Large font</u>	This is 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.

Label	Parameter / Result	Required	Measure	Allowed to be multiple	Type	Description
METHOD	PARAMETER	Cluster – Y Cluster Statistics - N	N	N	String	Determines which clustering algorithm to use. Valid values are BANG (preferred) or KMEANS.
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 you to configure 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, which indicates that it is invalid because the RPAS JNI cannot validate it at this point in time. Therefore, there is no validation for this rule. Refer to the Grade documentation for the appropriate input parameters and output measures.
3. Make sure that the resultant measures are at the right intersection levels by using the information in section 3.2 above.
4. The Cluster procedure is a multi-result procedure, which means 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. The RPAS JNI cannot validate the Cluster procedure call, so 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. You must use the latest version of RPAS to build the domain. You will get the following message in the log because the Cluster function is not validated:

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

This message is okay.

8. After the domain build, use the regfunction RPAS utility to register the Grade library .The library, which is located in the \$RPAS_HOME/applib directory, is libClusterEngine.so. Do not specify the *lib* or *.so* part of the function name with the regfunction utility. For example, regfunction -d /domains/D01 -l ClusterEngine.
9. Use the Mace command to execute the Cluster rule with the rule group (for instance, grade_batch).


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

Function Registration Requirements

The following functions are required to be registered in any domain(s) that will use the Cluster Solution Extension:

- AppFunctions
- ClusterEngine

Configuring the ASOSpace Function

ASOSpace function uses a specialized Dynamic Programming method (Karigma) to generate an optimal Assortment plan that maximizes total profit constraint by total store space. This is 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.
<i>Italics</i>	Italics indicate a temporary placeholder for a constant or a measure (will end with ‘meas’).
<i>Italics/meas</i>	This indicates that the placeholder can be either a constant or a measure.
<i>BoldItalics</i>	This 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.
<u>Underlined And Large font</u>	This is 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	

Function Registration Requirements

The following functions are required to be registered in any domain(s) that will use the ASO Solution Extension:

- AppFunctions
- ASOExpressions

Configuring the ASOAssort Function

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

Although the target of the optimization is for each item, the ASOAssort function divides the total problem into two levels and solves 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.

Input arguments to the ASOAssort function (like Cost and Retail Prices) 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 the Collection level are returned together with the profit/cost table at each level, the Stock Out unit/dollars table, and the 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.
<i>Italics</i>	Italics indicate a temporary placeholder for a constant or a measure (will end with 'meas').
<i>Italics/meas</i>	This indicates that the placeholder can be either a constant or a measure.
<i>BoldItalics</i>	This 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.
<u>Underlined And Large font</u>	This is used to identify the function name.

Syntax

SubcategoryExpectedProfileTableMeas,
SubcategoryRecommendedCostTableMeas,
SubcategoryRecommendedActualCostTableMeas,
CollectionRecommendedProfileTableMeas,
CollectionRecommendedCostTableMeas,
ItemRecommendedFacingsTableMeas,
ItemExpectedProfileTableMeas,
ItemExpectedCostTableMeas,
ItemExpectedBuyQuantityTableMeas,
ItemStockOutUnitTableMeas,
ItemStockOutDollarsTableMeas,
HistogramProfitDataPointsMeas,
HistogramCostDataPointsMeas
<- **ASOAssort**(
IncludeMeas,
MinimumNumberOfFacingMeas,
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 { "PIOtScatProfR.level([cInd].[phse]),
PIOtScatCostX.level([cInd].[phse]), PIOTsctActCstX.level([cInd].[phse]),
PIOtCollProfR.level([cInd].[phse]), PIOTCollCostX.level([cInd].[phse]),
PIOtltmRecFacX.level([cInd].[phse]), PIOTltmProfR.level([cInd].[phse]),
PIOTltmSolCstX.level([cInd].[phse]), PIOTltmBuyQtyX.level([cInd].[phse]),
PIOTltmStkOutU.level([cInd].[phse]), PIOTltmStkOutR.level([cInd].[phse]),
PIOTHistProfR.level([cInd].[phse]), PIOTHistCostX.level([cInd].[phse])

<- ASOAssort(

PIWpIncludeB.level([cInd].[phse]), PIWpMinFacX.level([cInd].[phse]),
PIWpMaxFacX.level([cInd].[phse]), PIWpRstkTrgU.level([cInd].[phse]),
PIWpFacCapX.level([cInd].[phse]), PIFcDemandU.level([cInd].[phse]),
PIWpDemandMaxU.level([cInd].[phse]), PIWpMaxLTX.level([cInd].[phse]),
PIWpSfySkMultX.level([cInd].[phse]), PIWpManltmB.level([cInd].[phse]),
PIWpltmCstC.level([cInd].[phse]), AdWpOpHrsX.level([cInd].[phse]),
AdWpStkCISRtX, AdWpStkClkCTX, PIWpltmRetR.level([cInd].[phse]),
PIWpltmWX.level([cInd].[phse]), PIWpConstypSI.level([cInd].[phse]),
PIWpNmShlvsX.level([cInd].[phse]), PIWpPogUOMSI, PIWpltmUOMSI,
PIWpStepSizeX.level([cInd].[phse]), PIWpToleranceX.level([cInd].[phse]),
PIWpScatToIX.level([cInd].[phse]), PIWpStpSzUOMSI, PIWpToIUOMSI,
PIWpInclCollB.level([cInd].[phse]), PIWpManCollB.level([cInd].[phse]),
PIWpPogOptsSI.level([cInd].[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	The Minimum Number of Facings that must be included in the planogram.	SKU/STR/PHSE	
MaximumNumberOfFacingsMeas	The Maximum Number of Facings that can be included in the planogram.	SKU/STR/PHSE	
MinimumShelfPresenceMeas	The 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; the number of units in demand per PHSE.	SKU/STR/PHSE	
MaximumDemandMeas	The Maximum Demand over multiple periods. It will used to calculate the MaximumLeadTimeDemand (MaximumDemand x MaximumLeadTime).	SKU/STR/PHSE	
SafetyStockMultiplierMeas	The Percentage of Safety Stock.	SKU/STR/PHSE	
MandatoryMeas	The 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	
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	

Parameter	Description	Intersection	Usage Comment
StockClerkCycleTimeMeas	The number of hours between the stock clerk's round.	STR	
ItemRetailPrice	The 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	The Number of Shelves that 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	This indicates whether a STYL is included in the calculation.	STYL/STR/PHSE	
MandatoryCollectionMeas	This indicates 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	

Function Registration Requirements

The following functions are required to be registered in any domain(s) that will use the ASO Solution Extension:

- AppFunctions
- ASOExpressions