Oracle® Retail Predictive Application Server

Administration Guide
Release 12.0.2

October 2006

ORACLE

Oracle® Retail Predictive Application Server Administration Guide, Release 12.0.2

Primary Author: Gary O'Hara

Contents

Preface ..... **vii**
Audience vii
Related Documents vii
Customer Support vii
Conventions vii

1 Introduction **1**
Overview 1
System Administration Workbooks 1
Workbook and Wizard Descriptions 1
General Workbook Procedures 2
Change a Workbook's Calculation Method 2
Refresh and Export Data 2
Global Domain 4
Overview 4
Measure Data 4

2 Domain Administration **7**
Overview 7
Starting the DomainDaemon 7
Monitoring the DomainDaemon 7
Stopping the DomainDaemon 8
Losing a Client-Server Connection 8
showActiveServers 8
stopServer 8
Other DomainDaemon Commands 9
deactivate 9
activate 9
showLockTable 9
releaseLocks 9
Environment Variables 10
Lock Timeout Variable 10
Log File Backups 10
Centralized Administration 11
Administrative Workbook Templates and Wizards 11
RPAS Utilities 12

3 Security and User Administration **13**
Functional Overview 13
User Logon Security 13
Measure Level Security 14
Position Level Security 14
Workbook Security 15
User Administration 16
Overview 16
Procedures 16
Security Administration Workbook 18
Overview 18
Procedures 21

4 Hierarchy Maintenance 25
Overview 25
Hierarchy Maintenance Workbook 25
Hierarchy Maintenance Wizard 26
Hierarchy Maintenance Worksheet 26
Procedures 28
Access Hierarchy Maintenance 28
Maintain a User-Defined Dimension within a Hierarchy 28

5 Measure Analysis 31
Overview 31
Measure Analysis Workbook 31
Measure Analysis Wizard 31
Measure Analysis Worksheet 31
Procedures 32
Access Measure Analysis 32
Review and Edit Sales or Other Registered Measure Data 32

6 Workbook Auto Build Maintenance 35
Overview 35
Workbook Auto Build Maintenance Wizard 35
Procedures 35
Accessing the Auto Build Maintenance Workbook 35
Add a Workbook to the Auto Build Queue 35
Delete a Workbook from the Auto Build Queue 36

7 Translation Administration 37
Overview 37
General Translations Worksheet 37
Measure Translations Worksheet 37
Template Translations Worksheet 37
Template Group Translations Worksheet 37
User Group Translations Worksheet 38
Procedures 38

8 Commit as Soon as Possible 41
Overview 41
Using Commit ASAP 41
Managing the Workbook Queue – showWorkbookQueues 42
Usage 42
Commit ASAP Settings – configCommitAsap 43
Usage 43
Logging and Technical Information 44

9 Batch Processes and RPAS Utilities 45
Overview 45
Using Shell Scripts to Run Batch Processes 45
A Sample Shell Script 45
Running Utilities in a Global Domain Environment 46
Common Information and Parameters for RPAS Utilities 47
Logging Batch Processes 48
Configuration Tools Log Files 48
Error Files 48

10 Hierarchy Management 51
Overview 51
Placeholder Positions in the Domain 51
Position Repartitioning 52
Procedures 52
Enabling Dummy Positions 52
Configuring and Scheduling the Rebuffering Process 53
Loading Hierarchies – loadHier 53
Reconfiguring the Partitions of a Global Domain – reconfigGlobalDomainPartitions 55
Updating or Reporting the Dummy Position Buffer – positionBufferMgr 58
Renaming Positions – renamePositions 59
Setting Properties for Dimensions – dimensionMgr 61

11 Data Management 62
Loading Measure Data – loadmeasure 62
Usage 62
Running Scripts 63
Exporting Measure Data – exportData 63
Usage 64
Mapping Data between Domains – mapData 67
Usage 67
Moving Data between Arrays – updateArray 67
Usage 68

12 Operational Utilities 71
Find Alerts – alertmgr 71
Usage 71
Copying Domains – copyDomain 72
Usage 72
Setting Miscellaneous Domain Properties – domainprop 73
Usage 73
Calculation Engine – mace 75
Usage 75
Managing Users – usermgr 77
Usage 77
Managing the Workbook Batch Queue – wbbatch 79
Usage 79
Workbook Manager – wbmgr 81
Usage 81

13 Informational Utilities 83
Retrieving Domain Information – domaininfo 83
Usage 83
Checking the Validity of a Domain – checkDomain 84
Usage 84
Determining RPAS Server Version – rpasversion 84
Usage 84
List Contents of a Database – listDb 85
Usage 85
Printing Data from Arrays – printArray 85
Usage 85
Printing Data from Measures – printMeasure 86
Usage 86

A Appendix: Integration Guide 87
RDF and Merchandise Financial Planning Integration with RMS and Price 87
Summary of Integration Approach with RMS 87
Environment Variable Setup 88
RDF and Merchandise Financial Planning Transformation Programs 88
RDF and Merchandise Financial Planning Transformation Matrix 92
RDF and Merchandise Financial Planning Extract Programs 93
Common Programs for Extracts 93
Extract of Forecast Data for RMS 94
Load of Extracted Forecast Data and Standard Deviations to RMS 95
Extract of Diff Profile Data for RMS 95
Extract of Store Grade Data for RMS 96
Extract of Receipt Plan for RMS 97
Extract of Markdown Budget Data for Price 98
Extract of Current Plan Data for RDW 99
Extract of Original Plan Data for RDW 101
RDF and Merchandise Financial Planning Extract Matrix 103

B Appendix: Curve Administration Guide 105
Curvevalidate 105
Usage 105
Curvebatch 106
Usage 106

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

This document is intended for the users and administrators of Oracle Retail Predictive Application Server. This may include merchandisers, buyers, and business analysts.

Related Documents

For more information, see the following documents in the Oracle Retail Predictive Application Server Release 12.0.2 documentation set:

- Oracle Retail Predictive Application Server Release Notes

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.

1
Introduction

Overview

All RPAS-based products require setup and the following administration activities to be performed:

- Domain administration
- User account management
- User and workbook template administration
- Hierarchy maintenance
- Measure analysis
- Workbook auto build maintenance
- Translation administration

System Administration Workbooks

Using the administration workbooks, designated employees manage other employees' use of the Oracle Retail Predictive Solutions. System administrators use the administration workbooks to perform the following:

- Set up and maintain users and user groups
- Manage users' access to specific workbook templates and individual measures
- Modify the labels associated with users and user groups
- Modify the labels associated with workbook templates and template groups
- Create custom workbook templates and template groups
- Edit the contents of translation tables to support multiple-language use of the application
- Specify the type, frequency, and format of workbooks in the automatic build queue

Workbook and Wizard Descriptions

- User Account Management wizards – a set of wizards for setting up and maintaining users and user groups
- Security Administration workbook – a workbook for setting up and maintaining user/template, user/measure, and template/measure access rights
- Workbook template definition – a workbook for specifying parameters of a custom workbook template
- Translation Administration workbook – a workbook for managing the foreign language translation of strings and label text throughout the application
- Workbook auto build maintenance – a workbook for managing the workbook auto build queue

General Workbook Procedures

Change a Workbook's Calculation Method

There are two types of calculation modes that can be set in the RPAS client:

1. "Deferred" calculation mode (the most common) allows you to make multiple edits in a workbook before recalculating the data. In this mode, the edits are effectively "queued" and executed once you hit "Calculate."
2. "Automatic" calculation mode forces the workbook to be recalculated every time a cell is changed. This forces immediate communication from the worksheet back to the database. In this mode, there may be a pause between one data change and your ability to effect the next change.

For efficiency and usability purposes, Oracle recommends that you always operate in "deferred" calculation mode.

Set the Workbook to Deferred Calculation Mode

Click the Edit menu and select Manual Calculation.

Return the Workbook to Automatic Calculation Mode

Click the Edit menu, and select Automatic Calculation.

Send the Queue of Data Changes to the Server

Click the Edit menu, and select Calculate Now.

Refresh and Export Data

Refresh the Data in a Worksheet

The Refresh feature allows you to update a workbook with the data that is currently stored in the domain. This lets you work with the most current data without having to rebuild the workbook.

Refresh the Data for all Worksheets and Measures in the Workbook

From the File menu, select Refresh.

Export the Current Worksheet View to an Output File

Navigate: From the File menu, select Export Sheet. The Save As dialog box displays.

1. In the Save In field, select a directory on your computer where you want to save the export file.
2. In the File Name field, type a name for the export file.
3. Click the Save As Type drop-down list, and select a file type for the export file.

4. Make a selection for each of the following:
   a. Delimiter – Specify the character that is used to separate information in the output file. Standard choices are Tab, Comma, or Space; but you can specify a different delimiter by selecting the Other radio button.
   b. Labels – Specify the format of the label headers across the top of the cells in the output file. The options are:
      – **Do Not Include** – no labels provided
      – **Include Once** – one label placed across the top of each section of related cells
      – **Repeat** – a separate label, repeated as necessary, appears atop each cell
      – **Descriptions** – specify whether to identify dimensional positions in your output file with concise system names (for instance, SKU00012) or the descriptive labels (for instance, Cashmere Sweater – L – beige) assigned to each position
5. Select Save to export the file.
6. Click **OK**.

Insert Measures into an Open Worksheet

If you have the necessary access rights, you can insert a new measure or group of measures into a workbook that is already open. This functionality reduces the need to build new workbooks whenever a view of currently unrepresented measures is required. The ability to insert new measures into open workbooks is particularly useful when establishing access to alerts.

Use the following procedure to select a measure or group of measures to be inserted in a workbook that is currently open.

> **Note:** A worksheet must be open and active for the Insert Measure menu option to be enabled. Any measure(s) that are to be inserted in the workbook will be placed on the currently active worksheet.

Navigate: From the Edit menu, select Insert Measure.

1. From the Metric list box, select the desired metric(s). If a measure exists that matches the combination of attribute types selected in the preceding steps, its name/label will appear in the Measures list box. Multiple measures may appear here if multiple selections were made in any of the preceding steps.

   > **Note:** You can right-click in the Metric window to produce a quick menu where you can toggle the display between metric name (the system name) and metric label as it appears in the workbook.

2. Select the desired measure(s) in the Measures list box.
3. Click **OK**.

Global Domain

Overview

"Global Domain" is a type of domain structure that provides you with the ability to view data from multiple domains and to administer common activities of an RPAS domain and solution.

Domains can be built in one of two methods:

- **Simple domain –** this is the traditional, stand-alone domain that has no visibility to other domains.
- **Global domain –** this is a domain environment that contains two or more "local" domains (or "sub-domains") and a "master" domain that has visibility to all local domains that are part of that environment.

Using a Global Domain environment has two primary functional benefits. The first feature allows you to have a global view of data in workbooks. You can build workbooks with data from local domains, refresh global workbook data from local domains, save global workbooks, and commit the data from global workbooks to the individual local domains.

"Local" domains are typically organized ("partitioned") along organizational structures that reflect user roles and responsibilities. Most users will only work within the local domain(s) that contain their area of responsibilities, and they may not need to be aware of the Global Domain environment. For performance and user contention reasons, Global Domain usage should be limited to relatively infrequent processes that require data from multiple local domains.

The other primary feature of Global Domain is centralized configuration and administration. Most of the mechanisms that are required to build and administer a domain have been centralized and they need only be run in the "master" domain, which either propagates data to the local domains or stores the data centrally so that the local domains reference it in the master domain.

Note: For a Global Domain environment to function properly, all local domains must be structurally identical.

Measure Data

In a global domain environment, measure data can be physically stored in two different ways:

1. across the local domains
2. in the master domain

Measure data that is stored in local domains is split across domains based on a pre-determined level of a given hierarchy. This level is defined during the configuration process, and it is referred to as the "partition" level.

The base intersection of a measure (for instance, what dimensions a measure contains) determines whether data is stored in the local domains or in the master domain. The data will be stored in the master domain if the base intersection of a measure is above the "partition" level or if it does not contain the hierarchy on which the Global Domain environment is partitioned. This type of measure is referred to as a "Global Domain measure," or a "Higher Base Intersection measure."

Consider a global domain environment where the partition-level is based on the Department dimension in the Product hierarchy. Data for measures that have a base intersection in the Product hierarchy at or below Department is stored in the local domain based on the Department that the underlying position in the Product hierarchy belongs to. Other hierarchies are irrelevant for this discussion.

However, measures that have a higher base intersection in the Product hierarchy than Department (for instance, Division) or measures that do not contain the Product hierarchy (such as a measure based at Store-Week) cannot be split across the local domains. These measures will reside in the master domain and will be accessed from there when these measures are required in workbooks.

All measures will be registered in the master domain, and they are automatically registered in all local domains. RPAS automatically determines where the measure needs to be stored by comparing the base intersection of the measure against the designated partition-level of the Global Domain environment.

The physical location of the measure data will be invisible to the user after the measure has been registered. However, administrators must know where data for each measure is to be stored (master vs. local) as the data must be loaded in the proper location.

2

Domain Administration

Overview

The RPAS DomainDaemon is a process that is used to enable the communication channel between RPAS clients and RPAS domains.

The DomainDaemon runs on the server side and waits for requests from RPAS clients on a given port. Once DomainDaemon receives a request from a client, it starts a server process that the client connects to. From this point, the client and server communicate directly. The system administrators may choose to have one single DomainDaemon process for all of the users, or they may choose to have separate processes per domain, per enterprise, and so on.

The DomainDaemon is installed in the [RPASDIR]/bin directory. [RPASDIR] stands for the full path to the directory where the RPAS server is installed. The system administrators can start, stop, and monitor the DomainDaemon processes by using scripts that are provided in this directory.

Starting the DomainDaemon

In order to start the DomainDaemon, execute the script called DomainDaemon in the Acumate installation directory. The port number where the DomainDaemon will be running must be passed in as an argument. The port number must be between 1025 and 65535. If **auto** is specified instead of a number, the DomainDaemon is started on any available port.

> **Note:** In the following examples, [RPASDIR] stands for the full path to the directory where the RPAS server is installed.

Example:

Issuing the following command from a UNIX shell will start a DomainDaemon on port 55278:

```
([RPASDIR]/bin)$ DomainDaemon –port 55278 –start &
```

Monitoring the DomainDaemon

The –ping argument can be used to see whether a DomainDaemon is active. The port number must also be passed as an argument. If the DomainDaemon is active on the port, a message will be printed, and the script will return true. Otherwise, the script will return false.

Example:

```
([RPASDIR]/bin)$ DomainDaemon –port 55277 –ping
DomainDaemon on port 55277 is alive.
```

Stopping the DomainDaemon

Use the –stop argument to stop the DomainDaemon running on a given port.

Example:

```
([RPASDIR]/bin)$ DomainDaemon –port 55277 –stop
```

Losing a Client-Server Connection

There are certain procedures to follow if the connection between the RPAS client and the RPAS server is lost. This connection can be lost for any number of reasons, but most commonly when your computer crashes or if your network connection is lost.

If this situation occurs, notify your system administrator if you do not have access to the server processes.

The system administrator needs to perform two steps:

1. Find the lost RPAS server process that is associated with that user. This is done by using the "—showActiveServers" argument as specified below. Make note of the user's process ID.
2. Stop the user's RPAS server process, which will remove any locks and allow the user to log into the RPAS client and begin a new RPAS server process. This is done using the "—stop server command" and the user's process ID as specified below.

You can then log back into the RPAS client.

showActiveServers

Use the –showActiveServers argument to list all the active server processes managed by the DomainDaemon. You must specify a port number.

For each active server, the DomainDaemon shows the process ID, domain, and user ID.

Example:

```
DomainDaemon –port 55277 –showActiveServers
```

Returns:

```
Registered Servers:
```

Users	ProcessID	DomainPath
'adm'	56789	/mydomains/domain1

stopServer

Use the –stopServer argument to stop a specified server process. You must specify a port number and a process ID.

Example:

```
DomainDaemon –port 55277 –stopServer 56789
```

Returns:

```
Stop Server succeeded.
```

Other DomainDaemon Commands

deactivate

Use the `–deactivate` argument to mark a domain as temporarily unavailable. Deactivating a domain also terminates all user sessions in that domain. Domains are most commonly deactivated before beginning a routine nightly/weekly batch process. This ensures that no users make updates to the system during these processes.

You must specify the port number and the complete path to the domain.

Example:

```
DomainDaemon –port 55277 –deactivate /mydomains/domain1
```

activate

Use the `–activate` argument to reactivate a domain that you previously deactivated. You must specify the port number and the complete path to the domain.

Example:

```
DomainDaemon –port 55277 –activate /mydomains/domain1
```

showLockTable

Use the `–showLockTable` argument to display the contents of the database lock table.

