

Oracle® Retail Demand Forecasting
Administration Guide
Release 12.1

June 2007

Copyright © 2007, Oracle. All rights reserved.

Primary Author: Gary O'Hara

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, and Siebel 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**
 - Audiencev
 - Related Documentsv
 - Customer Supportvi
 - Conventionsvi

- 1 Batch Processes 1**
 - Overview.....1
 - Batch Processes.....2
 - PreGenerateForecast2
 - generate.....3
 - RDFvalidate.....5
 - UpdateFnhbiRdf9

- 2 AutoSource..... 11**

- 3 Forecast Approval Alerts 13**
 - Alerts.....13
 - Step 0: Using the Mock Install Configuration, build the Global Domain environment....13
 - Step 1 (option 1): Run PreGenerateForecast or Generate13
 - Step 1 (option 2): Use regTokenMeasure to Manually Register Any Token Measures Needed to Support the Alert Expression.....14
 - Step 2: Register the Alert Measure14
 - Step 3: Register the Expression for the Forecast Approval Alert14

Preface

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

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

Audience

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

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

Related Documents

For more information, see the following documents:

- Oracle Retail Demand Forecasting Release Notes
- Oracle Retail Demand Forecasting Installation Guide
- Oracle Retail Demand Forecasting Configuration Guide
- Oracle Retail Demand Forecasting User Guide
- Oracle Retail Predictive Application Server documentation

Customer Support

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

When contacting Customer Support, please provide:

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

Conventions

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

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

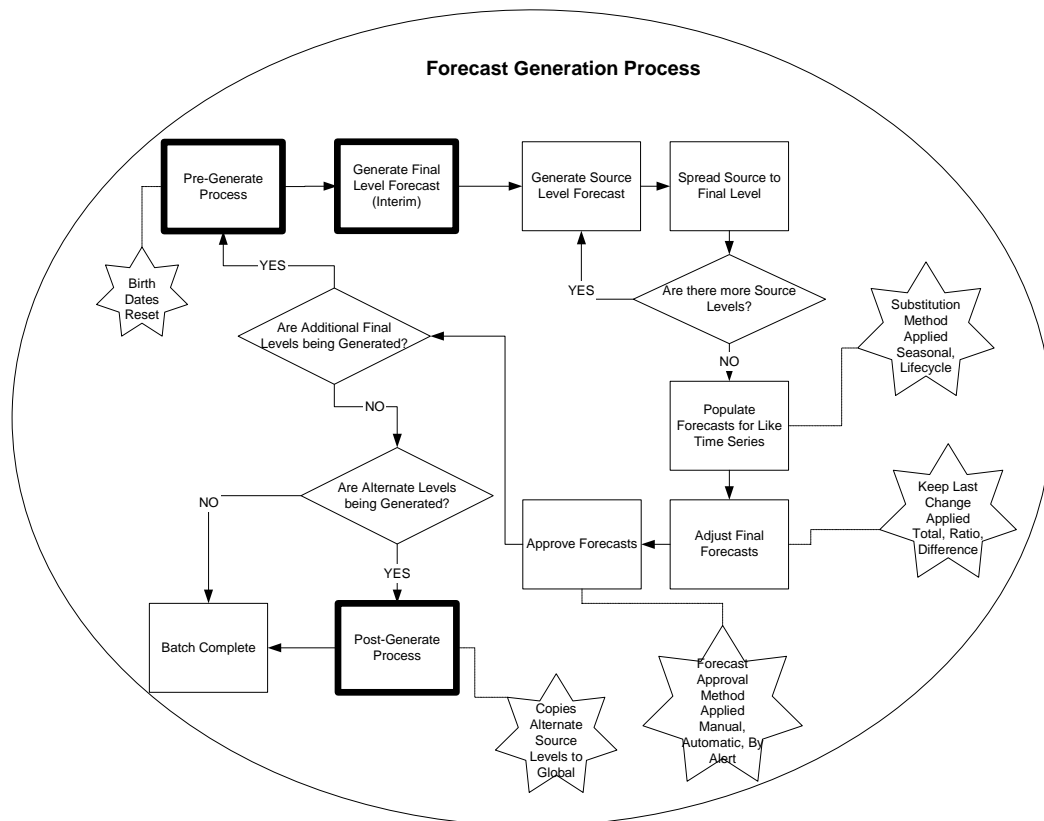
This is a code sample
It is used to display examples of code

A hyperlink appears like this.

