

**Retek® Demand Forecasting™  
Retek® Curve™  
Retek® Promote™  
11.1.8  
Administration Guide**



---

**Corporate Headquarters:**

Retek Inc.  
Retek on the Mall  
950 Nicollet Mall  
Minneapolis, MN 55403  
USA  
888.61.RETEK (toll free US)  
Switchboard:  
+1 612 587 5000  
Fax:  
+1 612 587 5100

**European Headquarters:**

Retek  
110 Wigmore Street  
London  
W1U 3RW  
United Kingdom  
Switchboard:  
+44 (0)20 7563 4600  
Sales Enquiries:  
+44 (0)20 7563 46 46  
Fax:  
+44 (0)20 7563 46 10

The software described in this documentation is furnished under a license agreement, is the confidential information of Retek Inc., and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403, and the copyright notice may not be removed without the consent of Retek Inc.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

The functionality described herein applies to this version, as reflected on the title page of this document, and to no other versions of software, including without limitation subsequent releases of the same software component. The functionality described herein will change from time to time with the release of new versions of software and Retek reserves the right to make such modifications at its absolute discretion.

Retek<sup>®</sup> Demand Forecasting<sup>™</sup>, Retek<sup>®</sup> Curve<sup>™</sup>, Retek<sup>®</sup> Promote<sup>™</sup> are trademarks of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2005 Retek Inc. All rights reserved.

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.

---

# Customer Support

## Customer Support hours

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

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

Contact Method	Contact Information
----------------	---------------------

E-mail	support@retex.com
--------	-------------------

Internet (ROCS)	<a href="https://rocs.retek.com">rocs.retek.com</a> Retek's secure client Web site to update and view issues
-----------------	---

Phone	+1 612 587 5800
-------	-----------------

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

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835

Mail	Retek Customer Support Retek on the Mall 950 Nicollet Mall Minneapolis, MN 55403
------	---

## When contacting Customer Support, please provide:

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

# Contents

<b>Chapter 1– Installation and Patching Processes .....</b>	<b>1</b>
sciInstall .....	1
Usage .....	2
sciPostInstall .....	2
sciPatch .....	3
Usage .....	3
sciPostPatch .....	4
<b>Chapter 2 – Batch Processes .....</b>	<b>5</b>
Overview .....	5
Batch processes .....	6
PreGenerateForecast .....	6
generate .....	7
RDFvalidate .....	9
Curvevalidate .....	14
ManageFNHBI .....	15
curvebatch .....	16
<b>Chapter 3 – AutoSource .....</b>	<b>17</b>
<b>Chapter 4 – Forecast Approval Alerts .....</b>	<b>19</b>
Alerts .....	19
Step 0: Using the LabsGA3 configuration, build the Global Domain environment. ....	19
Step 1 (option 1): Run PreGenerateForecast or Generate .....	19
Step 1 (option 2): Use regTokenMeasure to manually register any token measures needed to support the alert expression .....	19
Step 2: Register the alert measure .....	20
Step 3: Register the expression for the Forecast Approval Alert .....	20
<b>Chapter 5 – Exporting Forecast Data .....</b>	<b>21</b>
Exporting Data .....	21



# Chapter 1– Installation and Patching Processes

## sciInstall

Located in the `$SCI_HOME/scripts` directory, the `sciInstall.ksh` script is used to install domains that support RDF, Curve, Promote, or Grade. The `sciInstall.ksh` script:

1. Verifies the path to the 'globaldomainconfig.xml'
2. Runs the 'rpasinstall' executable with the `-fullinstall` option
3. Registers the following libraries:
  - a. AppFunctions: RDF Common Functions
  - b. ClusterEngine: Grade Functions
  - c. LostSaleFunctions: Preprocessing Functions
  - d. RdfFunctions: Predict and Promote Functions
4. Curve, RDF, Promote, or Grade-specific install tasks, depending on the solutions implemented
5. Runs 'checkparents' on each loaded hierarchy
6. Runs 'sciPostInstall.ksh'
7. Outputs the installation log file to either:
  - The default location if the `-o` option is not used with `sciInstall`:  
`$SCI_DATA/configurations/domainPath/sciInstallLog.txt`
  - The user-specified path if the `-o` option is used with `sciInstall`
8. Creates the domain(s)

The installation process creates two directories under `domainPath`.