Example:

```
DomainDaemon –port 55277 –showLockTable
Lock Table:
```

releaseLocks

Use the `–releaseLocks` argument to release all database locks held by the specified process. You must specify a port number and a process ID.

Example:

```
DomainDaemon –port 55277 –releaseLocks 15920
All locks released.
```

Environment Variables

RPAS includes a number of environment variables that are set at the system level in UNIX. At the system level, the variables are applicable to all RPAS servers (DomainDaemons) that are run on the system.

The common syntax for setting these variables is as follows:

Export ENVIRONMENT_VARIABLE=XXXXXX

"ENVIRONMENT_VARIABLE" is a defined variable that is recognized by RPAS. "XXXXXX" is an appropriate value for the variable, which could be a string, Boolean, or numeric data type. If the value represents time, this number normally represents time in milliseconds.

Note: The DomainDaemon must be restarted after setting any environment variables. An example of how this process is completed is as follows:

```
DomainDaemon -port 55123  -start -debug &
```

Lock Timeout Variable

When performing certain operations, it is possible for two or more users to be "contending" for access to the same database (.gem file), which happens most commonly when two users attempt to simultaneously commit/save the same data back to the domain. By default, RPAS is set up to wait one minute before returning a lock contention error when this situation occurs.

If desired, an administrator can override this default value by setting the "RPAS_LOCK_TIMEOUT" environment variable. This variable is set to the number of milliseconds to wait for a file lock before returning a lock contention error. As with any environmental variable, the variable must be set prior to starting the process that uses that variable. The variable was introduced for use with the RPAS database server, which means that the variable is set for the DomainDaemon.

For example, the line below demonstrates how an administrator would tell RPAS to wait two minutes before returning a lock contention error with the RpasDbServer after launching the client and logging in. Any client that connects to that domain daemon would see lock contention after a two minute delay.

Export RPAS_LOCK_TIMEOUT=120000

Log File Backups

The "RPAS_LOG_BACKUPS" environment variables allow an administrator to define the number of log file backups to retain for a given user. A log file is created each time for each session that a user has with the RPAS client.

The environment is set by executing the following command:

Export RPAS_LOG_BACKUPS=X

"X" is an integer value that represents the number of backup log files to keep for each user.

Centralized Administration

Note: If a solution is built in a Global Domain environment, most administrative activities can only be performed in the "master" domain. This applies to RPAS administrative workbook templates and wizards as well as RPAS utilities that are run on the back-end against the domain.

Administrative Workbook Templates and Wizards

The following list includes the standard RPAS workbook templates and/or wizards that have been centralized. It can only be run in the master domain of a Global Domain environment. See the individual sections for additional information.

- Alert Manager window – Results of the alert finder run on the global domain are collated and displayed in this window.

1. This applies to all alerts registered in the global domain.
2. Results are based on data from all the individual local domains.
3. Results are consolidated (added together) to display a single result per measure.

- Alert Manager workbook template – This template is used to build alert workbooks from the Alert Manager dialog window. Data will be retrieved from the local domains.
- Measure Analysis – This is used to analyze measure data from local domains.
- Security Administration – This is used to set security by template, measure, and positions. This workbook template can only be used in the master domain, and it is disabled for use in local domains.
- User Administration – User information will be set up and maintained in the global domain, but it will be replicated to the local domains. Updates are effective immediately after the changes are committed. This workbook template can only be used in the master domain, so it is disabled for use in local domains.
- Translation Administration – This is a template that is used to modify the labels of translatable data in RPAS. This workbook template can only be used in the master domain, so it is disabled for use in local domains.
- Hierarchy Maintenance – This is used to set up and maintain positions of user-defined dimensions. User-defined dimensions must be registered in the Global Domain by using utility reguserdim.

RPAS Utilities

The following list includes the standard RPAS utilities that have been centralized. These utilities can only be run in the master domain of a Global Domain environment. See the individual sections for additional information.

- Alerts (**alertmgr**)
  a. Alerts registered in the global domain will be propagated to local domains.

     Note: Alerts registered in local domains will not (and do not have to) be included in the global domain.

  b. Utility for finding alerts (a.k.a. alert finder) will be updated to run against the local domains and collate results in the global domain for a centralized view of the alert results.
- Loading hierarchies (loadhier) – It is required that hierarchy information be centrally administered in the Global Domain and replicated to the local domains.
- Loading measure data (loadmeasure) – Utility for loading input data into measures in a domain; loadmeasure can load a single input file that contains data for multiple local domains. The utility splits the data for loading into the local domains, and the utility can be run in parallel.
- Exporting measure data (exportdata) – Utility for exporting measure data from a domain to flat files. The utility is called from the master domain and included the data from one or more local domains as specified by the call to the utility.
- Users (usermgr) – The back-end utility for managing users will propagate changes from the global domain to local domains. Ultimately this will also be updated for the RPAS Administrative Template "User Administration."
- Domain properties (domainprop) – This is used for manipulating properties, such as specifying password properties and locking user accounts.
- Miscellaneous registration utilities – Some of the registration utilities have been updated to operate in the global domain where all changes are propagated to the local domains.
  a. Measures (regmeasure)
  b. Templates (regtemplate)
  c. Functions (regfunction)
  d. Token measures (regTokenMeasure)
  e. User-defined dimensions (reguserdim)

3

Security and User Administration

Functional Overview

This chapter describes the security model in RPAS, which includes workbook templates, workbooks, measures, and positions. The levels of security are defined as measure level, position level, and workbook level.

This chapter also describes user administration and security administration.

> **Note:** If a solution is built in a Global Domain environment, it is only required and possible to perform the administrative activities included in this section in the "master" domain.

User Logon Security

Your user account may be marked as **locked out** by the domain administrator.

This will prevent you from logging in to the RPAS 11.0 client. The account remains locked out until the administrator re-enables the account.

Account lockouts may be set or cleared by the domain administrator by using the User Management utility.

Your account may be marked as **must change password**.

This is useful for brand-new accounts. You will be allowed to logon with the current password and then forced to select a new password.

Must change password may be set or cleared by the domain administrator using the User Management utility.

Account Lockout may be enabled for a domain.

The domain administrator selects a number of failed logon attempts after which the User account will be marked as locked out. The account will remain locked out until the administrator re-enables it.

Account Lockout can be enabled through the domainprop utility by using the `-lockAccount` flag.

Password expiration may be enabled for a domain.

The domain administrator selects a number of days after which passwords expire. When you log in, the system requires a new password to be entered if the configured number of days has passed since this user entered a new password.

Password expiration can be enabled through the domainprop utility using the `-expirePassword` flag.

Password history may be enabled for a domain.

The domain administrator selects a number of passwords to save. When you attempt to change passwords, the system will not permit any password already stored in the password history to be used again.

Password history may be enabled through the domainprop utility using the `-passwordHistory` flag.

Measure Level Security

Measures have access rights; which are read-write, read-only, or denied. Measures that are read-write or read-only may be selected in the extra measures and insert measure dialogs. RPAS ensures that read-only measures are not editable by the user and the presence of read-only measures does not affect the ability to commit a workbook.

Measure security can be specified when the measure is registered, and it can be changed later by using the Security Administration workbook. The Measure Rights worksheet allows you to specify Read Only, Deny, or Read/Write access to a measure for each user.

A workbook template can override the default security of a measure, but it can only narrow the security of the measure. For example, a measure could have default read-write access for a user and a template could specify that all users have read-only access to the measure when a workbook is built. However, if the default measure security was read-only, the template could not expand the security of that measure to read-write. Measures that are explicitly made read-only by a workbook template will not be expanded to read-write access by RPAS.

Position Level Security

Position Level Security allows access control for dimensions on a position-by-position basis. This capability is completely optional. If position level security is not explicitly defined and configured, all users in a domain have access to all positions in all hierarchies. Once position level security is defined, access to a position can be granted or denied for individual users, users in a group, or for all users.

Position level security can be defined at levels (dimensions) at or above base (such as class in the product hierarchy) in any hierarchy other than calendar. As positions are added at a level/dimension lower in the hierarchy than where the position level security is maintained, access to those positions is automatically granted if a user has access to the "parent" position. In other words, if security is maintained at the subclass level, users are automatically granted access to all the SKUs in a given subclass if they have access to that subclass. This includes those that were added after security was established.

Exactly one dimension in each hierarchy can be defined as the security dimension for the hierarchy. If a security dimension is defined for the hierarchy, all dimensions in the hierarchy have position level security enabled, but position security is set at or above the designated dimension. For instance, if the "class" dimension is designated as the security dimension, an administrator can maintain access to positions in the class dimension or at any level above class.

To specify the security dimension for a hierarchy, use the RPAS Configuration Tools or the **hierarchyMgr** utility.

After a security dimension is defined for a hierarchy, all users in the domain default to having access to all positions in any dimension in the hierarchy. Additionally, users automatically have access to newly added positions to a domain. Worksheets in the Security Administration workbook are used to control position access for individual users, user groups, or all users (referred to as "world" or default access). There are three worksheets in this workbook for each hierarchy with a defined security dimension. The default worksheet controls access to positions for all users (for instance, Prod Security Default); one worksheet controls access to positions by user group (for instance, Prod Security Group); and the last worksheet controls access to positions by individual users (for instance, Prod Security User).

Access must be granted at all levels for a user to have access to a position. This means that a position must have a value of true at the levels default/world, group, and user.

The following table demonstrates how access is granted or denied based on all combinations of settings:

Security set by Position **Denied = False** **Granted = True**			**Based on settings on left, user is Granted or Denied access**
User	**User Group**	**World**	**Resulting Access**
Denied	Denied	Denied	Denied
Denied	Denied	Granted	Denied
Denied	Granted	Denied	Denied
Granted	Denied	Denied	Denied
Denied	Granted	Granted	Denied
Granted	Denied	Granted	Denied
Granted	Granted	Denied	Denied
Granted	Granted	Granted	Granted

Position level security is used when a user selects positions in the wizard process before building a workbook. Only positions to which a user has access are available for selection in the 2-tree, which are then included in the build of the workbook.

Workbook Security

Currently, workbook access is either granted or denied. If you have been granted access to a workbook; you can open, modify, and commit the workbook. No distinction is made between read-write-commit, read-write, and read-only access. Workbook access is automatically granted to the user that built it, and it may be shared with multiple groups or the world.

> **Note:** A user must have access to the workbook template in order to access the workbook, even if the workbook has world or group access rights.

Users with administrator status automatically have access to all workbook templates. By default, administrators have access to all workbooks that are saved with world access. If a workbook is saved with group access, administrators can only access the workbook if they are members of the default user group of the user who saved the workbook.

Another aspect of workbook security is the ability to set limits for the number of workbooks that a user can have saved at any given time. Limits can be set for a user per template, for a user group per template, or for a template for all users. The limits are evaluated in the above order, which means that a limit defined at user-template overrides any values defined at group-template or template. If the above limits are not defined, the default value is one billion.

The limits are checked when you begin the workbook build process. When your limit is reached, an error message displays informing you that the workbook build process cannot complete because you have reached your limit. The message also lets you know what that limit is. The wizard process then terminates.

Administrative users have full access to all workbook templates regardless of the access rights that other admin users may assign to them in the Security workbook.

User Administration

Overview

User administration is the process by which administrators add and/or delete authorized system users, create and/or delete user groups, and edit user profiles. These tasks are performed through completion wizards on the User Administration tab. The following procedures are discussed in this area:

- Access the User Administration tab
- Add a user
- Add a user group
- Delete a user
- Delete a user group
- Edit a user's profile

Once users and user groups are set up, you can set access permissions to workbook templates and measures within workbooks through Security Administration. You can also modify the label, default workbook template, and/or Admin status associated with individual users.

Procedures

Access the User Administration Tab

1. Select New from the File menu. The New dialog box is displayed.
2. Select the User Administration tab.

Add a User

1. From the File menu, select New.
2. Click the User Administration tab.
3. Select Add User
4. Click **OK**.
5. In the ID field, type the ID string that the user will use for logging on.

   Note: Each user ID must begin with a letter, contain no spaces (the underscore character is acceptable), and not exceed eight characters total.

6. In the User Label field, type a label that describes the user (for example, the user's full name). This identifying label appears in various locations throughout the application. For example, labels appear on the File > Open dialog box to identify the owner of a given workbook, and on the Forecast Approval worksheet to specify which user approved a given forecast.
7. In the Default Group field, select the user group to which the user will belong.
8. If a user will belong to more than one group, select the additional groups from the list in the Other Groups field.
9. In the Password field, type a password for the user.

10. In the Password Verification field, type the same password.
11. If the user should have Admin status, check the Administrator box.

> **Note:** Admin status enables users to perform the Format menu option Save Format/Admin, which creates new system-wide default styles for workbook templates. If you are not sure whether a user should be granted this ability, you can modify a user's Admin status at a later date on the Users worksheet of the User and Template Administration workbook.
>
> **Note:** Granting users Admin status gives them access to all workbook templates, but it does not automatically give them access to all workbooks.

12. If the user must change his or her password when logging on for the first time, check the Force Password Change box.
13. If you want to temporarily disable the user's account, check the Lock User Account box.
14. Click **Finish** to add the new user to the database.

You can now assign workbook template and measure access rights to the user. To do so, access the User and Template Administration workbook.

Add a User Group

User groups provide an intermediate level of security to workbooks that were created and saved by specific users. When new users are assigned to the system, they must be assigned to existing user groups. User groups should consist of individuals with similar job functions or responsibilities. In the Oracle Retail Predictive Planning Suite, the user group corresponds to the user's planning role.

1. Select New from the File menu.
2. Click the User Administration tab.
3. Select Add User Group.
4. Click **OK**.
5. In the Group Name field, type a name for the group.
6. In the Group Label field, type a descriptive label for the group. This label is displayed when referring to the group throughout RPAS.
7. Click **Finish** to add the user group to the database.

Delete a User

If a user profile is no longer needed, it should be deleted from the system in order to maintain system security.

1. From the File menu, select New.
2. Click the User Administration tab.
3. Select Delete User.
4. Click **OK**.
5. Select the name of the user you want to delete.
6. Click **Finish** to delete the user from the system.

Delete a User Group

If a user group no longer exists, you should delete the group from the system as soon as possible to maintain system security.

> **Caution:** Deleting a user group will delete every user in that group.

1. From the File menu, select New.
2. Click the User Administration tab.
3. Select Delete User Group.
4. Click **OK**.
5. Select the user group you want to delete.
6. Click **Finish** to delete the user group from the system.

Edit a User

1. From the File menu, select New.
2. Click the User Administration tab.
3. Select Edit User.
4. Click **OK**.
5. Select the user you want to edit.
6. Click **Next**.
7. Make the necessary changes to the user's profile. You can change anything except the User Name. See "Add a User" for details.
8. Click **Finish** to save the changes.

Security Administration Workbook

Overview

The Security Administration workbook is only available to system administrators. After users and user groups are created, you may set up and maintain access permissions to workbook templates and measures within those workbook templates. You can determine which templates individual users can access, as well as the measures that users can access while manipulating workbooks in the system. You can also specify and restrict the measures that are available to be added to a given workbook template. Setting access permissions in this way provides a high degree of measure security, because users can be restricted to viewing and editing only certain relevant measures.

All administrative users have full access to all workbook templates regardless of the access rights that they were assigned in the Security workbook by other administrative users. The administrative user can build the Security workbook to change the access right back, so the nominal assignment does not matter for admin users.

The Security Administration workbook has one or more worksheets for each of the following:

- Workbook Template Rights worksheet
- Workbook Template Measure Rights worksheet
- Measure Rights worksheet
- Dimension Modification Rights
- Position Level Security
- Workbook Template Limits

Security Template Administration also allows you to modify the label, Admin status, and/or default workbook template associated with each user. You also access this workbook template to modify the labels associated with user groups, workbook templates, and workbook template groups. Using this workbook, you can:

- Assign and modify access rights of each user to all workbook templates. User/template permissions are set in the Workbook Template Rights worksheet.
- Determine which optional measures are to be accessible through individual workbook templates. Template/measure permissions are set in the Workbook Template Measure Rights worksheet.
- Assign/restrict user access to individual measures. User/measure permissions are established in the Measure Rights worksheet.

Workbook Template Rights Worksheet

The Workbook Template Rights worksheet is for setting and maintaining access permissions of each user to specific workbook templates.

The worksheet contains a checkbox for each available workbook template and user combination. A checkmark in the cell indicates that the user has access rights to that specific template.

To grant a user access rights to a workbook template, put a checkmark in the checkbox in for that workbook template.

To deny a user access rights to that specific workbook template, leave the checkbox blank or clear the checkmark.

After changing a user's profile, the changes must be committed to the database in order for them to take effect.