Batch Processes

Overview

The diagram below provides a high-level overview of the batch forecast process.



Overview of Batch Forecast Process

Batch Processes

PreGenerateForecast

Used in a Global Domain or Simple Domain environment, 'PreGenerateForecast' is an RDF executable that registers all measures with a birth date prior to forecast generation using 'generate.' The first time 'PreGenerateForecast' is run for a level, it registers the appropriate token measures for that level. If a Global Domain environment is implemented, 'PreGenerateForecast' may be run against the Master or a Local domain. At either level, the necessary measures to produce the batch forecast will be registered across all domains.

'PreGenerateForecast' requires an input file in the form of an XML. The XML is configured with the following values:

- **FinalLevel:** The Final Level Number that will be used to generate the forecast.
- **OutputFile:** The name of the resulting file located at the root of the domain after 'PreGenerateForecast' is run. The OutputFile includes the values set for FinalLevel and Override in addition to the birth date. This date is the Forecast Generation Date, and it is passed to the domains when 'generate' is run.
- The date is produced in the following format: yyyyymmddHhhMmm (Example: 20050327H13M36). When this birth date is selected in the Forecast Approval wizard, it will be viewed as: (03/27/2005 13:36).
- **Override:** A true or false value. When 'generate' is passed a true value, the Next Run Date is ignored, and the batch forecast uses today's date as the Next Run Date; and the batch is executed. When 'generate' is passed a false value, the batch forecast will run if the Next Run Date is the same as today's date.

Note: When the 'Run Batch' template is used to generate the batch forecast, 'PreGenerateForecast' is run automatically. If a Global Domain environment is implemented, forecasts produced across Local domains using 'Run Batch' cannot be aggregated in the Master domain because they do not share the same Forecast Generation Date.

Usage

PreGenerateForecast -InputFile filename

InputFile is required.

The input file should be an XML file that looks like this:

```
<Parameters>
  <Parameter>
    <Key>FinalLevel</Key>
    <Value>1</Value>
  </Parameter>

  <Parameter>
    <Key>OutputFile</Key>
    <Value>MyOutput.xml</Value>
  </Parameter>
  <Parameter>
    <Key>Override</Key>
    <Value>true</Value>
  </Parameter>
</Parameters>
```

FinalLevel and OutputFile are required parameters of the XML file.

Override is an optional parameter of the XML file (default is false).

Other parameters may be included in the input XML file. They will be passed through to the output XML file.

Return codes:

- 0 - Success (either ran pregenerate or did not need to run)
- 1 - Bad input
- 2 - Failure

To set the logger verbosity level, use `-loglevel` with values of: `all`, `profile`, `debug`, `information`, `warning`, `error`, or `none`. To disable timestamp header use `-noheader`.

generate

'generate' is an RDF executable used to produce the batch forecast. This executable requires as an input, the OutputFile resulting from 'PreGenerateForecast.'

This binary runs RDF's batch process. Generate can take two optional inputs: level and override. The generate usage is as follows:

```
generate -InputFile Filename
```

The following parameters setting are included in the input file: `birth`, `startdate`, `finallevel`, and `override`.

The override input must be true or false. The defaulted value is false if this option is not included in the input file. When override is false, generate will only start the batch process if current time is later than the next run date in the domain. When the override is true, generate will start the batch forecast regardless of the next run date.

The generate binary invokes code in the BatchForecast library to execute the batch process.

Usage

`generate -InputFile filename`

`FinalLevel` and `Birth` are required parameters of the XML file. `Override` (false) and `StartDate` (Default Forecast Start Date) are optional parameters of the XML file (defaults in parentheses).