- **ProjectName:** The name of the project that is defined in the RPAS Configuration Tools. This will be the domain directory root. If installing a Global Domain environment (LabsGA3), you will see within the D01 directory each of the local domains (`ldom0`, `ldom1` and `ldom2`). In this case D01 is the Master domain.
- **Installs:** Contains information on the configuration used to create the domain(s). It is necessary to support the domain patching process. Do not delete or move this directory.



**Note:** You may see an error similar to the following during the install:

**Error when reading domain properties: [[DatabaseOpenFailed: path to \$SCI\_HOME/scripts/data/admin]].**

This error should be ignored, but it will not be displayed in the `sciInstallLog.txt`. An open defect has been logged to RPAS to remove this error message.

## Usage

sciInstall.ksh -d DomainPath -c ConfigName [-j ProjectName] [-p ConfigPath] [-i InputPath] [-o the path and name of log file] [-g Global Domain DimName] [-f Global Domain ConfigDirectory]

**Example 1:** Install a Simple domain environment using the LabsGA2 configuration.

```
$SCI_HOME/scripts/sciInstall.ksh -d /Domains/LabsGA2 -c LabsGA2 -j D01
```

**Example 2:** Install a Global Domain environment using the LabsGA3 configuration and the `-g` option.

```
$SCI_HOME/scripts/sciInstall.ksh -d /Domains/LabsGA3 -c LabsGA3 -j D01  
-g pgrp
```

**Example 3:** Install a Global Domain environment using the LabsGA3 configuration and the `-f` option.

```
$SCI_HOME/scripts/sciInstall.ksh -d /Domains/LabsGA2 -c LabsGA2 -j D01  
-f /SCI_HOME
```



**Note:** The globaldomainconfig.xml, any custom xml, data or scripts that is not specified in this document should not be stored at or below the `$SCI_DATA/configurations/ProjectName` directory.



**Note:** If you are building a Global Domain environment, the `-g` and the `-f` options with sciInstall SHOULD NOT be used together.

## sciPostInstall

Custom post-domain creation steps can be automated by creating a shell script named sciPostInstall.ksh in the `$SCI_HOME/scripts` directory. The sciInstall.ksh script will call the sciPostInstall.ksh script before exiting. For the LabsGA2 and LabsGA3 configurations, this script is pre-configured to do the following:

1. Load data files from the `$SCI_DATA/data` directory into the domain.

If building a Global Domain environment, sciPostInstall.ksh is pre-configured to split the data files (.dat files) for the LabsGA3 configuration. This is based on the partition positions.

2. Run a Rule Group in the domain(s).

**sciPostInstall must be edited or removed before installing any configurations other than LabsGA2 or LabsGA3.**



## sciPatch

Located in the `$SCI_HOME/scripts` directory, the `sciPatch.ksh` script is used to patch domains that support RDF, Promote, Curve, or Grade. Invoke the domain patch script (`sciPatch.ksh`) using the following syntax:

### Usage

`sciPatch.ksh -d DomainPath -c ConfigName [-j ProjectName] [-p ConfigPath] [-i InputPath] [-o the path and name of log file] [-g Global Domain DimName | -f Global Domain ConfigDirectory]`

**Example 1:** Install a Simple domain environment using the LabsGA2 configuration.

```
$SCI_HOME/scripts/sciPatch.ksh -d /Domains/LabsGA2 -c LabsGA2 -j D01
```

**Example 2:** Install a Global Domain environment using the LabsGA3 configuration and the `-g` option.

```
$SCI_HOME/scripts/sciPatch.ksh -d /Domains/LabsGA3 -c LabsGA3 -j D01 -g pgrp
```

**Example 3:** Install a Global Domain environment using the LabsGA3 configuration and the `-f` option.

```
$SCI_HOME/scripts/sciPatch.ksh -d /Domains/LabsGA2 -c LabsGA2 -j D01 -f /SCI_HOME
```



**Note:** The `globaldomainconfig.xml`, any custom xml, data or scripts that is not specified in this document should not be stored at or below the `$SCI_DATA/configurations/ProjectName` directory.