Workbook Template Measure Rights Worksheet

The Workbook Template Measure Rights worksheet allows administrators to determine which registered measures will be available for optional inclusion in newly built workbooks.

When a measure is initially registered as a public measure, all templates default to having access to that measure. This means that it is possible for this measure to be added to a workbook template, even if it is not one of the standard measures displayed when a workbook of that type is built. Some new workbook wizards include a dialog that prompts users to select any additional measures to be included in the workbook build. By default, all newly registered measures are included on this list of available additional measures. The other method of inserting new measures into a workbook is via the Insert Measure command.

The Workbook Template Measure Rights worksheet is used to modify template/measure permissions, which allows only certain templates to optionally include specified measures in new workbook builds.

This worksheet contains a checkbox for each available workbook template and registered measure combination. A checkmark in the cell indicates that the measure will appear on the additional measures list in the new workbook wizard and is therefore available for inclusion in the initial workbook build.

To make a given measure available in a workbook template, put a checkmark in the checkbox that represents that template/measure intersection.

To prevent a measure from appearing in a specific workbook template, clear the checkbox located at that template/measure intersection.

After changes are made to a template's profile, they must be committed in order for them to take effect.

> **Note:** Measures that are registered as private measures will not appear in this worksheet. If there are no public measures available to be displayed in this worksheet, the worksheet will not be built.

Measure Rights Worksheet

The Measure Rights worksheet allows you to restrict user access to individual measures on a user-by-measure basis. User/measure permissions are initially determined by the system by integrating the current user/template and template/measure settings and applying the following rule: "A user cannot have access to any measure that is not available in at least one template to which the user has access."

Permissions can be made even more restrictive on a user by measure basis by using the Measure Rights worksheet to deny users access to measures that they would normally be permitted to edit.

The worksheet contains a checkbox for each available user and registered measure combination. A checkmark in the cell indicates that the user has access rights to and is permitted to view and edit the specified measure. More specifically, that given user has access to at least one template that is permitted to include the specified measure.

You can further restrict a user's access to specific measures by removing checkmarks that are displayed at those associated intersections. Removing the checkmark (by clicking on it) prevents you from accessing a given measure, even though that measure is normally permitted to be displayed in a particular workbook template to which you do have access.

After changes are made to a user's profile, they must be committed before they take effect.

> **Note:** The Measure Rights worksheet contains only public measures; that is, measures that can be optionally included in a worksheet, depending on choices made in a new workbook wizard. Measures that are registered as private measures will not appear in this worksheet. If there are no public measures available to be displayed in this worksheet, the worksheet will not be built.

Dimension Modification Rights Worksheet

The Dimension Modification Rights worksheet allows you to determine which dimensions, if any, a user can modify. The worksheet contains a checkbox for each available user and dimension combination. A checkmark in the cell indicates that the user is permitted to modify the specified dimension.

After changes are made to a user's dimension modification rights, they must be committed before they take effect.

Position Level Security Worksheets

The position-level security worksheets are used to grant or deny access to positions for individual users, user groups, or all users. Position-level security is set for a specific dimension of a hierarchy (other than calendar). This is completed by using the **inithier** utility or the Configuration Tools.

For each hierarchy/dimension for which position level security is enabled (normally just a single hierarchy/dimension), there are three worksheets: one each for user, user group, and world/all users.

After changes are made to position level security, they must be committed before they take effect.

Workbook Template Limits Worksheets

The Workbook Template Limit worksheets are used to limit the number of workbooks that you can have saved. Limits can be set for a user per template, for a user group per template, or for a template for all users. The limits are evaluated in the above order, which means that a limit defined at user-template will override any values defined at group-template or template. If the above limits are not defined, the default value is one billion.

The limits are checked when you begin the workbook build process. If your limit has been reached, an error message appears that informs you that the workbook build process cannot complete because you have reached your limit. The wizard process then terminates.

Procedures

> **Note:** These tasks are performed through the Security Administration workbook. This workbook is only available to system administrators.

Access Security Administration

1. From the main menu, select File > New. The New dialog box is displayed.
2. Select the Administration tab to display a list of workbook templates for Administration.
3. Highlight Security Administration.
4. Click **OK**.

Set or Modify User Access to Workbook Templates

1. From the File menu, select New.
2. Click the Administration tab.
3. Select Security Administration.
4. Click **OK**.
5. On the Workbook Template Rights worksheet, select each template for which a user needs access rights. For templates to which the employee should not have access rights, make sure there is no check mark.
6. Changes must be committed to the master database before they take effect. To commit your changes, select Commit Now from the File menu.
7. Save your workbook by selecting Save from the File menu, if desired.
8. To close the workbook, select Close from the File menu.

Set Measure Availability for Workbook Templates

1. From the File menu, select New.
2. Click the Administration tab.
3. Select Security Administration.
4. Click **OK**.
5. On the Workbook Template Measure Rights worksheet, select each registered measure that should be available for inclusion in the associated workbook template. For measures that should not be included in the associated template, make sure there is no check mark.
6. Changes must be committed to the master database before they take effect. To commit your changes, select Commit Now from the File menu.
7. Save your workbook by selecting Save from the File menu, if desired.
8. To close the workbook, select Close from the File menu.

Assign or Restrict User Access to Measures

1. From the File menu, select New.
2. Click the Administration tab.
3. Select Security Administration.
4. Click **OK**.
5. On the Measure Rights worksheet, for each measure that a user should have access to, select either Read Only or Read/Write from the drop-down list. For measures to which the user should not have access, make sure Denied is selected.
6. Any changes made must be committed to the master database before they take effect. To commit your changes, select Commit Now from the File menu.
7. Save your workbook by selecting Save from the File menu, if desired.
8. To close the workbook, select Close from the File menu.

Change a User's Ability to Modify Dimensions

1. From the File menu, select New.
2. Click the Administration tab.
3. Select Security Administration.
4. Click **OK**.
5. On the Dimension Modification Rights worksheet, select each dimension for which the user needs modification rights. For dimensions that the user should not be able to modify, make sure there is no check mark.
6. Any changes made must be committed to the master database before they take effect. To commit your changes, select Commit Now from the File menu.
7. Save your workbook by selecting Save from the File menu, if desired.
8. To close the workbook, select Close from the File menu.

Set or Modify Access to Positions (if position level security has been enabled)

1. From the File menu, select New.
2. Select the Administration tab.
3. Select Security Administration.
4. Click **OK**.
5. Select the worksheet for which security needs to be set or modified: User, User Group, or World.
6. By default, the dimension (level) at which position level security is enabled will be displayed. To manage security at a level above the designated level (only levels above are possible), right-click and "Select Rollup" to view the available dimensions.
7. To grant access to a position, click the checkbox of the cell.

> **Note:** A user must have access at the User, User Group, and World levels to have access to a position.

8. Changes must be committed to the domain before exiting in order for them to take effect.

Limit the Number of Workbooks that a User can Save

1. From the File menu, select New.
2. Select the Administration tab.
3. Select Security Administration.
4. Click **OK**.
5. Select the worksheet for which the limit will be set: User / Template, Group / Template, or Template.
6. Set the values as necessary.
7. Commit the data to the domain before exiting.

4

Hierarchy Maintenance

Overview

Hierarchy Maintenance Workbook

Oracle Retail Predictive Solutions provide the ability to set up and maintain user-named and user-defined dimensions within hierarchies. Hierarchy Maintenance is the means by which custom-created dimensions within a hierarchy can be established and maintained through the application interface in order to meet individual business needs.

When Oracle Retail Predictive Solutions are installed, implementation scripts define the dimensions and hierarchical structures specific to your organization. For example, the system can be built to recognize that SKUs roll up into styles, styles roll up into product classes, and so on within the product hierarchy. Occasionally, you might want to group products according to some ad hoc personal design to suit a particular business need. You can group arbitrary items in a hierarchy to use in functions such as forecasting, replenishment, and measure analysis. These user-defined groupings act as normal dimensional levels. In other words, they allow you to roll data up from lower levels of aggregation along the hierarchical paths that you define.

For example, suppose that your experience has shown that the accuracy of forecasts for your top 50 products (A products) reflects the relative accuracy of all forecasts. Therefore, you would like to group elements within a user-defined dimension as the top 50 products by designating them 'A Products.' Then, when you select products in a wizard or look at data in a worksheet, you can change the rollup to your user-defined dimension to see your top 50 products grouped together.

Note: Your collection of 50 products may comprise elements from a wide range of product classes or departments, and your grouping scheme may have little to do with the normal dimensional relationships of these items in the product hierarchy.

Note: The group of items you designate as 'A Products' may change over time as consumer preferences change. From this example, you see that user-defined dimensions can be used to create any ad hoc groupings to provide additional support in analyzing, selecting, or summarizing data in Demand Forecasting. The Hierarchy Maintenance interface allows you to change the nature of the groupings as required.

Note: The number and names of user-definable dimensions are set by your company when an RPAS-based solution is initially installed. The positions within each dimension and their associated labels can be altered and maintained through the hierarchy maintenance process.

Keep in mind that any hierarchy in RPAS can have user-defined dimensions within it as long as they are set up by your company at the time of installation. The examples in this section refer to the Product hierarchy, but other hierarchies could be maintained in the same way.

Hierarchy Maintenance Example

Suppose you want to designate SKUs in your product hierarchy as either A, B, or C products so that you can group these items together when you view information, such as forecasting, replenishment, or measure analysis reports.

To do this, you need to maintain a user-defined dimension that will allow you to map the SKUs to the various positions of your classification scheme (A, B, or C). The user-defined dimension used in the following example is named Product Status. To maintain this user-defined dimension, you use the Hierarchy Maintenance Wizard.

Hierarchy Maintenance Wizard

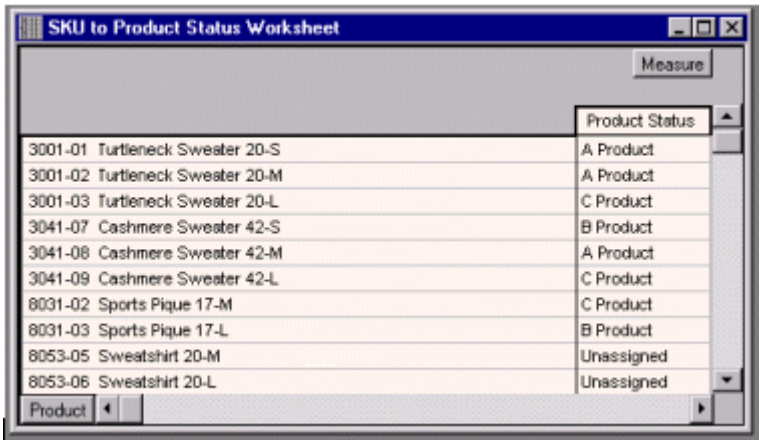
The first step in maintaining hierarchies is to access the Hierarchy Maintenance Wizard. In this wizard, you select the SKUs that will be mapped to the various positions of your user-defined dimension. Responses to prompts in the wizard are used to format a new Hierarchy Maintenance workbook.

Hierarchy Maintenance Worksheet

The Hierarchy Maintenance worksheet displays the position assignment fields for your selected custom dimension. You can edit the cells associated with your custom dimension as required.

Returning to the example dimension Product Status, you want to classify each selected SKU in your workbook as an A Product, a B Product, or a C Product. This example provides only three positions, or values, in the Product Status dimension; however, you can enter any character string in an individual SKU's Product Status cell. This new string will be treated as a separate user-defined grouping. If this is the first time a particular SKU has been mapped to the Product Status dimension, the label assigned to that SKU will not yet be defined. The Product Status field is automatically filled with 'Unassigned.'

Assign labels to each product with regard to the Product Status dimension. In the following example, products that were previously 'Unassigned' are now designated as A, B, or C Products.

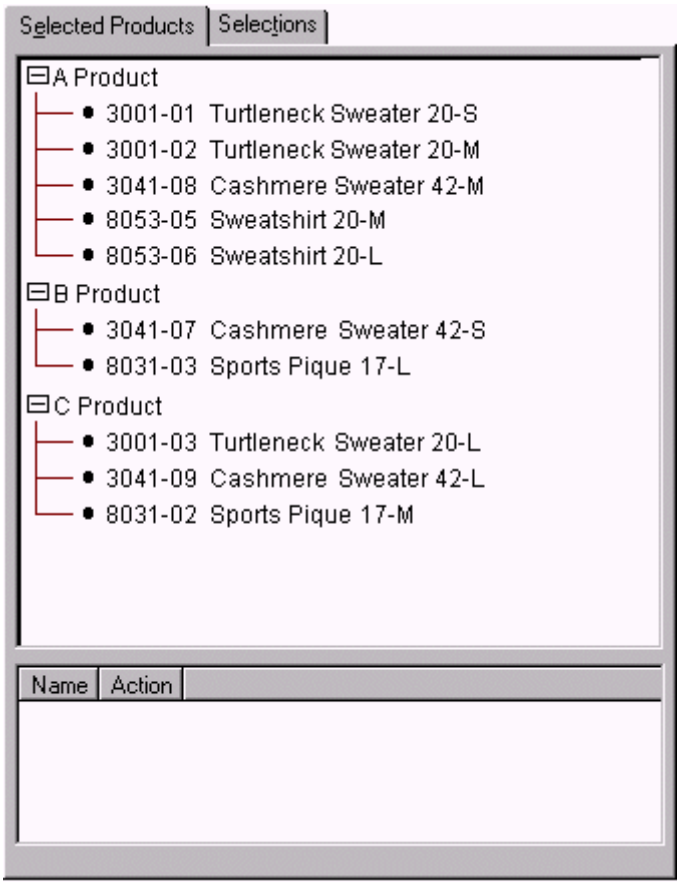


Note: The Oracle Retail Predictive Solutions system is case-sensitive when a new position name (label) is entered in the Hierarchy Maintenance workbook. After the workbook is committed, the typing of the group name is not case-sensitive. For example, "B Product" can later be entered as "b product" after the "B Product" group label has been committed.

After making the A, B, or C Product designations for the selected SKUs, you must commit the workbook for any changes to take effect.

For this example, labels have now been assigned to the various positions within the Product Status dimension, and selected products in the product hierarchy have been classified with regard to the custom dimension. Demand Forecasting treats Product Status, a user-defined dimension, as a normal dimensional level within the product hierarchy.

The following figure displays the results when, in a wizard, you access a quick menu and change the rollup to the Product Status dimension. The products shown here are classified according to the position values (A Product, B Product, or C Product) that you assigned while maintaining the Product Status dimension.



Procedures

Access Hierarchy Maintenance

> **Note:** If a solution is built in a Global Domain environment, it is only required and possible to perform all the administrative activities in this section in the "master" domain.

1. Select Open from the File menu to bypass the Hierarchy Maintenance wizard, and open an existing Hierarchy Maintenance workbook.

   OR

1. Select New from the File menu.
2. Select the Administration tab to display the list of Administration templates.
3. Select Hierarchy Maintenance.
4. Click **OK**.
5. Select the hierarchy for which you want to specify a user-defined dimension (for example, Product or Location). Only the hierarchies that have been set up to contain user-defined dimensions are represented here.
6. Click **Next**.
7. Select the user-defined dimension to be updated. The number and names of available custom dimensions are set at installation.
8. Click **Next**.
9. On the Available side of the selection wizard, choose the items to be mapped to positions within your custom dimension.
10. Click the right arrow button to move them to the Selected side.
11. When you have chosen all items to appear in your workbook, click **Finish**.

Maintain a User-Defined Dimension within a Hierarchy

Use this procedure to assign product or location items to custom-defined positions within a specialized dimension. Custom-created dimensions are distinct from those in the standard hierarchical roll-ups established at your system implementation. You can use these dimensions like normal Demand Forecasting levels, aggregating data along these new hierarchical paths.

1. Select New from the File menu.
2. Select the Administration tab to display the list of Administration templates.
3. Select Hierarchy Maintenance.
4. Click **OK**.
5. Select the hierarchy for which you want to specify a user-defined dimension (for example, Product or Location). Only the hierarchies that have been set up to contain user-defined dimensions are represented here.
6. Click **Next**.
7. Select the user-defined dimension to be updated. The number and names of available custom dimensions are set at installation.
8. Click **Next**.

9. On the Available side of the selection wizard, choose the items to be mapped to positions within your custom dimension.
10. Click the right arrow button to move them to the Selected side.
11. When you have chosen all items to appear in your workbook, click Finish.
12. The Hierarchy Maintenance workbook is displayed. In the position assignment field for your custom dimension, assign a value to each product or location position in your workbook. You can enter any text string in a cell. Each unique string will be treated as a separate user-defined position within the custom dimension.
13. Select Commit Now from the File menu to commit your changes to the master database. If desired, you may also save your workbook by selecting Save from the File menu.
14. To close the workbook, select Close from the File menu.