Return codes:

- 0 - Success (either ran generate or did not need to run)
- 1 - Bad input
- 2 - Failure

To set the logger verbosity level, use `-loglevel` with values of: `all`, `profile`, `debug`, `information`, `warning`, `error`, or `none`.

To disable timestamp header use `-noheader`.

The input file should be an XML file that looks similar to the following:

```
<Parameters>
  <Parameter>
    <Key>Birth</Key>
    <Value>20041027H11M52</Value>
  </Parameter>
  <Parameter>
    <Key>StartDate</Key>
    <Value>DAY20041027</Value>
  </Parameter>
  <Parameter>
    <Key>FinalLevel</Key>
    <Value>1</Value>
  </Parameter>
  <Parameter>
    <Key>Override</Key>
    <Value>true</Value>
  </Parameter>
</Parameters>
```

RDFvalidate

'RDFvalidate' automatically executes during the domain install, and it can also be run at any time against a Master or one subdomain. If run against the Master Domain, it checks the master and all subdomains. If run against a subdomain, it checks the Master and only the subdomain (not all other subdomains). This function verifies that:

- Domains are cleanly partitioned
- All data, measures, and levels are defined properly based on the partition dimension
- Causal parameters are properly defined based on final, source, and causal levels

Usage

```
rdfvalidate -d pathToDomain
```

To get this usage text, use `-?`, `-help`, or `-usage`. To get the version of this utility, use `-version`. To set the logger verbosity level, use `-loglevel` with values of: `all`, `profile`, `debug`, `information`, `warning`, `error`, or `none`. To disable timestamp header use `-noheader`.

RDF Validation

1. Hierarchies and Dimensions:
 - a. Verify "day" dimension exists on calendar hierarchy.
 - b. If there is a partition dimension, it must be along the product hierarchy.
 - c. Check whether or not "FMTR," "FLVL," and "FBRT" exist in Data Hierarchy. If not create them.
2. For final levels:
 - a. Intersection (`rdf::RDFSysConstants::INTERSECTION` or `fintxlxb`)
 - Cannot be blank
 - Must be at or below all source level intersections
 - Must be at or below the partition dimension on the partition branch
 - b. Seasonal profile (`rdf::RDFSysConstants::SEASONAL_PROFILE` or `seasprofxlxb`) can be either:
 - Blank
 - Measure name (only one)
 - Must be valid measure
 - Should be of type real
 - Measure intersection must be equal to the level intersection
 - c. Source data (`rdf::RDFSysConstants::DATA_SOURCE` or `datasrcxlxb`) must be a measure name (only one)
 - Must be a valid measure
 - Should be of type real
 - Measure intersection must be at or below the final level intersection

- d.** Plan data (rdf::RDFSysConstants::PLAN or fplanlxb) must be either:
 - Blank
 - Measure name (only one)
 - Must be valid measure
 - Should be of type real
 - Measure intersection must be equal to the final level intersection
- 3.** For source levels:
 - a.** Intersection (rdf::RDFSysConstants::INTERSECTION or fintlxb)
 - Cannot be blank
 - Must be at or above final level intersection
 - Must contain a dimension from the partition hierarchy
 - Must be either:
 - At or below the partition dimension on the partition branch.
 - On a branch of the partition hierarchy.
If on a branch of the partition hierarchy, also check if domains are “cleanly” partitioned (executable only). This means for the branched dimension on the partition hierarchy, each position for that dimension can exist in only one sub-domain.
 - b.** Seasonal profile (rdf::RDFSysConstants::SEASONAL_PROFILE or seasproflxb) can be either:
 - Blank
 - Measure name (only one)
 - Must be valid measure
 - Should be of type real
 - Measure intersection must be equal to the level intersection
 - c.** Spreading profiles (rdf::RDFSysConstants::SPREADING_PROFILE or sprdproflxb)
 - Can only be blank if source level intersection equals final level intersection
 - Must be comma-separated list of curve levels and measure names (can be mixed)
 - If curve level, must be a valid curve level (final profile)
 - If measure:
 - Must be a valid measure
 - Should be of type real
 - Measure intersection must be at or above final level