The `sciPatch.ksh` script:

1. Runs the ‘`rpasinstall`’ executable with the `-patchinstall` option
2. Registers the following libraries:
  - a. AppFunctions: RDF Common Functions
  - b. ClusterEngine: Grade Functions
  - c. LostSaleFunctions: Preprocessing Functions
  - d. RdfFunctions: Predict and Promote Functions
3. Curve, RDF, Promote, or Grade-specific patching tasks depending on the solutions implemented:

If Curve is implemented:

- a. Install data hierarchy: Register positions (minor classes under those major classes) that are in dimensions: "CMTR", "TMP", and "TNPD"
- b. Generate data and load the following measures: “`dataintXL`”, “`pgdataXL`”
- c. Install message labels that are in wizards
- d. Run `curvevalidate` without “-s”

If RDF is implemented:

- a. Install data hierarchy: Register positions (minor classes under those major classes) that are in dimensions: "FMTR," "TMP," and "TNPD"
- b. Install message labels that are in wizards
- c. Install RDF messages
- d. Generate data and load the following measure: "datasrcXLXB"
- e. Run rdfvalidate and manageFNHBI

If Promote is implemented:

- a. Install data hierarchy: Register positions (minor classes under those major classes) that are in dimensions: "PROM" and "TMP"
- b. Install message labels that are in wizards

If Grade is implemented:

- a. Register positions (minor classes under those major classes) that are in dimensions: "GMTR" and "GTMP"
- b. Install message labels that are in wizards



**Note:** For patch installs, dimensions are not created for major classes. Therefore, the changes to major classes cannot be patched, except for the custom created major classes.

4. Runs 'checkparents' on each loaded hierarchy
5. Runs sciPostPatch.ksh
6. Outputs the log file to: \$SCI\_DATA/configurations/domainPath/sciPatchLog.txt



**Note:** You may see an error similar to the following during patching:

**Error when reading domain properties: [[DatabaseOpenFailed: path to \$SCI\_HOME/scripts/data/admin]].**

This error should be ignored, but it will not be displayed in the sciInstallLog.txt. An open defect has been logged for the removal of this error message.