5

Measure Analysis

Overview

Measure Analysis Workbook

The Measure Analysis workbook template allows you to view data associated with any registered measure in the Oracle Retail Predictive Solutions applications, such as actual sales data for specified product/location/calendar combinations. You may also use the Measure Analysis workbook to edit values for read-write measures.

Although a common use of the Measure Analysis workbook is to view actual sales data, the workbook is not restricted to presenting sales data alone. You can view any data loaded into the Oracle Retail Predictive Solutions master database, such as selling prices, shipments, and orders. The Measure Analysis Wizard provides you with a list of all registered measures from which to select. You simply choose the measures to be displayed in the new workbook.

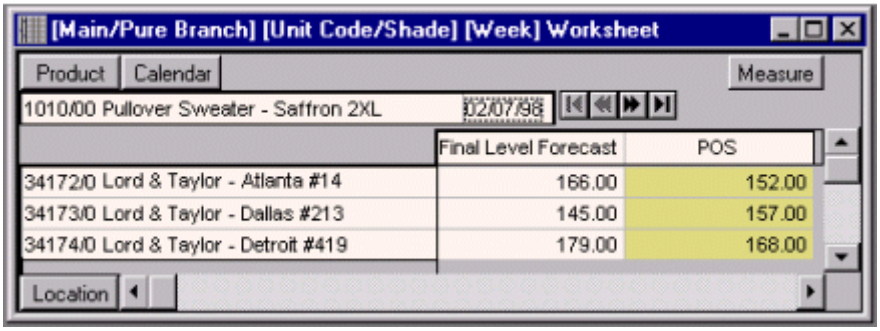
Measure Analysis Wizard

The Measure Analysis Wizard guides you through the process of creating a new Measure Analysis workbook in which you can view data associated with any registered measure(s) in Demand Forecasting.

Measure Analysis Worksheet

The Measure Analysis workbook displays data associated with selected registered measures, and it contains one type of worksheet, the Measure Analysis worksheet. A separate worksheet will exist for each distinct product/location intersection that is associated with the measures selected in the wizard.

The Measure Analysis worksheet allows you to view the chosen measure data for the selected products, locations, and time periods. Each Measure Analysis worksheet is displayed at a different dimensional intersection, depending on the measure selections made in the wizard. This dimensional intersection is shown in the worksheet title bar.



The example above shows a Measure Analysis worksheet that displays Final Level Forecast data and POS data for three locations. The location/product/calendar dimensional intersection of this worksheet, as shown in the title bar, is [Main/Pure Branch][Unit Code/Shade][Week]. The Final Level Forecast measure, because it is registered as a read/write measure, can be edited in this worksheet. POS values cannot be changed because this measure is registered as read only.

Procedures

Access Measure Analysis

1. Select Open from the File menu to bypass the Measure Analysis wizard and open an existing Measure Analysis workbook.

OR

1. Select New from the File menu.
2. On the Analysis tab, select Measure Analysis.
3. Click **OK**.

Review and Edit Sales or Other Registered Measure Data

1. To open an existing Measure Analysis workbook: select Open from the File menu, double-click on the workbook to be opened, and go to step 8.

OR

1. To open a new workbook, select New from the File menu.
2. On the Analysis tab, select Measure Analysis.
3. Click **OK**.
4. The Measure Analysis Wizard opens and prompts you to select the registered measures to be displayed in the new workbook. Use Ctrl-Click and/or Shift-Click to select multiple measures.
5. Click **Next**.
6. Select the dimensional level at which you want to view measure data.
7. Click **Next**.

> **Note:** This screen only appears if there is more than one level possible for one of your selected measures. A separate wizard screen will be displayed for every measure chosen in step 3 if multiple aggregation levels are available.
>
> **Tip:** Selecting a pre-aggregated level (a level other than base level) allows you to view aggregated sales data faster than if you select the base level and later aggregate data from within the workbook itself. Building the workbook with pre-aggregated data can save time and lessen the need for data transfers later.

8. Select all calendar periods for which you want to view data.
9. Click **Next**.
10. Select the locations whose measure data you want to view.
11. Click **Next**.

12. Select the products whose measure data you want to view.
13. Click **Next**.
14. Click **Finish** to open the Measure Analysis workbook.
15. On the Measure Analysis Worksheet(s), view the stored data associated with the measures, products, locations, and calendar periods you selected. Make any changes as required.
16. If desired, save your workbook by selecting a Save option from the File menu.
17. To close the workbook, select Close from the File menu.

6

Workbook Auto Build Maintenance

Overview

The Workbook Auto Build feature allows users to set up workbook builds to take place on a regular basis during nightly batch runs. Workbooks to be built in this way are added to the auto build queue. Because the workbook build process is automated, users are spared the processing time required to regularly enter the same wizard selections each time a new workbook is built. And because the build process occurs overnight, users are spared the wait time associated with constructing new workbooks.

The Workbook Auto Build feature works through the Workbook Auto Build Maintenance Wizard.

Workbook Auto Build Maintenance Wizard

The Workbook Auto Build Maintenance wizard steps you through the processes of adding and/or deleting workbooks from the auto build queue.

Procedures

Accessing the Auto Build Maintenance Workbook

1. Select New from the File menu.
2. Select the Administration tab.
3. Highlight Auto Workbook Maintenance.
4. Click **OK**.

Add a Workbook to the Auto Build Queue

Workbooks in this queue are designated to be built automatically on a specified regular basis as part of the nightly batch run.

1. Select New from the File menu.
2. Select the Administration tab.
3. Highlight Auto Workbook Maintenance.
4. Click **OK**.
5. From the task list, select Add Workbook.
6. Click **Next**.
7. Select a workbook template type.
8. Click **Next**.
9. Select an owner for the workbook.
10. Click **Next**.
11. Fill in the workbook Build Label, the Build Frequency (in days) with which the workbook should be built, and the Next Build Date.
12. Specify the Saved Access for the workbook: select User, Group, or World.

13. Select the group that owns the workbook. You can choose from the list of groups to which you belong.
14. Click **Next** to initialize the wizard for the workbook template selected in step 5 above. The choices you make are saved under the name you specified for the Build Label.

Delete a Workbook from the Auto Build Queue

1. Select New from the File menu.
2. Select the Administration tab.
3. Highlight Auto Workbook Maintenance.
4. Click **OK**.
5. From the task list, select Delete Workbooks.
6. Click **Next**.
7. Select the workbook or workbooks to delete from the auto build queue.
8. Click **Finish** to delete the workbooks from the Auto Workbook Build queue.

7

Translation Administration

Overview

All of the Oracle Retail Predictive Solutions can be presented in multiple languages. Every product, location, and calendar position can be displayed in multiple languages, as can messages presented through the client. Translation administration is the process by which administrators maintain the translation tables for system text. This includes workbook template labels, template group labels, user group labels, wizard instructions, pick list options, and error messages.

The Translation Administration workbook contains worksheets for translating text used in measure labels, workbook template names, template group names, user group labels, and general areas (for instance, wizard instructions, pick list options, and error messages).

The worksheets contained in the Translation Administration workbook include:

- General Translations worksheet
- Measure Translations worksheet
- Template Translations worksheet
- Template Group Translations worksheet
- User Group Translations worksheet

General Translations Worksheet

The General Translations worksheet allows you to view and edit the translations of text strings found throughout the Oracle Retail Predictive Solutions. Translations into each of the system's alternative languages are supported for all wizard instructions, pick list options, system warnings and error messages, menu options, and so on.

Measure Translations Worksheet

The Measure Translations worksheet allows you to view and edit the translations of measure labels. Translations are supported for each of the system's allowable alternative languages.

Template Translations Worksheet

The Template Translation worksheet allows you to view and edit the translations of workbook template names. Translations are supported for each of the system's allowable alternative languages.

Template Group Translations Worksheet

The Template Group Translations worksheet allows you to view and edit the translations of template group names. Translations are supported for each of the system's allowable alternative languages. Translations in this worksheet affect the labels on the tabs that appear in the File > New dialog. For example (in English), Administration, Analysis, and Predict.

User Group Translations Worksheet

The User Group Translations worksheet allows you to view and edit the translations of user group labels. Translations are supported for each of the system's allowable alternative languages. The list of user groups includes the Administration, Default, and Internal user groups, plus any other user group names set up by the system administrator. For products in the Oracle Retail Predictive Planning Suite, the list of user groups also includes the various planning roles.

Procedures

Accessing the Translation Administration Workbook

1. Select New from the File menu.
2. Select the Administration tab.
3. Highlight Translation Administration.
4. Click **OK**.

Edit the Foreign Language Translations of Labels and System Text

1. Select New from the File menu.
2. Select the Administration tab.
3. Highlight Translation Administration.
4. Click **OK**.
5. Maximize the appropriate worksheet.
   - To translate measure labels, maximize the R_TRANS_MEAS worksheet.
   - To translate workbook template names, maximize the R_TRANS_WBT worksheet.
   - To translate template group names, maximize the R_TRANS_WBTG Translations worksheet.
   - To translate user group labels, maximize the R_TRANS_GRP worksheet.
   - For all other translations, such as wizard instructions, list options, error messages, and menu choices, maximize the R_TRANS_MSG worksheet.
6. At the intersection of the language and the text value to be translated, enter the correct translated text. Repeat this step as required for all text strings to be edited.
7. On the File menu, select Commit Now.
8. On the File menu, select Close. Select a close option from the Close dialog.

Process for using the General Translations Worksheet

Enter the correct translated text at the intersection of the language and the string value to be translated.

Process for using the Measure Translations Worksheet

Enter the correct translated text at the intersection of the language and the measure label to be translated.

Process for using the Template Translations Worksheet

Enter the correct translated text at the intersection of the language and the workbook template name to be translated.

Process for using the Template Group Translations Worksheet

Enter the correct translated text at the intersection of the language and the template group label to be translated.

Process for using the User Group Translation Worksheet

Enter the correct translated text at the intersection of the language and the user group label to be translated.

8

Commit as Soon as Possible

Overview

Commit As Soon As Possible (Commit ASAP) allows users to schedule the commit process of workbook data so that it executes as soon as all the system resources are available. Commit ASAP is an option in the File menu of the RPAS client. Procedures for using Commit ASAP are provided in the "RPAS 11.2 Users Guide."

Commit ASAP takes a copy of the data to be committed. Unlike Commit Later, which adds a workbook commit process to a queue that is run in batch, the data that is eventually committed is the data that was present at the time the commit instruction was issued. With Commit Later, if the user makes further changes to the workbook and saves that workbook before the batch commit process is run, those changes will also get committed.

Using Commit ASAP

After attempting to commit a workbook using Commit ASAP (File\Commit ASAP), you will see a message in the client that the workbook has been scheduled for a commit. You can continue with your work. The system will then try to commit the workbook as soon as it can, taking into account any other scheduled commits. If the commit cannot be done prior to the domain's Commit ASAP deadline, it will be canceled and listed as failed.

There are four states for commit processes to be added to the Commit ASAP queue.

- Pending – The commit process is queued up to take place at some point in the future.
- Committing – The workbook is currently being committed.
- Success – The commit succeeded.
- Failed – The commit failed.

The status of each commit ASAP process can be viewed by using a dialog window called "**Commit Status**" from the File menu. This dialog window displays all of the Commit ASAP processes with their respective status for all processes that have not been purged (see below). This dialog can be used to sort the tasks based on any of the columns.

You can filter the entries in a variety of ways. If the checkbox **All Users** is not checked, you will only see your entries. If it is checked, you will see the entries for all users. The checkboxes in the **Status To Display** group allow you to filter the output so that you see only the processes with the specified statuses. The window can be updated by using the **Refresh** button. The dialog remembers the settings based on the last use.

Note: If a user attempts to commit a workbook ASAP that already has a process in the queue, the original processes will be removed from the queue. That means that there can only ever be one pending commit ASAP in the queue for a given workbook/user/template name combination.

Note: Workbooks must have been saved at least once before attempting a Commit ASAP. A workbook has not been saved if the label says "untitled."

Managing the Workbook Queue – showWorkbookQueues

The RPAS utility **showWorkbookQueues** is used for viewing the status of Commit ASAP processes and for purging entries in the Commit ASAP status window. The usage of this utility follows below.

The purge option requires a date before which entries will be removed, as well as specification for which entries to remove: succeeded, failed, or both.

Usage

```
showWorkbookQueues -version
showWorkbookQueues -d domainPath –show
[all|pending|waiting|working|success|failed]*
showWorkbookQueues -d domainPath -purge date [success | failed]*
```

The following table provides descriptions of the arguments used by the `showWorkbookQueues` statements.

Argument	Description
–version	Prints the RPAS version, revision, and build information of the utility.
-d domainPath	Specifies the path to the domain.
-show	Lists the contents of the queue in the order in which the parameter is specified. Possible values: all, pending, waiting, working, success, failed.
All	Used with the –show parameter. This lists all of the workbooks in all statuses.
Pending	Used with the –show parameter. This lists all workbooks that are waiting to be committed.
Waiting	For Oracle Retail development use only.
Success	Used with the –show parameter. This lists all workbooks that have been successfully committed.
Failed	Used with the –show parameter. This lists all workbooks that did not successfully commit.
-purge date	Purges entries in the Commit ASAP status window. Entries before the date provided will be removed. The date should be a string of the following DateTime format: YYYYMMDDHHmm For example “200406071529” equals June 7, 2004 3:29 PM. Administrator must select to purge commit processes that either succeeded or failed.

Commit ASAP Settings – configCommitAsap

There are two settings for Commit ASAP that are managed by an administrator. Both are set using the utility **configCommitAsap**.

- Maximum number of simultaneous commit processes (property **MaxProcesses**, default value is 4).
- Deadline for which all pending processes must be completed, after which they will be cancelled and marked as failed.
  This deadline will likely be used by administrators before beginning nightly batch processes (property deadline, default value is 00:01 [meaning 12:01 AM], in 24-hour time).

  A commit process that starts before the deadline is reached will be processed. Commit requests that were in the queue before the deadline that did not get processed will be cancelled and marked as failed. Commit requests added to the queue after the deadline will use the deadline of the following day.

Usage

```
configCommitAsap -d pathToDomain [-maxProcs numProcs]
[-deadline time] [-display]
```

The following table provides descriptions of the arguments used by the `configCommitAsap` statements.

Argument	Description
–version	Prints the RPAS version, revision, and build information of the utility.
-maxProcs numProcs	Sets the maximum number of concurrent commit processes where numProcs is an integer greater than 0. Workbooks can be committed in parallel if they do not require access to the same measure databases. If they do share databases, they will be committed sequentially.
-deadline time	The time of the day when all outstanding commit ASAP operations will timeout. If a commit ASAP operation is submitted after this time, it will not timeout until the deadline time on the next day. This string must have the following format: HH:MM For example “13:30” refers to 1:30 PM.
-display	Displays the current commit ASAP settings.
-loglevel level	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.
-noheader	To disable timestamp header use.

Logging and Technical Information

A log file is available in the Commit ASAP directory that should be checked if a user reports an error with a Commit ASAP submission. The file is named **rpasServer.log** and is in the following directory: <Path to domain>/commitAsapQueue.

Another log file is generated for each Commit ASAP process and stored in a user's directory (users/<userid>/asapLogs). The format of the log file name is "orig_<original workbook name>asap_<temporary workbook name>.log." RPAS creates a temporary workbook in this process to capture the snapshot of the data that needs to be committed. Temporary workbooks are never viewed by a user. An administrator can use this log if something does not properly commit.

> **Note:** These "snapshot" workbooks cannot be viewed or used in the RPAS client.

An example of this log file is orig_t1_asap_t5 where "t1" is the name of the original workbook and "t5" is the name of the snapshot workbook.

The following directories are used to store the copies of the workbook as they are processed through the system:

- **pending** directory – Contains one file per submitted commit ASAP that has not yet been processed. These files are, in general, binary and cannot be easily read.
- **working** directory – Contains one file per submitted commit ASAP that is currently in the commit process.
- **success** directory – Contains one file per submitted commit ASAP that has successfully completed its commit process.
- **failed** directory – Contains one file per submitted commit ASAP that either had a failure during its commit process or could not be committed prior to the deadline.
- **unknown** directory – If the Commit ASAP process detects a corrupted queue file, a message gets logged and the file gets moved into the unknown directory.

9

Batch Processes and RPAS Utilities

Overview

Included with an RPAS installation is a collection of stand-alone executables and scripts that are used for a variety of operations. RPAS utilities are run directly against a domain. If in a Global Domain environment, most utilities can only be run on the master domain. RPAS utilities can be categorized into the following groupings:

- Hierarchy management – The loading and refreshing of hierarchies, and the process of updating the data structures in the domain to reflect hierarchy changes.
- Measure data – Utilities for loading, exporting, and moving data within and between domains.
- Miscellaneous – A variety of utilities for performing certain procedures in batch and for setting a number of parameters on an environment/domain.
- Information RPAS utilities – A variety of utilities that retrieve information about a domain, data, the RPAS server code, or an object used by the server.