Plug-in Only

1. Should call “manageFNHBI -validate” at install time
2. Should call the RDF validation at install time

Executable Only

1. Make sure there is only one partition dimension per sub-domain (Basic assumption of our code)
2. Final and source levels:
 - a. Causal Aggregation Profile (rdf::RDFSysConstants::AGG or aggxlxb) values should be either:
 - Blank
 - Measure name (one only)
 - Should be a valid measure
 - Should be of type real
 - The intersection of the measure must be at or above final level.
 - b. Causal Calculation Intersection (rdf::RDFSysConstants::CALC_INT or calcintlxb) values should be either:
 - Blank
 - Intersection
 - Must be valid intersection
 - Must contain the calendar dimension
 - Must be at or above level intersection.
 - c. Causal Data Source (rdf::RDFSysConstants::CALC_DATA_SOURCE or calcdtsrcxlb) values should be either:
 - Blank
 - Measure name (one only)
 - Should be a valid measure
 - Should be of type real
 - The intersection of the measure must be at or above level intersection.
 - d. Causal Higher Intersection (rdf::RDFSysConstants::CAUSAL_HIGHER_INT or cslhint) values should be either:
 - Blank
 - Intersection
 - Must be valid intersection
 - Must not contain the calendar dimension
 - Must contain a dimension from the partition hierarchy.
 - Must be at or above level intersection.
 - Must be either:
 - At or below the partition dimension on the partition branch.
 - On a branch of the partition hierarchy.

Note: If on a branch of the partition hierarchy, also check if domains are “cleanly” partitioned (executable only). This means that for the branched dimension on the partition hierarchy, each position for that dimension can exist in only one sub-domain.

- e. Causal Spread Profile (rdf::RDFSysConstants::SPREAD or spreadxlxb) values should be either:
 - Blank
 - Measure name (one only)
 - Should be a valid measure
 - Should be of type real
 - The intersection of the measure must be at or above final level.
 - f. Deseasonalized Demand Array (rdf::RDFSysConstants::DESEASONALIZED_DEMAND or ddemandxlxb) values should be either:
 - Blank
 - Measure name (one only)
 - Should be a valid measure
 - Should be of type real
 - The intersection of the measure must be the level intersection less the calendar dimension
3. Final Levels only
- a. Default History Start Date (rdf::RDFSysConstants::DEF_HISTORICAL_START_DATE or defhstdt) values should be either:
 - Blank
 - A date within the calendar
 - b. Forecast Start Date (rdf::RDFSysConstants::DEF_FORECAST_START_DATE or dfxlxb) values should be either:
 - Blank
 - A date within the calendar

Promote Validation

Plug-in and Executable

1. Hierarchies and Dimensions:
Check whether or not "PTYP", "FLVL", and "PROM" exist in Data Hierarchy. If not, create them.
2. Promotion Names:
Check if promotion names have 1-4 characters.
3. Causal levels must be at or below the partition dimension on the partition branch.

UpdateFnhbiRdf

- 'UpdateFnhbiRdf' is required after 'Generate' is executed if an alternate hierarchy dimension from the Product hierarchy is used as a dimension in a forecast level
- Checks that certain measures are cleanly partitioned
- Copies corresponding cells (based on the partition) from each sub-domain to the master domain
- Runs automatically with the 'Run Batch' wizard
- After ensuring that the FNHBI (Forced non-Higher Based Intersections) measures are cleanly partitioned, UpdateFnhbiRdf copies corresponding cells (based on the partition dimension) from each sub-domain into the master domain

Usage

```
UpdateFnhbiRdf -d pathToDomain -InputFile filename
```

To get this usage text, use `-, -help, or -usage`. To get the version of this utility, use `-version`. To set the logger verbosity level, use `-loglevel` with values of: `all, profile, debug, information, warning, error, or none`. To disable timestamp header, use `-noheader`.

The InputFile format expected is as printed by the usage information. The timestamp or the birth key will have to be the same as the one output by `pregenerateForecast`, that is used by `generate`.