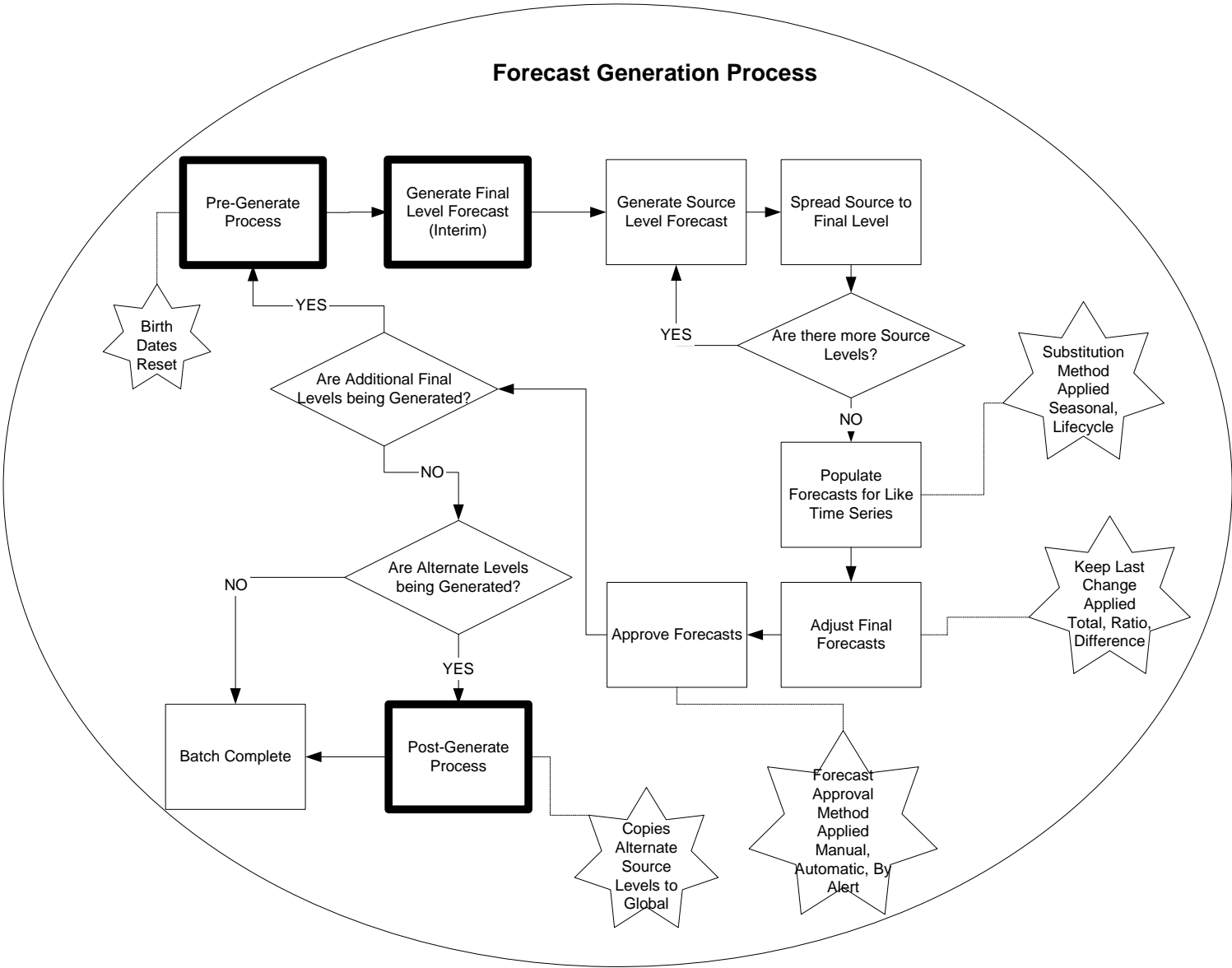
## sciPostPatch

Custom post-patch process can be automated by creating a shell script, named sciPostPatch.ksh in the \$SCI\_HOME/scripts directory. The sciPatch.ksh script will call the sciPostPatch.ksh script before exiting. **sciPostPatch.ksh is delivered as part of the release package but is not pre-configured for any post-patching processes.**

# Chapter 2 – Batch Processes

## Overview

The following is a high-level overview of the 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 predict'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, 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 datasrcx1xb)  
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 fplanx1xb)  
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 fintx1xb)
    - 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 seasprof1xb)  
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 sprdprofxlxb)
  - 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 calcintxlxb)

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 calcdtsrclxb)

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 spreadlxb)

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 ddemandxlb)

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 dfxlb)

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.

### Plug-in Only

None

### Executable Only

None

## Curvevalidate

‘Curvevalidate’ 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:

- Profile and Source intersections and source data are properly defined
- Profile intersections respect the partition dimension

## Usage

curvevalidate -d domainpath [-s]

-s : set defaults

-d domain : set path to domain

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.

1. Each Profile must have at least one Source Level.
2. For each Profile:
  - a. For global domains, ALL intersections {Data Intersection, Profile Intersection, Stored Intersection, Aggregation Intersection, and Approval Intersection} must be below the partition (NOT HBI).
  - b. Data Intersection (if a data source is specified) must conform to X in {Profile Intersection, Stored Intersection, Aggregation Intersection, and Approval Intersection}.
  - c. Profile Intersection must conform to the Stored Intersection.
  - d. Aggregation Intersection must conform to the Approval Intersection.
  - e. Aggregation Intersection must not be below the Approval Intersection.
  - f. Aggregation Intersection must be above the Data Intersection (if data source specified).
  - g. If the Aggregation Intersection conforms to Profile Intersection:
    - The Profile Type must NOT be diff(8).
    - The Aggregation Intersection must be above the Profile Intersection.
    - The Aggregation Intersection must be above the Stored Intersection.
  - h. If Aggregation Intersection does not conform to Profile Intersection:
    - The Profile Type must be Diff (8).
    - There must be at least one common hierarchy between the Aggregation Intersection and X in {Profile Intersection, Stored Intersection}.
    - For each common non-PROD hierarchy H of Aggregation Intersection and X in {Profile Intersection, Stored Intersection}:

Aggregation Intersection’s H dimension must not be below X’s H dimension.

3. For each Source Level:
  - a. For global domains, ALL intersections {Profile Intersection, Stored Intersection, and Aggregation Intersection} must be below the partition (NOT HBI).
  - b. Parent Profile's Data Intersection (if data source specified) must conform to X in {Profile Intersection, Stored Intersection, and Aggregation Intersection}.
  - c. Profile Intersection must conform to Stored Intersection.
  - d. Aggregation Intersection must be above parent Profile's Data Intersection (if data source specified).
  - e. If Aggregation Intersection conforms to Profile Intersection:
    - The Profile Type must NOT be diff(8).
    - The Aggregation Intersection must be above the Profile Intersection.
    - The Aggregation Intersection must be above the Stored Intersection.
  - f. If Aggregation Intersection does not conform to Profile Intersection:
    - The Parent Profile Type must be Diff (8).
    - There must be at least one common hierarchy between the Aggregation Intersection and X in {Profile Intersection and Stored Intersection}.
    - For each common non-PROD hierarchy H of Aggregation Intersection and X in {Profile Intersection and Stored Intersection}:
 

Aggregation Intersection's H dimension must not be below X's H dimension.