Using Shell Scripts to Run Batch Processes

Batch processes should be written using scripts that call the RPAS 11 binaries found in the $RPAS_HOME/bin/ directory. Any log files generated by scripts will be in the [DOM]/scripts/err/ directory. Examples of tools include Korn shell, Python, and Perl.

A Sample Shell Script

The following is a sample shell script that loads the product and location hierarchies into a domain. It is assumed that this script is invoked from the [DOM]/scripts/ directory.

```
1  #!/bin/ksh
2  loadHier –d .. –load prod > ./err/loadhier.prod.log
3  loadHier –d .. –load loc >> ./err/loadhier.loc.log
```

Line 1 defines the shell that will execute the script. In this example, it is defined to be the Korn shell. Therefore, this script will always be executed from the Korn shell even if the user's shell is different.

Lines 2 and 3 call the loadHier utility to load the latest product and location hierarchy information. Depending on the batch process to be performed by the shell script, lines 2 and 3 can be replaced by one or more lines to call one or more RPAS 11 utilities.

Running Utilities in a Global Domain Environment

In a Global Domain environment, most of the RPAS utilities can only be run on the master domain. These changes are then propagated to the local domains or stored centrally for access by each local domain.

The following utilities are centralized and can only be run on the master domain in Global Domain environments:

- Alerts (**alertmgr**)
  a. Alerts registered in the global domain will be propagated to local domains.

     Note: Alerts registered in local domains will not (and do not have to) be included in the global domain.

  b. Utility for finding alerts (for instance, alert finder) will be updated to run against the local domains and collate results in the global domain for a centralized view of the alert results.
- Loading hierarchies (**loadhier**) – Hierarchy information must be centrally administered in the Global Domain and replicated to the local domains.
- Loading measure data (**loadmeasure**) – Utility for loading input data into measures in a domain. The utility can load a single input file that contains data for multiple local domains. The utility splits the data for loading into the local domains, and the utility can be run in parallel.
- Exporting measure data (**exportdata**) – Utility for exporting measure data from a domain to flat files. The utility is called from the master domain and included the data from one or more local domains as specified by the call to the utility.
- Users (**usermgr**) – The back-end utility for managing users will propagate changes from the global domain to local domains. Ultimately, this will also be updated for the RPAS Administrative Template "User Administration."
- Domain properties (**domainprop**) – Manipulating properties, such as specifying password properties, and locking user accounts.
- Workbook Manager (**wbmgr**) – For listing and removing workbooks in a domain. This utility will be able to be run in the master domain and local domains. When run in the master domain, an administrator will be able to manage workbooks in local domains as well.
- **changedomainsparseness –** Changing the sparsity of a domain to/from sparse and hypersparse.

Common Information and Parameters for RPAS Utilities

A number of standard parameters are available for most RPAS utilities. The following table provides a description of these parameters. Check the usage of a specific utility to verify whether or not it is available.

Argument	Description
–version	Use this argument to get the version information of the utility (for instance, RPAS 11.2.0). It does not require –d domainPath.
–d pathtodomain	Common to most utility this specifies the path to the domain against which the utility will run or from which data will be used.
–loglevel	Use this argument to set the logger verbosity level. Possible values: none, error, warning, information, or all (*see additional information below).
–n	Certain utilities contain this parameter to perform a dry run. Using this option will show the administrator what would change, but makes no actual changes to the system or data. See the usage of a specific utility to see whether this option is applicable.
–noheader	To disable the use of a timestamp in the header of the log file.
–help | –? | –usage	Any of these arguments will output the usage of the utility to the terminal. This can also be accomplished by running the utility with no arguments.

Logger verbosity levels determine how much information is generated on the terminal when running a given utility. An administrator can set these levels for each RPAS utility.

- none – There should be no output if the utility successfully executes.
- error – Outputs only error messages.
- warning – Outputs warnings in addition to error messages.
- information – Outputs status and progress of the operation in addition to the error and warning messages.
- all – Outputs all available information generated by the utility, including error, warning, and informational messages.

Each of the lines that contain the above types of feedback is normally preceded with a code that indicates what type of information is being output. Each code should have an angle bracket ("<") in front of it. E indicates the message is an error. W indicates the message is a warning. I indicates the message is informational.

Logging Batch Processes

RPAS batch processes may run for a long time. Therefore, it can be difficult to know the step that is currently being performed. Fortunately, the RPAS server code provides a large amount of detailed output messages from its processes to help the system administrator troubleshoot, benchmark, and estimate job completion times. The batch processing shell scripts direct those messages to output files (log and error files) located in [DOM]/scripts/err. These output files typically use the following naming convention:

batchProcess.date&timestamp.fileType

where:

BatchProcess = The name of the shell script or the corresponding batch process.

data&timestamp = The date and time when the script is executed in the format yyyymmddhhmmss.

fileType = The type of the file that can be "log" or "err" to indicate whether it is a log file or an error file.

Configuration Tools Log Files

For the RPAS Configuration Tools, information is logged in the files **stderr.txt** and **stdout.txt**, which are located in the **bin** sub-directory of the Tools directory. If you experience a problem with the configuration tools, you should send these two files to Oracle Retail Customer Care along with a description of the problem.

Error Files

Error files are usually generated during data loading. These files include a list of bad records followed by the total number of records read and related information. Part of the output from a typical data loading batch job is as follows:

```
Loading array TEMP in nonoverlay mode. Zero values are loaded.
Skipping cell with invalid position:
    INFO    1
    DAY 1996D364
    SKU SKU_00726828
    STR STR_0107
Skipping cell with invalid position:
    INFO    1
    DAY 1996D364
    SKU SKU_00726828
    STR STR_0201
Skipping cell with invalid position:
    INFO    1
    DAY 1996D364
    SKU SKU_00726828
    STR STR_4008
Skipping cell with invalid position:
    INFO    1
    DAY 1996D364
    SKU SKU_00726828
    STR STR_4009
Skipping cell with invalid position:
    INFO    1
    DAY 1997D6
    SKU SKU_00726828
    STR STR_0107
Skipping cell with invalid position:
    INFO    1
```

```
    DAY 1997D6
    SKU SKU_00726828
    STR STR_0201
Array -- TEMP. Load time:   0:02
LoadComplete -- Records read: 4240, Total cell updates: 4240
  New cells created: 4028
  Cells with invalid positions: 212
  Cell updates: 0
```

10

Hierarchy Management

Overview

There are a number of key concepts and processes that are critical to the hierarchy management process:

- Hierarchy structures are loaded into a domain using the **loadHier** utility.
- The process of updating the data structures in the domain is commonly referred to as "reshaping." This process is handled automatically by the system.
- The length of position names is 24 characters or less by default. RPAS provides the ability to increase this length using the **dimensionMgr** utility.
- RPAS provides the ability to have placeholder positions in the domain that can be used when loading new hierarchy positions. Use of "dummy" positions defers the time consuming process of "reshaping" until a scheduled time, thus saving valuable time in the batch window for time-constrained batch processes.
- RPAS can automatically handle the movement of positions and their corresponding data between local domains when their parent-child relationships change and cause such a scenario. This requires the use of dummy positions and is only applicable in a global domain environment.
- Positions at the partition level in a global domain environment can be moved between local domains using the **reconfigGlobalDomainPartitions** utility.
- New local domains can be added to an existing global domain environment using the **reconfigGlobalDomainPartitions** utility.

Placeholder Positions in the Domain

RPAS provides the ability for a domain to contain "dummy" hierarchy positions, which are placeholder positions that allow new positions to be added to a hierarchy without having to update the data structures in the domain (process referred to as "reshaping"). These dummy positions are not visible to applications or users, so they cannot be seen in workbooks. They should not be confused with any form of 'dummy' or 'placeholder' positions that are part of a business process and visible to users, such as in the process of new store or item introductions.

RPAS provides the ability for any dimension, other than those in the calendar hierarchy, to contain dummy positions. The number of dummy positions is a percentage of total positions for a given dimension in the domain. For example, if the SKU dimension of the PROD hierarchy contains 1 million positions; a dummy position buffer of 1% will allow for 10,000 dummy SKU positions. When dummy positions are enabled for a dimension, positions have an 'internal' name (known only to RPAS and not visible to users or applications) and an 'external' name. Dummy positions are positions with an internal name, but no external name. Reshaping is required whenever the collection of internal names changes.

New positions are added to a domain through the hierarchy load process (**loadHier** utility). If a position is new, RPAS will map an existing unused dummy position to the newly added position so that it can be used in the domain without having to "reshape" the domain.

Similarly, as old positions are deleted, the external name for the internal position is removed from the mapping table (and data for the positions is removed from the arrays), and the position effectively become a dummy position. Therefore, deletes can happen without the need to "reshape" the domain.

As dummy positions are consumed, the number of available dummy positions will be reduced. Dummy positions are held in the "buffer," and the process of updating this is "re-buffering." The buffer can be updated automatically, but unpredictably, when required (based on the lower and upper bounds that are defined for a PNI dimension). It is recommended that the rebuffering process is scheduled to ensure that batch process windows are predictable. The manual rebuffering process is executed with the **positionBufferMgr** utility.

Position Repartitioning

Position repartitioning is the automated process of moving positions and all corresponding measure data between local domains. This functionality is only available (and relevant) in Global Domain environments. Positions need to be moved between local domains when they are assigned a new parent that exists in a different local domain. Note that moving positions at the partitioning level is a manual process and requires the use of the **reconfigGlobalDomainPartitions** utility.

For example, imagine Style1 belongs to Sub-Class1 in LocalDomain1. If Style1 is reassigned to be a child of Sub-Class2, which is located in LocalDomain2, RPAS will move the Style1 position, Style1's children (if any), and all corresponding data to LocalDomain2. This process is often referred to as "reclassification" by RPAS customers. RPAS refers to this functionality as "Position Repartitioning" because it technically does not handle the many complex functional requirements of true reclassification as most retailers define the term to mean.

Procedures

Enabling Dummy Positions

The use of dummy positions is enabled in a domain per dimension. Dimensions are enabled for dummy positions using the Configuration Tools both before and after a domain has been built. There are two properties in the Hierarchy Tool, and these properties can be set for each dimension except the calendar hierarchy. These properties are the "PNI Buffer Percent" for both the lower and upper bounds of dummy positions.

New positions are added to a domain by including new positions in the hierarchy data input files, and then running **loadHier**. Newly added positions will be immediately available for use unless all dummy positions have been consumed, which launches an automatic rebuffering process. Positions that have been assigned a new parent that require movement between domains will be automatically processed as part of the position repartitioning process.

Configuring and Scheduling the Rebuffering Process

Administrators need to determine the process by which they wish to rebuffer the dummy positions. Rebuffering can be completed automatically as part of the hierarchy load process when a domain runs out of dummy positions, or it can be scheduled using an RPAS utility.

If it is desired to have a predictable batch window, it is recommended that administrators schedule the rebuffering process rather than use the automated process. Scheduling rebuffering will minimize the possibility that the automated rebuffering process occurs during a critical batch process. The automated rebuffering would then only be used as a backup and run as an exception.

If customers do wish to reschedule the rebuffering, they will need to determine an approach that fits their business and batch processes. One approach might be to schedule all dimensions (with dummy positions enabled) in all domains to be rebuffered together on a weekly or monthly basis when there is a large amount of system down-time. Another approach could be to rebuffer a few domains on a cyclical basis, such as a few a day or week.

There are high and low settings for the dummy position buffer. Within the Configuration Tools, these settings are the "Buffer % Low" and "Buffer % High." When executing a scheduled rebuffering process using the **positionBufferMgr** utility, the buffer for a dimension in a given local domain is updated when the number of dummy positions falls outside the high or low target buffer percentages. The number of dummy positions is calculated by taking the average of the high and low percentages multiplied by the total number of positions of the dimension in the local domain.

Deciding the buffer percentages will depend on a number of criteria. The goal is to have sufficient dummy positions to allow for growth in the local domains without having to execute an automated rebuffering process. Determining the targeted number of dummy positions will be a product of the anticipated growth in a given time period (for instance, 100 SKU's per week) and the frequency of the scheduled rebuffering process (for instance, rebuffering once a week or month).

These buffer settings and rebuffering processes are managed by the **positionBufferMgr** utility.

Loading Hierarchies – loadHier

The **loadHier** utility is used to load and refresh a hierarchy. All hierarchy data files are saved in fixed width (or space delimited) files with a ".dat" file extension. The width of positions (number of characters) is specified in a configuration file before a domain is built. The width of positions can be increased after a domain has been built using the **dimensionMgr** utility or by changing a property in the Configuration Tools and patching the domain.

The hierarchy load utility also handles the process of "reshaping" data in the domain after adding or removing positions. This reshaping process is required to update the underlying data structures to reflect the hierarchical changes; however, note that the use of dummy positions can reduce the frequency of reshaping (see above).

RPAS allows for multiple input files to be loaded for the same hierarchy. The primary use for this is loading hierarchy data for multiple languages. Hierarchy files are in the format hier.dat or hier.dat.language where hier is the name of a registered hierarchy, and multiple hierarchy files can have different language extensions. For example, a run of the

loadHier utility would load **prod.dat**, **prod.dat.japanese**, and **prod.dat.spanish** if those files were available in the input folder.

RPAS automatically generates a backup copy of hierarchy files prior to performing a load for a hierarchy. If any type of error occurs during the load process, the hierarchy is restored from the backup copy.

Usage

```
loadHier –d domainPath –load hiername {–purgeAge purgeage}
{–checkParents} {–noClean}{–loglevel level}
```

The following table provides descriptions of the arguments used by the `loadhier` statement.

Argument	Description
–d domainPath	Specifies the domain in which to load the hierarchy.
–load hierName	Specifies the name of the hierarchy to load and refresh.
–purgeAge purgeage	Specifies the purgeage during loadHier. If not specified, loadHier gets purgeage from domain.
–checkParents	Use this argument to check the parents while loading.
–noClean	If specified input files and the meta data used during load process are not cleaned. It is used only for debugging purposes.
–loadAll	Loads all hierarchy input files (with a .dat file extension) that are located in the input directory of the domain. If necessary, the utility will not execute the reshaping process until all files have been loaded.

Reconfiguring the Partitions of a Global Domain – reconfigGlobalDomainPartitions

It is common for many customers to regularly add, remove, or change the parent-child relationships for positions in hierarchies, most commonly for positions in the product hierarchy. While this movement/reassignment of positions is normally handled automatically within the loadhier utility, a special process must be followed for positions at the partition level of a Global Domain environment.

The RPAS utility **reconfigGlobalDomainPartitions** is used for the following activities in a global domain environment:

- Add new positions at the partition-level.
- Add new local domains to an existing global domain environment.
- Remove existing positions at the partition level.
- Remove local domains (this is automatic if all partition-level positions in a local domain are removed or moved).
- Move existing positions at the partition level (and their data) between local domains.

The following processes must be followed to add, remove, or move positions at the partition level in a Global Domain environment:

- The administrator must be notified in advance that positions at the partition level are being added, removed, or moved.
- The administrator should run the utility **reconfigGlobalDomainPartitions** to by specifying the sub-domain to which the positions do or will belong.
- This utility calls the loadHier utility at the end of the reconfiguration process to apply the hierarchy changes to the domain. When adding positions (using the –add argument) an updated hierarchy file must be available in the input directory when the **reconfigGlobalDomainPartitions** utility is called. Otherwise the utility will fail.

Updated hierarchy files are not required to remove (using the –remove argument) or move positions (using the –move argument).

> **Note:** The use of this utility is only required for positions at the partition level. Positions below the partition level can be added, removed, or moved between local domains by loading a modified hierarchy input file with these changes.

Usage

```
reconfigGlobalDomainPartitions –d pathToMasterDomain –add posName1,posName2, ... –sub pathToSubDomain
reconfigGlobalDomainPartitions –d pathToMasterDomain –remove posName1, posName2, ...
reconfigGlobalDomainPartitions –d pathtoMasterDomain –move posName1,posName2, ... –sub pathToSubDomain
reconfigGlobalDomainPartitions –d pathToMasterDomain –input pathToInputDir
```

The following table provides descriptions of the arguments used by the `reconfigGlobalDomainPartitions` statements.