AutoSource

The AutoSource binary may be used to determine the optimal source level for a product/location. For the final level specified, AutoSource produces a forecast using each source level. The source level that produces the best MAE (Mean Absolute Error) for a time series is selected as the Optimal Source Level. The AutoSource results may be accessed by the user through the Forecast Maintenance workbook. If the Optimal Source Level is to be used for a product/location, the 'Use Optimal Source' parameter should be set to true.

The AutoSource binary invokes code in the BatchForecast library to execute the batch process. AutoSource can take four inputs: mode (required), finallevels (required), today, and timelimit (required).

- Provides a starting Source Level recommendation for new forecasting customers. The recommended Source Level can be applied to the Final Level, which would allow the user to be focused on other tuning activities.
- Helpful for existing customers that are starting to forecast new businesses. AutoSource can be included as an activity in the customer's forecasting roll-out process.
- Useful for merchandise groups that have shifting demand patterns due to business or market changes such as pricing and marketing strategy changes, or product realignment.

The following AutoSource measures are available in the Forecast Maintenance workbook.

Optimal Source Levels

Displayed only at final levels, a value will be populated in this field if 'AutoSource' has been run on the final level. The 'AutoSource' executable evaluates all levels associated to a final level and returns the Source Level that yields the optimal forecast results or lowest error.

Pick Optimal Level

Set only at final levels, a checkmark in this field indicates that the batch forecast should use the 'Optimal Source Level' selected by 'AutoSource.'

The final level measure "Optimal Source Levels" is used for reference. The RDF user can view the 'optimal' Source Level that was determined by AutoSource. This Source Level was chosen by generating forecasts at all Source Levels and determining the lowest forecast error (PAE) at the final level.

If the user would like to use the "Optimal Source Level" during forecast generation they can set the "Pick Optimal Level" Boolean measure to TRUE.

If "Pick Optimal Level" is set to TRUE, when forecast generation is run, the 'optimal' Source Level will be used. The Forecast Method set at the 'optimal' Source Level and the additional associated forecast parameters will also be used.

Usage

Usage: `autosource -d pathToDomain -mode RESTART/ONCEONLY/CYCLE -flvllist lvlx,lvly`
[-today] todayString(the same format as in `dim_day`)
[-timelimit] minutes [-preserveTemp]

To get this usage text, use `-?`, `-help`, or `-usage`.

To get the version of this utility, use `-version`.

To set the logger verbosity level, use `-loglevel` with values of: `all`, `profile`, `debug`, `information`, `warning`, `error`, or `none`.

To disable the timestamp header, use `-noheader`.

The mode input must be one of `RESTART`, `CYCLE`, or `ONCEONLY`.

The `flvllist` must be a comma separated list of final levels.

The `today` input must be the same format as `dim_day`.

The `timelimit` is in minutes.

To run in `ONCEONLY` mode, `RESTART` mode has to be run first:

- **RESTART:** This mode initializes the system in preparation for a new Autosource batch process.
- **ONCEONLY:** This mode will run the Autosource batch process until it completes or until the `timelimit` has been reached (whichever comes first).
- **CYCLE:** This mode will continuously run the Autosource batch process by first running the `RESTART` mode, and then running `ONCEONLY`. The `CYCLE` mode allows the Autosource batch process to always use the latest data in determining the optimal source level for a `prod/loc`.

Example 1:

```
Autosource -mode RESTART -flvllist 1,6 -today DAY20050101 -timelimit 10
Autosource -mode ONCEONLY -flvllist 1,6 -today DAY20050101 -timelimit 10
```

Example 2:

```
Autosource -mode CYCLE -flvllist 1,6 -today DAY20050101 -timelimit 10
```

If AutoSource is to only run periodically, the 'RESTART' and 'ONCEONLY' modes should be used. If the run exceeds the time limit, then 'ONCEONLY' should be run again to pick up where it left off. If you want to start from the beginning again, 'RESTART' and 'ONCEONLY' should be run.

If AutoSource is to be scheduled as part of the daily cron job, 'CYCLE' should be used. 'CYCLE' will run 'RESTART' then 'ONCEONLY' consecutively.