## ManageFNHBI

- 'ManageFNHBI' 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, ManageFNHBI copies corresponding cells (based on the partition dimension) from each sub-domain into the master domain

## Usage

`managefnhbi -d pathToDomain [-ignoreModDim] [-update]`

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`.

## curvebatch

### Usage

curvebatch -d domainpath [-level # ] [-debug] | -h | -version

-level must be followed by a valid Profile ID

-debug causes temporary measures to be retained for debugging purposes

-h prints this usage message

-version prints version information

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

## Chapter 3 – 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).

### 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
```





# Chapter 4 – Forecast Approval Alerts

## Alerts

Alerts are one of the only configurable functions that are not configured through the RPAS Configuration Tools, but they must be registered on the backend of the domain. The alert expressions require a good familiarity with the RPAS rule functions (rpas-1104-rfrg.pdf). 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 (LabsGA3).

### Step 0: Using the LabsGA3 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\LabsGA\D01>regTokenMeasure -d . -add sf01=pos -fnhbi
```

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\_al\_1” with label of “Alert1level1” is being registered.

Example:

```
C:\Domains\LabsGA3\D01>regmeasure -d $DOMAIN_DEST_DIR/D01 -add
"rdf_al_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\LabsGA3\D01> alertmgr -d . -register "rdf_al_1" -category
"FORECAST_APPROVAL" -categoryLabel "Alert1level1" -expression
"rdf_al_1=if(tssum(@sf01,index([clnd].[week],flookup(lfsXLXB.level([data].
[flvl]+[prod].top),[data].[flvl].[flvl01])),
index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[data].
[flvl].[flvl01]))+3)>=3.0, abs(1-
tssum(@sf01,index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[pr
od].top),[data].[flvl].[flvl01])),
index([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[da
ta].[flvl].[flvl01]))+3)/(tssum(lappf01XB,index([clnd].[week],flookup(l
fsXLXB.level([data].[flvl]+[prod].top),[data].[flvl].[flvl01]))+3,index
([clnd].[week],flookup(lfsXLXB.level([data].[flvl]+[prod].top),[data].[
flvl].[flvl01])))+0.001))>.33,false)"
```

See the RPAS 11.1 Administration Guide for more information on registering alerts and token measures.

# Chapter 5 – Exporting Forecast Data

## Exporting Data

The RPAS exportData utility is a generic utility that is used to support the export of data based on a specified format that is indicated in the call to the utility. More information on exportData can be found in the RPAS 11.1, Administration Guide. The following is an example of how this script may be configured:

```
exportData -d . -out exportOut -array "data/appf|appf01xb%1 %12.4f 0
%12.4f" -dim "WEEK data/hmaint|week2date %-8s 1" -dim "ITEM 4 %-20.20s
2" -dim "STR 4 %-20.20s 3"
```

Using the above example, the data is exported with the date that corresponds to the week level position; however, for execution systems, an additional step prior to using the exportData utility is required if the final forecast level is at an aggregate dimension above 'day' and the data is needed to be exported with the day level date position. The following script creates a 'week2date' array in hmaint database to be used for converting week IDs in export data to end-of-week dates. It uses MSPL to do this because there is no utility in RPAS 11 to do the same.

Example:

```
cd data
aculmode hmaint -m256m -ttt <-- EOF

    set datelist dim_week(hdr:posnames dim:week);
    set format to 'yyyymmdd';
    set timedim to 'week';
    set position to 'last';
    set dayArray to 'dim_day';
    set dayDimName to 'day';
    set convarray to 'week2date';
    update fromstream datelist list.(lookup.(<position> datelist
                                                <dayArray>(info: (timedim))
                                                <dayArray>(hdr: posnames dim:
<dayDimName>)))));
    set <convarray> lit.(datelist strip left 3);
    update array_options__(hdr:posdel dim:array) with convarray;
    update array_options__(hdr:posadd dim:array) with convarray;
    update fill array_options__(array:<convarray> info:popcount) with
populated.(<convarray>);
    update fill array_options__(array:<convarray> info:naval) with 0;
    update fill array_options__(array:<convarray> info:lastupdate) with
'190001011111111111';
    update fill array_options__(array:<convarray> info:sparse) with
false;
```

```
        erase format timedim position dayArray dayDimName datelist;  
        exit;  
EOF
```