Argument	Description
–d pathToMasterDomain	Specifies the path to the master domain in a Global Domain environment.
–add posName1, posName2, …	Adds one or more positions at the partition level to a specified local domain. The path to the local domain must follow the list of positions to add, using the –sub argument. If the specified path is to a local domain that does not yet exist, the system will create a new local domain with the specified positions at the partition level. This argument cannot be used with –remove or –input.
–remove posName1, posName2, ...	Removes the designated positions from the local domain to which the positions belong. The path to the local domain does not need to be specified with this argument. The local domain will be deleted if all the positions at the partition level in a local domain are removed. This argument cannot be used with –add or –input.
–move posName1, posName2, ...	Moves the specified positions at the partition level from the current domain in which the positions are located to the specified local domain. This argument requires specification of the –sub argument. To move positions, all dimensions below the partition level must be enabled for dummy positions.
–sub pathToSubDomain	Specifies the path to the local domain to which positions are being added or the destination local domain for positions being moved. This argument is required for the –add argument and –move argument.

Argument	Description
–input pathToInputDir	Specifies the path to the input directory that contains an xml configuration file (reconfigpartdim.xml) to specify positions to either add or move. The file must have all the information to run the process including the command name, position names to add or move, and paths to the local domains. This option is useful for adding or moving positions to multiple local domains. This argument does not handle both adding and moving in the same call. This argument cannot be used with –add or –remove.

Using an Input File

When using the –input argument, the file must be in a particular format and must contain the "add" or "move" commands, the path to each local domain to which positions are being added or the destination for positions being moved, and the list of positions for each local domain. The file must be XML and named "**reconfigpartdim.xml**".

> **Note:** The –input argument only supports the addition or movement of positions.

Below is the required format of the input file:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
  <rpas>
      <command name="command_name">
          <subdomain>
              <subpath>path_to_local_domain_1</subpath>
              <subpositions>sample_pos_1</subpositions>
          </subdomain>
          <subdomain>
              <subpath>path_to_local_domain_2</subpath>
              <subpositions>sample_pos_2,sample_pos_3</subpositions>
          </subdomain>
      </command>
  </rpas>
```

> **Note:** The entries in bold are the parameters that must be specified in the input file.

The following table provides descriptions of the arguments used by the input file.

Argument	Description
command_name	Valid values are "add" or "move."
path_to_local_domain	Path to the local domain to which positions are being added or moved.
sample_pos	One or more positions that are being added or moved to the designated local domains.

Notes, Assumptions, and Limitations

- Position names are separated by commas and must be valid **external** position names without the prefix of a dimension.
- The utility backs up the required data and will automatically restore the domains to the original state in case of failure.
- In a single call to the utility without using the –input argument, positions can only be added or removed or moved. That is, the –add,–remove, and –move arguments cannot be mixed in the same call.
- Multiple positions can be added or moved to a single local domain in a single call to the utility using the –add or –move option, respectively.
- Multiple positions can be added or moved to multiple local domains in a single call to the utility using the –input option.
- When adding positions at the partition level, an updated hierarchy file must be available in the input directory when running this utility as the loadhier utility is called after adding positions. If the updated hierarchy file is not present in the input directory when attempting to add positions, the utility will fail.
- No updated hierarchy file is required when moving or removing positions. If a hierarchy file is in the input directory, the utility will back up this file.
- A log file (loadHier.log) will be created in the root directory if loadHier fails.

Updating or Reporting the Dummy Position Buffer – positionBufferMgr

The position buffer manager is used to manually rebuffer dummy positions in the domain or to report on the status of dummy positions. This utility can only be used if dummy positions have been enabled in the domain environment

Usage

```
positionBufferMgr -d pathtodomain [-rebuffer|-report] {-hier hierName}* {-
partitionPositions "pos1,pos2..."}
```

The following table provides descriptions of the arguments used by the `positionBufferMgr` statement.

Argument	Description
–d pathToDomain	Specifies the path to the domain.
–rebuffer	Adjust all dimensions of the specified hierarchy to have the number of dummy positions based on the configuration of the dimension for the environment. If a hierarchy is not specified with this argument, the utility will run on all hierarchies other than calendar configured to have dummy positions.
–report	Report the count status of positions for all dimensions of the provided hierarchy. If a hierarchy is not specified with this argument the utility will run on all hierarchies configured to have dummy positions other than calendar.
–hier hiername	The hierarchy that will be rebuffered. If no hierarchy is specified, the utility will rebuffer or report on all hierarchies configured to have dummy positions other than calendar.
–partitionPositions "pos1, pos2, …"	By specifying positions at the partition level, the utility will only rebuffer local domains to which the specified positions belong. Positions specified that are not at the partition level will be ignored.

Renaming Positions – renamePositions

RPAS provides the ability to change the name of a position using an RPAS utility named **renamePositions**. Positions that are to be renamed should be included in a hierarchy data file that is located in a designated input directory (specified when running utility) and that follows the naming convention **hierName.rn.dat** (for instance, "prod.rn.dat").

After the hierarchy data file(s) has been updated and placed in the designated location, an administrator must run the **renamePositions** utility.

Usage

```
renamePositions –d domainPath –i inputDirectory –hier hierName {–log logFileName}
{–n}
```

The following table provides descriptions of the arguments used by the `renamePositions` statement.

Argument	Description
–d domainPath	Specifies the full path to the domain.
–i inputDirectory	Input directory where input file with positions to rename is located. Utility looks for hierarchy data files with "rn" between hierarchy name and .dat extension (for instance, prod.rn.dat).
–hier hierName	Hierarchy for which positions are being renamed.
–log logFileName	Optional parameter to generate log file to file name other than default. The default file name is hierRename.log.
–n	Does not apply changes, but generates a report that identifies changes that would be applied.

Notes:

The `–n` is a dry run, which means that it does everything as a true run (for instance, writing to a log file) except that it does not actually commit the changes to the domain.

`–log` is an optional argument that you can use to name the log file other than the default, which will be created as hierRename.log in the current directory.

Input File

There will be three columns of data in each input line. These columns correspond to the dimension name, the old position name, and the new position name. The three fields will be tab-delimited. Any line not formatted this way will be ignored, and empty lines are also ignored.

Old position names must be an existing position name.

New position names cannot be an already used external name or an existing internal name. Lines having invalid position names will be ignored and added to the log file.

Old Position Name and New Position Name should not be prefixed with the name of the dimension.

The 'width' attribute in the domain must be greater than or equal to the max length of the new external names in the input file. Otherwise, width reconfiguration must be done before the rename process (using the **inithier** utility). If the width of the new name is greater than the width attribute of the corresponding dimension, RPAS will print an error in the log file and ignore the record.

Dimensions specified in the input file should belong to the hierarchy that is specified in arguments. Otherwise, the record will be ignored, and RPAS will print an error in the log file.

Setting Properties for Dimensions – dimensionMgr

The dimension manager utility is used for setting a number of parameters for dimensions and positions. These parameters are set when using the functionality of Position Name Indirection (PNI). This feature provides the ability to have position names that are longer than the default 24 characters and for a dimension to have dummy positions.

Usage

```
dimensionMgr –d pathToDomain –dim dimensionName [COMMAND]
```

The following table provides descriptions of the arguments used by the `dimensionMgr` statement.

Argument	Description
–d pathToDomain	Specifies the path to the domain.
–dim dimensionName	Specifies the name of the dimension to which the settings will apply.
–specs	Command that displays the properties of the specified dimension.
–width widthVal	Command that sets the width of position names for the specified dimension. The default width for positions of a given dimension is 24 characters. Widths can only be extended and cannot be decreased.
–bufPctMin minVal	Command that sets the minimum percent of unused positions.
–bufPctMax maxVal	Command that sets the maximum percent of unused positions.

Notes:
Multiple command arguments are allowed.

Buffer Minimum and Maximum size are specified as a percentage of the total size of the Dimension. For example, a Dimension with 200 real positions and a Buffer Mininum of 5 and Maximum of 20 could have between 10 and 40 extra buffer positions at any given time.

If set, the Buffer Maximum must be greater than the Buffer Minimum and less than 10000. To turn off buffering for a Dimension, set both Minimum and Maximum to zero.

11

Data Management

Loading Measure Data – loadmeasure

The **loadmeasure** utility is used to load data into the domain. You must specify the measure name(s) and the path to the domain that contains the measure(s).

In a Global Domain environment, **loadmeasure** is centralized and can only be called in the master domain. The utility will load one or more input files that can contain data from one or all of the local domains within the given Global Domain environment. The utility will split the input files and load them into the required domain, which is the local domain to which the position belongs or the master domain if the measure has a base intersection above the partition level. The split will only occur once in the case of multiple measures. Local domains will be checked for files even if there is no file in the global domain. The utility can be run in parallel in a global domain environment.

Usage

```
loadmeasure –d pathToDomain –measure measureName{,measureName,…} {–logdirectory
directoryName} {–applyloads} {–forcePurge} {–loglevel level}
```

The following table provides descriptions of the arguments used by the `load,easure` statement.

Argument	Description
–d pathToDomain	Specifies the domain in which to load the measure.
–measure measureName	Specifies the name of the measure to load. The name must be lowercase. If more than one measure is specified, all the measures must have the same input file.
–logdirectory directoryName	Specifies the location of the output error log. The default location is pathToDomain/scripts/err.
–applyloads	Use this argument to apply any staged loads for the named measure.
–forcePurge	Force the purge routine to run even if no new data is loaded.
–loglevel level	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.
–processes count	Specifies the maximum number of child loadmeasure processes to be running concurrently across the local domains in a Global Domain environment. If this argument is omitted or is less than two, the application will do all processing in a single process and no child processes will be created. This only specifies the number of child processes and the controlling process is not included, so you will actually have count + 1 processes.

Running Scripts

The **loadmeasure** utility also provides the ability to automatically run scripts before and after the utility is executed. These are referred to as "pre-processing" and "post-processing" scripts.

When **loadmeasure** is called, the utility checks for the existence of scripts named "pre<*measurename*>.sh" in the "../scripts" directory of the domain.

If scripts exist, they are run on the domain in which the utility has been called prior to the execution of the utility. Likewise, after the utility has completed running, it checks for the existence of scripts named "post<*measurename*>.sh" and executes them immediately. When loading multiple measures in a single call only the preprocessing script for the first listed measure will have any affect on the data.

Exporting Measure Data – exportData

Use exportData to export measure data from RPAS into text files. Each line that is exported contains the position name for the exported dimension followed by the value in the cell for each array being exported.

> **Note:** More than one array may be exported and more than one dimension in each array can be exported.

The utility may be invoked by specifying all parameters on the command line or by specifying an array that contains a list of the parameters.

When running this utility in a Global Domain environment, the utility should only be called to export data from the master domain. The utility will extract the data from either the local domains or the master domain depending on where the data resides, which depends on the level at which the Global Domain environment is partitioned.

The parameters specify what arrays and dimensions are exported and how to format the data. It is best to specify the arrays first. An array specification begins with `–array` followed by the array information. This includes the array name, formatting string, NA cell value, and NA cell value formatting string. The formatting string for both the cell value and NA value is based on the C language printf function formats. See the documentation on the printf function if you are not familiar with the possible values. The –array parameter can be repeated as needed to export more than one array into the same export file. Remember that the order the arrays appear in the –array parameter is the order that they will appear in the export file.

After the arrays have been specified, you need to specify the dimensions to be exported within the arrays. The `–dim` parameter is used to specify a dimension in an array. The `–dim` parameter is followed by the dimension name, a convert option, the formatting string (just like an array), and the order the dimension appears in the export file. Because arrays are not required to contain identical dimensions, it is important to list all dimensions in all arrays with the `–dim` parameter. This makes it possible to track dimensions across arrays and line the data up correctly. If a dimension in an array is not to be in the export file, set the last value of this parameter to 0. The conversion option specifies either the number of characters to be removed from the position name or it specifies an array that contains the real position name. If an array name is given, this array must be a vector. The function will go to this array and use the original position name to jump to the cell of the same position name. It will then get the cell value and use that as the position name in the export.

It is possible to specify the number of decimal places when exporting numeric measures of data type "real." This setting is defined in the specifications for measures, arrays, and dimensions (measSpec, arraySpec, and dimSpec). The format is %[*.precision*]*type* where [*.precision*] is the number of decimal places and *type* is the letter "f" (without quotes). For example, the setting *%.2f* would export numbers with two decimal places. Other settings are provided below.

If all parameters are contained in an array, after the export file name and source database name, the `--params` parameter is used to specify the database name and array name that contains all of the parameters needed for the export.

> **Note:** Either the `–array`, `–meas`, or `–params` parameters must be specified when using this utility.

Usage

```
exportData –d domainPath –out outputFile –params db array
exportData –d domainPath –out outputFile –meas \"measSpec\"
{–wb wbName} {options}
exportData –d domainPath –out outputFile –array \"arraySpec\" {options}
```

The following table provides descriptions of the arguments used by the `exportData` statements.

Argument	Description
–d domainPath	Specifies the domain that contains the data that you want to export.
–out outputFile	Specifies the file that will contain the exported data. The outputFile is relative to the domain unless the full path is specified.
–params db array	Instead of specifying all parameters on the command line, this parameter allows the parameters to be read from an array. db specifies the name of a Gem file where the array of parameters is stored. array specifies the name of an array in the specified database that has the above parameters.
–meas \"measSpec\"	Specifies the measure to export. measSpec must be quoted, and the format is \"measName cellFormat naValue naFormat\" The –meas argument can be repeated to export multiple measure arrays to the same output file. Measures are exported at the base intersection.

Argument	Description
–array \"arraySpec\"	Specifies the array to export. arraySpec must be quoted, and the format is \"dbName|arrayName cellFormat naValue naFormat\" dbName can be a path to the database (relative paths are relative to the domain root). Both cellFormat and naFormat use printf format commands. See the documentation on the printf function if you are not familiar with the possible values. The –array argument can be repeated to export multiple arrays to the same output file. The order in which arrays are listed is the order in which they will be exported. **Note:** This argument cannot be used in a Global Domain environment and can only be used in simple domains.

The following table provides descriptions of the options used by the `exportData` statements.

Option	Description
–dim \"dimSpec\"	Specifies the dimension to be exported. dimSpec must be quoted, and the format is \"dimName conversion format order\" conversion is either a count of the number of characters to strip from the start of the position name or the name of an array to be used to translate the position name before writing to the output file. format is a printf-style format for the position names. See the documentation on the printf function if you are not familiar with the possible values. order indicates the order the dimension is listed in the output file. If the value is 0, then the dimension is not exported. The –dim parameter can be repeated.
–skipNA always|allna|anyna|arrayna	Controls whether a line of data is exported based on having NAs in a cell. always exports data regardless of whether or not it contains NAs. allna does not export a row of data if all columns are NA (default). anyna does not export a row of data if any cell contains a NA value. arrayna does not export a row of data if the value in the given array name is NA (requires –naArray).

Option	Description
–naArray arrayName	When arrayna is specified using the –skipNA parameter, this option specifies the export array that is checked to determine if data is exported.
–index arrayName	Controls whether arrays are indexed by looking at a specified array. Only export the non-NA cells in the given array and each cell in the other arrays that have the same position names. If another array is at a higher dimension level, translate the given arrays cell index to the other arrays.
–range start:end	Used in conjunction with the –wide parameter, specifies a range along the innermost dimension. Only values in the range are considered for export.
–append	Specifies that output is appended at end of output file. The default is to overwrite output file.
–wide	This parameter causes the data to be exported wide, which means the innermost dimension will go across the row instead of each cell on a separate line. This is most useful when the innermost dimension is time. This format puts all time data on one row of output with breaks for each of the other exported dimensions. The –range parameter can be used in conjunction with wide format (–wide) to specify a range along the innermost dimension. Only values in the range will be considered for export. In future versions ranges could be expanded to include the column format.
–time	Specifies a YYYYMMDD format for dates.
–precision precisionValue	This parameter causes the utility to avoid exporting values that differ from the NA value by the specified value. Any values smaller than the precision value are not exported. For example consider a measure with the NA value of zero and a precision value of 0.01. A value of 0.0034 would not be exported while a value of 0.34 would be exported. The precision value must be less than one. If a value greater than one is provided the utility returns a warning.

Mapping Data between Domains – mapData

The **mapData** utility is used to move data from one domain to another. Specifically, it copies data from an existing domain, database, or array to a new domain, database or array.

Before running this utility, the new hierarchy must be loaded in the destination domain. After **mapData** has copied data, administrators can purge the source domain by calling **loadHier** with a purge age of 0. Tasks such as hierarchy loading, hierarchy purging, and the validation of source and destination domains are performed outside of this utility.

Usage