Forecast Approval Alerts

Alerts

Alerts can be configured through the RPAS Configuration Tools or can be registered on the backend of the domain. The alert expressions require familiarity with the RPAS rule functions. Registering an alert with the alert category of "FORECAST_APPROVAL" allows RDF to use the alert expression during the batch forecasting process to determine if a time series is automatically approved. When this category of alert is registered, the pick lists for Default Approval Method (in Forecast Administration) and the Approval Method Override (in Forecast Maintenance) are updated to include the label of the alert. The user then has the ability to select the alert for any product/location.

The following is an example of Forecast Approval Alert configuration using the mock install that is provided in the release package.

Step 0: Using the Mock Install Configuration, build the Global Domain environment.

Step 1 (option 1): Run PreGenerateForecast or Generate

If using a pristine Global Domain or Simple Domain environment, token measures have yet to be registered in the domain(s). Since you do not know the specific birth date at configuration time, token measures allow for measures with birth dates (a time stamp applied during the batch) to be evaluated. The token measure that we are using in this example is System Forecast for level 1 (sf01). The registration of the token measures can be accomplished by running PreGenerateForecast (in a Global Domain environment) or Generate (in a Simple Domain environment). This removes the need to manually execute 'regTokenMeasure.'

Step 1 (option 2): Use regTokenMeasure to Manually Register Any Token Measures Needed to Support the Alert Expression

If you prefer to manually register the token measures, the regTokenMeasure must be run with -FNHBI option if in a Global Domain environment. This allows the token measures to have different values across subdomains. The token measure requires a value to the measure while registering. In the following example, the token measure is registered in the **Master Domain** and are made to be equal to 'pos' (Weekly Sales) since 'pos' has the same base intersection (item/store/week) and data type (real) as the System Forecast for level 1.

Example:

```
C:\Domains\RDF>regTokenMeasure -d . -add sf01=pos -fnhbi
```

Note: DO NOT do this step if the batch has already been generated since the batch will have automatically registered sf01.

Step 2: Register the Alert Measure

The next step in the process is to register the alert measure **in the Master Domain**. In the following example, an alert with the name of "rdf_a1_1" with label of "Alert1level1" is being registered.

Example:

```
C:\Domains\RDF>regmeasure -d $DOMAIN_DEST_DIR -add "rdf_a1_1" -label  
"Alert1level1" -baseint "itemstr_" -db "data/myalerts" -type boolean -navalue  
False
```

Step 3: Register the Expression for the Forecast Approval Alert

The alertmgr utility is used to register the alert and the alert expression. In the following example, the alert expression first filters out time series with low volume sales (items with forecasts less than three units). This alert compares the System Forecast in the first three weeks in the forecast horizon with last approved forecast for the same three weeks. If the values are within a 33% range, the full forecast horizon is set to automatic approval, otherwise the Alert is triggered. This is all done in batch, so the Alert Manager is NOT necessary to apply the alert. For intersections that do not qualify for automatic approval, the Approval Comment on the Approval Worksheet in the Forecast Approval workbook will contain 'refused by alert.' You may use the Alert Manager to insert this alert into the workbook to display the intersections that have the alert flag set to true.

Example:

```
C:\Domains\RDF> alertmgr -d . -register "rdf_al_1" -category "FORECAST_APPROVAL" -
categoryLabel "Alert1level1" -expression
"rdf_al_1=if(tssum(@sf01,index([cldn].[week],flookup(lfsXLXB.level([data].[flvl]+[
prod].top),[data].[flvl].[flvl01])),
index([cldn].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[data].[flvl].
[flvl01]))+3)>=3.0, abs(1-
tssum(@sf01,index([cldn].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[d
ata].[flvl].[flvl01])),
index([cldn].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[data].[flvl].
[flvl01]))+3)/(tssum(lappf01XB,index([cldn].[week],flookup(lfsXLXB.level([data].[f
lvl]+[prod].top),[data].[flvl].[flvl01]))+3,index([cldn].[week],flookup(lfsXLXB.le
vel([data].[flvl]+[prod].top),[data].[flvl].[flvl01])))+0.001))>.33,false)"
```

See the RPAS Administration Guide for more information on registering alerts and token measures.