```
mapData -srcDomain srcDomainPath  -destDomain destDomainPath
        {-db dbName {-array arrayName}} {-loglevel}
```

The following table provides descriptions of the arguments used by the `mapData` statement.

Argument	Description
–srcDomain srcDomainPath	Specifies the full path of the source domain.
–destDomain destDomainPath	Specifies the full path of the destination domain.
–db dbName	Move data for the specified database, which must be a valid .gem file. If this argument is not specified the entire domain will be included in the operation.
–array arrayName	Move data the specified array within a given database. The database in which the array resides must be specified with the –db argument.

Moving Data between Arrays – updateArray

The updateArray utility moves data from a source array to a destination array. The destination array must contain the superset of dimensions in both source arrays. The source array's dimensions may be at the same or higher level as mapped by the dimension dictionary. If a dimension in the source array is at a higher level, the results are spread across the lower level dimension in the destination. If there are extra dimensions in the destination array, the results are replicated across these extra dimensions. The NA value of the destination array remains unchanged.

To limit the scope of the update, a mask array and an innermost range may be specified. If a mask array is given, the update is limited to cells in the source array for which the corresponding mask cell is on. If an innermost range is given for source or destination array, the update is limited to cells that are within the start and end of this range on the innermost dimension. If the source and destination arrays are not in the same domain, the measure store associated with the source domain is used to find hierarchy information.

Usage

```
updateArray –destArray dbPath.arrayName {–srcArray dbPath.arrayName}
{–destDomain domainPath {–srcDomain domainPath} {–maskDomain domainPath} {–
maskArray dbPath.arrayName} {–updateMethod method} {–srcRange first:last} {–
destRange first:last} {–srcScalar scalarCell} {–version} {–loglevel level}
updateArray –argFile filename {–version} {–loglevel level}
```

The following table provides descriptions of the arguments used by the `updateArray` statements.

Argument	Description
–destArray dbPath.arrayName	Required argument to specify the destination array where the data will be copied. dbPath is relative to destDomain.
–srcArray dbPath.arrayName	Optional argument. Default is no source array. **Note:** This parameter cannot be used with –srcScalar scalarCell.
–destDomain domainPath	Optional argument. Default is current working directory.
–srcDomain domainPath	Optional argument. Default is current working directory.
–maskDomain domainPath	Optional argument. Default is current working directory.
–updateMethod method	Optional argument. Default is OVERLAY. The following update methods are available: SKIPNA – omit NA cells in source. SKIPPOP – omit populated cells in source. OVERLAYNA – update NA cells in destination. OVERLAYPOP – update populated cells in destination. OVERLAY – update all cells in destination with source.
–srcRange first:last	Optional argument. Default is no range. Defines range along innermost dimension of source array.
–destRange first:last	Optional argument. Default is no range. Defines range along innermost dimension of destination array.

Argument	Description
–srcScalar scalarCell	Optional argument. Default is NA cell. Format for scalar cell is one of: NUMERIC: numeric value STRING: literal value BOOL: Boolean value DATE: date value NA. **Note:** This parameter cannot be used with –srcArray dbPath.arrayName.

12

Operational Utilities

Find Alerts – alertmgr

Alerts are an exception management tool for users. An alert is a measure that evaluates a business rule (returning a value of true or false). RPAS then notifies users of the "true" conditions and allows users to build workbooks to resolve the scenario that drove the alert.

Alert measures are first defined in the domain using the Configuration Tools. These measures are of type Boolean, which means they have a value of true or false. Next, rules (expressions) are registered in the domain for the alert measures to define the business rules used to evaluate the alert.

Once the registration process is complete, the alert utility is run to "find" the alerts in the domain. After the alert finder has been run, you can view the identified alerts in the Alert Manager dialog window in the RPAS client.

The following is a summary of the process for defining and finding an alert:

- Create an alert measure – This must be a Boolean measure (values are true-false, or yes-no) and must be defined in the RPAS Configuration Tools.
- Create the alert (the expression) for which the alert should be evaluated using the Configuration Tools. This flags the registered measure as an alert so that it is recognized when the "alert finder" is run.
- Run the "alert finder" on the domain to evaluate the number of instances when one or more alert expressions are true. This operation is completed using the RPAS utility **alertmgr**.

Usage

```
alertmgr –d pathtodomain –findAlerts {-alerts "a1 a2 …" | -categories "cat1 cat2 …"}
```

> **Note:** This utility includes arguments that are not documented in this guide as it is recommended that those operations are to be configured using the Configuration Tools to ensure consistency between the configuration and the domain.

The following table provides descriptions of the arguments used by the `alertmgr` statement.

Argument	Description
–d pathToDomain	Specifies the directory in which to run the utility.
–findAlerts	Finds alerts in the specified domain. The utility will find all alerts in the domain if neither the –alerts or –categories arguments are specified.
–alerts a1 a2 …	Evaluate specific alerts in the domain. a1 a2 … must be valid names of alerts that are defined in the domain.

Argument	Description
–categories cat1 cat2 …	Evaluate all alerts in the domain that are associated with specific categories of alerts. cat1 cat2 … must be valid names of alert categories that are defined in the domain.

Copying Domains – copyDomain

The **copyDomain** utility is used to copy a simple domain or all domains included in a global domain environment. Domains are often copied before upgrading the domains after receiving a patch to RPAS.

For a standard, simple domain (in other words, not a global domain environment), **copyDomain** copies the domain directory recursively from one location to another.

For a global domain environment **copyDomain** copies the master domain to the specified destination, and then it copies each local domain into corresponding subdirectories of the new location. As part of this particular replication process, the utility also updates all relevant data structures so that the domains are properly connected together.

Relative paths are supported with this utility and are used when creating the new copies of all the underlying data structures (arrays). Relative paths are based on the full pathname of the domain's root directory.

Usage

```
copyDomain –d  pathToSrcDomain –dest pathToDestDomain { -f }
copyDomain –version
```

The following table provides descriptions of the arguments used by the `copyDomain` statements.

Argument	Description
–d pathToSrcDomain	Specifies the path of the domain to be copied.
–dest pathToDestDomain	Specifies the path to where the domain is to be copied. The copied domain can also be renamed in this step by providing a name different than the source domain. This argument must be provided when using any other option of the utility. If this argument is not provided then the domain is updated to have relative paths.
–force	Deletes the existing domain at the specified destination path before copying the source domain.
–partitionPositions positions	Limits the copying of local domains to the specified positions at the partition level; positions is a comma-separated list of positions at the partition level.
–copyWorkbooks workbookList	Copies only the specified workbooks to the destination location; workbookList is either a comma-separated list of the workbooks to copy, or the value "none" such that no workbooks are copied. If this argument is not specified all workbooks in the environment will be copied.

Argument	Description
–skipInput	Do not copy the "input" directory located in the source domain.
–skipConfig	Do not copy the "config" directory located in the source domains.
–skipEmptyDir	Do not copy the empty directory located in the source domain.
–noSubDomains	Do not copy any local domains in the source domain.
–export	Export each Acumate database (.gem file) from the source domain into a format that can be used on a UNIX platform.
–gzip	Compress the copied domain into a gzip format.

Note: If the pathToSrcDomain argument is provided but the pathToDestDomain is not, then the utility will update the source domain environment to have relative paths to all the local domains. This is commonly used to update a global domain to have a recursive directory structure which is useful when copying a domain environment.

Setting Miscellaneous Domain Properties – domainprop

Use the domainprop utility to manipulate the properties of a domain. You can specify password properties, lock user accounts, and determine whether or not a daemon is currently managing a domain.

Usage

```
domainprop –d pathToDomain –expirePassword {days} {–passwordHistory
{oldPasswordCount}} {–property propertyname=value} {–lockAccount {failedLogins}}
{–daemonPort}
```

The following table provides descriptions of the arguments used by the `domainprop` statement.

Argument	Description
–d pathToDomain	Specifies the domain you wish to manipulate.
–expirePassword days	Used to set or view the number of days a password is valid. Specify a valid integer after the argument to set the number of days for which the password will be valid. If no number is provided the utility prints the current setting.
–passwordHistory oldPasswordCount	Used to set or view the number of previous passwords that are kept to ensure that a user does not repeat his password too often. If a number follows –passwordHistory, the property is set to that number. Otherwise, the current setting is printed.

Argument	Description
–property propertyname=value	Used to specify the property to be changed. See the list of properties below that can be set with this utility.
–lockAccount {failedLogins}	Used to set or view the number of failed login attempts that can occur before the account is locked out. If a number follows –lockAccount, the property is set to this value. Otherwise, the current setting is printed.
–daemonPort	Prints a message that indicates whether or not the domain is currently being managed by a domain daemon. If the domain is not currently being managed, the port of the last daemon to manage the domain is printed.

Available Properties

- disable_commit_later (value is Boolean) – Setting this property to "true" (or "t") disables the ability to use "Commit Later" in the File menu of the RPAS client. This property is set to false by default.
- help_path – This property is set when using custom help files for the RPAS client instead of the default help files that are provided. These files need to be in "Webhelp" format and this path would normally be a path to a network server where the common help files would be located.

Calculation Engine – mace

The mace utility (Multi-dimensional Array Calculation Engine) allows you to evaluate rule groups or expressions in order to manipulate measures. This allows the use of the RPAS 11 calculation engine in batch. The most commonly used commands for batch processes will be:

mace –d *domainPath* –run –group *groupName*

OR

mace –d *domainPath* –run –expression "*temporary expression*"

The first will cause an entire rule group to be evaluated, and the second will evaluate a single expression.

The mace utility can also be used to:

- create rules and rule groups
- add rules to rule groups
- add expressions to rules
- delete rules not contained in a rule group
- remove any or all rule groups
- validate expressions
- print a list of rules or rule groups.

Usage

```
mace –d domainPath –find string
mace –d domainPath –newGroup groupName
mace –d domainPath –newRule {-ruleName ruleName}
mace –d domainPath –addRule groupName:ruleName
mace –d domainPath –addExpression ruleName –expression exprString
mace –d domainPath –check –expression expString
mace –d domainPath –run (–group groupName | –expression expString) {–
debugRuleEngine}
mace –d domainPath –resolve groupName –measures measureList {–debugRuleEngine}
mace –d domainPath –transit workbookName –group groupList
{–debugRuleEngine}
mace –d domainPath –print (–rule ruleList | –group groupList | allGroups)
mace –d domainPath –removeGroup ruleGroup
mace –d domainPath –purgeRules
mace –d domainPath –removeAllRuleData
mace –d domainPath –validate (general | calc) –ruleGroup groupName
mace –d domainPath –validate refresh –ruleGroup groupName
–calcRuleGroup calcGroupName
```

The following table provides descriptions of the arguments used by the `mace` statements.

Argument	Description
–d domainPath	Specifies the domain in which to load the measure.
–find string	Use this argument to search all expressions for the specified string, printing all the rules and rule groups that have these expressions.
–newGroup groupName	Use this argument to create a new rule group with the specified name.

Argument	Description
–newRule {–ruleName ruleName	Use this argument to create a new empty rule. If desired use the –ruleName parameter to specify a name for the rule.
–addRule groupName:ruleName	Use this argument to add the specified rule to the specified rule group.
–addExpression ruleName	Use this argument to add an expression to the specified rule.
–expression exprString	Use the argument to specify the expression. This argument is used in conjunction with the –addExpression, –check, and –run arguments.
–check	Use this argument to validate the specified expression.
–run	Use this argument to evaluate the specified expression or rule group.
–group groupName	Use this argument to specify the rule group to evaluate using the –run argument.
–resolve groupName	Use this argument to order (does not evaluate) expressions within rule group. Requires a comma-separated list of edited measures.
–measures measureList	Use this argument to specify the measures to resolve.
–transit workbookName	Use this argument to run a calc engine by transitioning over a list of rule groups. Requires the name of an existing workbook and a comma-separated list of rule-group names.
–group groupList	Use this argument to specify a list of group names, separated by commas. Use this argument in conjunction with the –transit and –print arguments.
–print	Use this argument to print all the specified rules and rule groups. The ruleList is a comma-separated list of rule names. The groupList is a comma-separated list of group names. If "true" is supplied for either ruleList or groupList, all rules or rule groups are printed.
–rule ruleList	Use this argument to specify a list of rule names, separated by commas. Use this argument in conjunction with the –print argument.
–allGroups	Use this argument in conjunction with the –print argument to print all rule groups.

Argument	Description
–debugRuleEngine	Use this argument to generate a file "mace.log" in the working directory for logging RuleEngine specific debug information.
–addGroup	Use this argument to create a new rule group with the specified name
–removeGroup	Use this argument to remove specified group and non-shared rules in it.
–purgeRules	Use this argument to remove all rules not contained in any rule groups.
–removeAllRuleData	Use this argument to remove all rule groups and all rules.
–validate	Use this argument to validate rule groups. If you need to validate a calc rule group, use the calc param. If you need to validate a refresh rule group, use the refresh param along with the –calcRuleGroup param to specify the corresponding calc rule group. For all other types of rule group, use the general param.

Managing Users – usermgr

Use the usermgr utility to add a user, remove a user, or print information about a user in a specified domain.

Usage

```
usermgr –d domainPath –add userName –label label –password psw
–group grp {–admin} {–loglevel level}
usermgr –d domainPath –remove userName {–loglevel level}
usermgr –d domainPath –list {–loglevel level}
usermgr –d domainPath –print –user username {–loglevel level}
usermgr –d domainPath –print –group groupname {–loglevel level}
```

The following table provides descriptions of the arguments used by the `usermgr` statements.

Argument	Description
–d domainPath	Specifies the path to a domain that you want to add, remove, or get information about a user.
–add userName	Use this argument to add a user with a specified name. Use the other arguments specified in the usage to add those attributes for that user.
–label label	Use this argument to specify the label of the user that you are adding to the domain.
–password psw	Use this argument to specify the password of the user that you are adding to the domain.

Argument	Description
–group grp	Use this argument to specify the group of the user that you are adding to the domain.
–admin	Use this argument to specify that the user you are adding to the domain has administrative rights.
–remove userName	Use this argument to remove the user with the specified name from the domain.
–list	Use this argument to list all the users registered to the specified domain.
–print	Use this argument to print the specified user or group information.
–user username	Use this argument to specify the user name in the specified domain that you want to print. This argument is only applicable to –print option.
–group groupname	Use this argument to specify the group in the specified domain name that you want to print. This argument is only applicable to –print option.

Managing the Workbook Batch Queue – wbbatch

The **wbbatch** utility is used to manage workbooks in the workbook batch queue. The workbook batch queue is updated by using the standard RPAS wizard Auto-Workbook Build or using various options of the wbbatch utility.

The most common use of this utility is to build workbooks that have been scheduled to be automatically built using the "Auto-Workbook Build" wizard in the RPAS client.

When a user defers a workbook commit (using Commit Later), that workbook commit process is added to the Commit Later queue which is committed using this utility. An administrator can also add a workbook to the commit later queue with this utility.

RPAS provides the ability to update workbook data with domain data without having to rebuild the workbook; this refreshing process is completed using a workbook's default refresh rule group. Workbooks are added to the queue to be refreshed and refreshed using this utility.

The build and refresh operations can be executed in multiple, parallel processes using the –processes argument.

Usage

```
wbbatch –d pathToDomain –build {workbookName} {–loglevel level}
wbbatch –d pathToDomain –build –processes number
wbbatch –d pathToDomain –refresh {workbookName}{–loglevel level}
wbbatch –d pathToDomain –refresh –processes number
wbbatch –d pathToDomain –scheduleRefresh wbName {–loglevel level}
wbbatch –d pathToDomain –unscheduleRefresh wbName {–loglevel level}
wbbatch –d pathToDomain –commit {workbookName}{–loglevel level}
wbbatch –d pathToDomain –scheduleCommit wbName {–loglevel level}
wbbatch –d pathToDomain –unscheduleCommit wbName {–loglevel level}
```

The following table provides descriptions of the arguments used by the `wbbatch` statements.

Argument	Description
–d pathToDomain	Specifies the domain containing the workbooks.
–build {workbookName}	Builds workbooks scheduled to be built automatically. To build a single workbook in the auto-workbook build queue specify the name of the workbook. If no name is provided all workbooks in the queue will be built.
–build –processes number	Builds workbooks in the auto-workbook build queue in parallel using the specified number of parallel processes.
–commit {workbookName}	Commits workbooks with deferred commits. To commit a single workbook in the commit later queue specify the name of a workbook. If no name is provided all workbooks in the commit later queue will be committed.

Argument	Description
–refresh {workbookName}	Refreshes workbooks scheduled to be refreshed using this utility. To refresh a single workbook in the queue specify the name of the workbook. If no name is provided all workbooks scheduled to be refreshed will be completed.
–refresh –processes number	Refreshes workbooks in the auto-workbook refresh queue in parallel using the specified number of parallel processes.
–scheduleRefresh	Schedules a workbook to be refreshed later by adding it to the workbook batch queue.
–unscheduleRefresh	Removes a workbook from the workbook batch queue.
–scheduleCommit	Schedules a workbook to be committed later by adding it to the workbook batch queue.
–unscheduleCommit	Removes a workbook from the workbook batch queue.

Workbook Manager – wbmgr

Use the Workbook Manager utility to inspect or remove the existing workbooks. It is recommended that administrators use this utility to remove workbooks rather than doing so manually.

Usage

```
wbmgr –d pathToDomain –list –all {–loglevel level}
wbmgr –d pathToDomain –list –user userName {–loglevel level}
wbmgr –d pathToDomain –print  –wbList wb1,wb2,... {–loglevel level}
wbmgr –d pathToDomain –remove –all {–loglevel level}
wbmgr –d pathToDomain –remove –user userName {–loglevel level}
wbmgr –d pathToDomain –remove –user userName –wbList wb1,wb2,...
{–loglevel level}
```

The following table provides descriptions of the arguments used by the `wbmgr` statements.

Argument	Description
–d pathToDomain	Specifies the domain that contains the workbooks.
–list –all	Lists all workbooks in the domain.
–list –user userName	Lists all workbooks belonging to the user.
–print –wbList wb1,wb2,...	Prints detailed information about workbooks in the list.
–remove –all	Removes all workbooks from the domain.
–remove –user userName	Removes all workbooks from the domain belonging to the specified user.
–remove –user userName –wbList wb1,wb2	Removes all the workbooks in the specified list for the specified user.

13

Informational Utilities

There are numerous RPAS utilities that can be used for finding information about many of the different components of a domain or domain data. The following utilities are solely for retrieving information and to not make any changes to a domain or data in a domain.

Retrieving Domain Information – domaininfo

The **domaininfo** utility is used to provide miscellaneous details about a domain, such as the type of domain (simple, master, or sub/local), the sparsity of the databases in the domain, and the upgrade/version history of the domain.

The domain path (**-d**) is required for all commands except **–expectedversion**.

Usage

```
domaininfo –d pathToDomain [Command]
domaininfo –expectedversion
```

The following table provides descriptions of the arguments used by the `domaininfo` statements.

Argument	Description
–domainversion	Display the RPAS version of the specified domain.
–type	Command to display the type of the domain. Possible values are Simple, Master, and Sub. A Simple domain is a traditional, non-partitioned (non-global) domain. A Master domain is the central/master domain of a global domain environment. A Sub domain is one local domain in a global domain environment that can contain one or more partitions.
–xnames	
–sparsity	Reports whether a domain has been designated to use spare or hypersparse measure databases.
–listsubdomains	Displays a list of all the local domains in a global domain environment, and indicate which positions at the partition level are in each local domain. This argument is only valid when run on a global domain.
–subdomain dim,pos	Indicates to which local domain the specified position belongs. The position can be at or below the partition level.
–history	Displays the version history of the domain, specifically when the domain was upgraded to new versions of RPAS (patches or releases).
–all	Displays all of the above information about the domain.
–expectedversion	Displays the expected RPAS version of the domain that the utility expects to find.

Checking the Validity of a Domain – checkDomain

This utility is used to check the validity of an existing domain. Its primary purpose is to verify that a master domain matches its respective local domains and report all discrepancies to the administrator.

Usage

```
checkDomain –d pathToDomain –type expectedType {-q}
```

The following table provides descriptions of the arguments used by the `checkDomain` statement.

Argument	Description
–d pathToDomain	Path to the domain that needs to be validated.
–type expectedType	Expected type of domain: simple, master, or sub.
–q	Quiet mode – Do not display progress messages.

When **checkDomain** is run on a **simple** domain the following two items get validated:

- The domain directory exists
- It is of type "simple"

If **checkDomain** is run on a **Global Domain**, it verifies the following:

- The global domain exists
- The global domain is of type "master"
- The global domain checks all of the sub-domains for:
  - The sub-domain directory exists and is of type "sub"
  - If the master domain and the sub-domain have a repos directory
  - The measures, rules, rule groups, templates, and functions are the same in the global and sub-domain

If it is run on a **sub-domain**, it checks all of the items listed above for the global domain, but the validation is only performed between the global domain and the specified sub-domain.

Determining RPAS Server Version – rpasversion

Use the **rpasversion** utility to determine which version of the RPAS server is running in a particular location.

Usage

```
rpasversion –l pathToLibrary
```

List Contents of a Database – listDb

Use the listDb utility to list the contents of a database.

Usage

```
listDb –db dbName {–loglevel level}
```

The following table provides descriptions of the arguments used by the `listDb` statement.

Argument	Description
–db dbName	Specifies the database for which you want to list the contents.

Printing Data from Arrays – printArray

Use **printArray** to print the contents of an array.

Usage

```
printArray –array db.array –specs {–maxpos num}
printArray –array db.array {–cell "dim1:pos1,dim2:pos2,..."
{–format "formatString"}
printArray –array db.array –slice "dim1:pos1,dim2:pos2,..."
{–format "formatString"} {–cellsperrow num} {–noposnames}
printArray –array db.array –allpopulatedcells {–format "formatString"}
{–cellsperrow num} {–noposnames}
```

The following table provides descriptions of the arguments used by the `printArray` statement.

Argument	Description
–array db.array	Specifies the array you want to print. You must specify the full path to the database containing the array. Do not specify the .gem suffix.
–specs	Prints the specifications of the array and positions along each dimension.
–cell	Identifies a single cell. Must not contain spaces. Specify using the format "dim1:pos1,dim2:pos2,…"
–slice	Identifies a single 1-D slice. Must not contain spaces. Specify using the format "dim1:pos1,dim2:pos2,…"
–allpopulatedcells	Print all populated cells including the navalue of the array.

Argument	Description
–format	If –format is specified, any cells with numeric values are interpreted as dates. formatString determines how dates are interpreted, and can include: %Y - 4 digit year %m - month number (01 to 12) %d - numeric day of month (01 to 31) %H - 24 hour clock (00 to 23) %M - minute (00 to 59) %S - seconds (00 to 61) %s – milliseconds
–maxpos	Used in conjunction with –specs, prints only the first n positions for each dimension.
–cellsperrow	Specifies how many cells should be printed in each row.
–noposnames	Suppresses the output of position names.

Printing Data from Measures – printMeasure

Use the printMeasure utility to print measure information.

Usage

```
printMeasure –d pathToDomain –m measure {–wb workbookName}
{–specs} {–listDataIntersections} {–printData aggType.intersection}
{–loglevel level}
printMeasure –list
```

The following table provides descriptions of the arguments used by the `printMeasure` statements.

Argument	Description
–d pathToDomain	Specifies the domain that contains the measure you want to print.
–m measure	Specifies the measure you want to print.
–wb workbookName	Specifies the workbook associated with the measure you want to print. If a workbook is not specified, the domain measure information is printed.
–specs	
–listDataIntersections	
–list	It does not require –d domainPath.
–printData aggType.intersection	Prints out the nobs, nods and sparse format of the measure array at the specified intersection and agg type.

A

Appendix: Integration Guide

RDF and Merchandise Financial Planning Integration with RMS and Price

Summary of Integration Approach with RMS

The strategy for the extraction of foundation data from RMS is for the extract programs (RMSE) to provide flat files in a generic format. For each solution that will use this data, transformation scripts are used to reformat the data as needed to produce a file suitable for loading into the application. For the instances of data coming from RPAS to non-RPAS applications, extract programs are specific to the application in need of the data. Other scripting languages are then used (Perl or AWK) to perform additional data formatting.

This appendix summarizes the following:

- RMS 11.0/12.0 to RDF 12.0 and Merchandise Financial Planning 12.0 transformation programs
  - Merchandise Hierarchy
  - Organization Hierarchy
  - Calendar Hierarchy
  - Store Close Dates
  - Store Open Dates
  - Daily Sales and Issues
  - Weekly Sales and Issues
  - Out of Stock Indicator
- RDF 12.0 to RMS 11.0/12.0 extract programs
  - Approved Forecasts and Standard Deviations (Cumulative Intervals)
- Grade (RPAS 12.0) to RMS 11.0/12.0 extract programs
  - Store Grades
- Curve (RPAS 12.0) to RMS 11.0/12.0 extract programs
  - Differentiator Profiles
- Merchandise Financial Planning 12.0 to RMS 11.0/12.0 extract programs
  - Receipt Plan
- Merchandise Financial Planning 12.0 to Price 4.5 extract programs
  - Markdown Budget
- Merchandise Financial Planning 12.0 to RDW 12.0 extract programs
  - Current Plan
  - Original Plan

Specifics on the usage of RMS extract programs (RMSE's) within the RDF transformation programs are beyond the scope of this document. See the RMS Operations Guide for more information on the RMS extract programs.

Environment Variable Setup

In addition to any variables identified in the RMS integration documentation, the transformation and/or extract programs require the following four environment variables:

- $RPAS_INTEGRATION_HOME: Identifies the location of the integration scripts when /common/header.ksh is run. This variable is used for all integration scripts packaged with the ARPOPlatform EXCEPT those included in 'rfx' (see $RDF_HOME below).
- $TO_RPAS: The staging area for the data to be loaded into RPAS. This directory should be located at the same level as the root of the RPAS domain. For example, if the domain RDF is located in Domains directory (example: /Domains/RDF), then $TO_RPAS should located at the same level as RDF (example: /Domains/to_rpas).
- $FROM_RPAS: The staging area for the data extract out of RPAS. This directory should be located at the same level as the root of the RPAS domain. For example, if the domain RDF is located in Domains directory (example: /Domains/RDF), then $FROM_RPAS should located at the same level as RDF (example: /Domains/from_rpas).
- $RDF_HOME: Identifies the location of the root of the 'rfx' directory. The 'rfx' directory packaged with the ARPOPlatform should be added to the location 'rfx' directory packaged with the RMS RETL programs.
- $RI_RMSVERSION: Identifies the version of RMS. This environment variable is only used by the extract scripts. If this variable is not set, the extract scripts assume an RMS version of 12. Set the value of this environment variable to 11 or 12, depending on your RMS version.

RDF and Merchandise Financial Planning Transformation Programs

Common Program for All Transformations

The **rdft.ksh** script runs all of the necessary data extraction and transformation scripts (**rmse_*.ksh** and **rdft_*.ksh**, respectively) that are needed to produce the files to be loaded into RPAS/RDF/Planning. Most of these scripts are run in parallel (as background jobs).

Usage

```
rdft.ksh  [-x]  [-c]  [-d dir]
```

Arguments:

- `-x`: This option will cause the execution of the RMS data extraction wrapper (rmse.ksh) to be skipped.
- `-c`: This option will cause FILE_DATE in rmse_config.env to be set to the current date instead of using VDATE.
- `-d`: This option will cause all programs executed by rdft.ksh to be obtained from the "dir" directory.

Transformations of Merchandise Hierarchy Data

rdft_merchhier.ksh is the primary script used to build the data for RPAS from the RMS Merchandise Hierarchy tables. The schema used to produce the output file depends on the attributes and differentiator settings in RMS:

- Case 1: If PROD_ATTRIBUTES_ACTIVE = False and DIFFS_ACTIVE = False, then rdft_merchhier.base.schema will be used to produce the file. In this case attributes and diff fields will not be included in the merchandise hierarchy file.
- Case 2: If PROD_ATTRIBUTES_ACTIVE = True and DIFFS_ACTIVE = False, then rdft_merchhier.attributes.schema will be used to produce the file. This schema must be manually edited to support a specific attribute model and must be kept in sync with rmse_attributes.schema and rmse_attributes.ksh (see the RMSE end user documentation).
- Case 3: If PROD_ATTRIBUTES_ACTIVE = False and DIFFS_ACTIVE = True, then rdft_merchhier.schema will be used to produce the file. In this case diff fields will be included in the merchandise hierarchy file.
- Case 4: If PROD_ATTRIBUTES_ACTIVE = True and DIFFS_ACTIVE = True, then an error will result. In this release, the combination of diffs and attributes is not supported.

Intermediate schema and scripts which may be used (depending on configuration options) to produce the merchandise hierarchy file:

- **rdft_diff.domain.schema**
- **rdft_merchdiff.domain.schema**
- **rdft_merchhier_diff_trans.ksh**
- **rdft_merchhier_split_by_domain.ksh**
- **rdft_clean_partition.ksh**

Additional merchandise hierarchy support for issue domains is provided in **rdft_item_loc.ksh**. This script is designed to produce a full item list for issues domains, only containing items that exist in the warehouses.

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Location Hierarchy Data

rdft_orghier.ksh is the primary script used to build the location data file needed for RPAS from the RMS Organizational Hierarchy Table.

The following five constants may be modified in the script based on location hierarchy data requirements:

- **COMPANY_NAME –** The label for the company position to be populated in the file.
- **COMPANY_ID –** The name for the company position to be populated in the file.
- **STORE_CLASS_CONCAT –** When set to "True", causes the STORE_CLASS to be concatenated on the left of the STORE_CLASS_DESCRIPTION field in the final Store data output file.
- **ADD_AT_SIGN_TO_WH_DESC –** When set to "True", will cause the WHSE_NAME field in the Warehouse ouput file to have an "@" prefix.

- **LONG_WAREHOUSE_RECORDS –** When set to "True", the Warehouse output records will consist of 16 fields. If it is "False", the records will contain only four fields, WH, WHSE_NAME, COMPANY and CO_NAME.

Intermediate schemas which may be used (depending on configuration options) to produce the location hierarchy file:

- rdft_issues.schema
- rdft_issues_long.schema
- rdft_orghier_store.schema

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Calendar Hierarchy Data

rdft_calhier.ksh transforms the Calendar Hierarchy data extracted from RMS for loading into RPAS.

Configuration inputs to the script include:

- **DATE_PREF –** The path to the file that contains text indicating whether the format of the Date Description field will be mm/dd/yyyy or dd/mm/yyyy. See the RMS Operations Guide for date format options.
- **LAST_DOW –** The path to the file that contains a day of week name or abbreviation indicating which day of the week is considered to be the end of the week for the fiscal calendar being used at this installation.

Transformations of Daily Sales and Issues Data

rdft_daily_sales.ksh produces the daily sales and issues data files based on regular, promotion, clearance, and issues.

The following constant may be modified in the script based on data requirements:

- **DOM_START_COL –** Defines the starting column position of the Domain ID in the RETL output schema. This is needed by **rdft_merchhier_split_by_domain.ksh** to split the files by domain ID. If the OUTPUT_SCHEMA file is modified, the value of DOM_START_COL may also require modification from the default value.

Intermediate schemas which may be used (depending on configuration options) to produce the sales and/or issues data file:

- rdft_daily_sales.schema

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Weekly Sales and Issues Data

rdft_weekly_sales.ksh produces the weekly sales and issues data files based on regular, promotion, clearance and issues.

The following constant may be modified in the script based on data requirements

- **DOM_START_COL –** Defines the starting column position of the Domain ID in the RETL output schema. This is needed by rdft_merchhier_split_by_domain.ksh to split the files by domain ID. If the OUTPUT_SCHEMA file is modified, the value of DOM_START_COL may also require modification from the default value.

Intermediate schemas which may be used (depending on configuration options) to produce the sales and/or issues data files:

- rdft_weekly_sales.schema

> **Note:** Issues-specific data transformation functionality is triggered based on the issues setting in RMS (ISSUES_ACTIVE must be set to True).

Transformations of Store Open Date Data

rdft_open_date.ksh produces the Store/Warehouse Opening Date data file.

Intermediate schema used to produce the store open date data files:

- rdft_open_date.schema

Transformations of Store Close Date Data

rdft_close_date.ksh produces the Store/Warehouse Closing Date data file.

Intermediate schema used to produce the store closing date data files:

- rdft_close_date.schema

Transformations of Out of Stock Indicator Data

rdft_outofstock.ksh produces the Store and Warehouse (issues) Out of Stock Indicator data extracted from RMS.

Intermediate schema and scripts which may be used (depending on configuration options) to produce the Out of Stock Indicator data file:

- rdft_outstock_split_by_domain.awk
- rdft_outofstock.schema
- rdft_outofstock_issues.schema
- rdft_outofstock_sales.schema

RDF and Merchandise Financial Planning Transformation Matrix

The following matrix identifies the transformation scripts and schemas used for each the hierarchy and data files produced for RDF 12.0 and Merchandise Financial Planning 12.0:

Directory	Script or Schema Name	Merchandise Hierarchy	Location Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
rfx/lib	rdft_merchhier_diff_trans.ksh	X							
	rdft_merchhier_split_by_domain.ksh	X							
	rdft_outofstock_split_by_domain.ksh						X		
rfx/schema	rdft_close_date.schema								X
	rdft_daily_sales.schema				X				
	rdft_diff.domain.schema	X							
	rdft_merchierdiff.domain.schema	X							
	rdft_merchier.attributes.schema	X							
	rdft_merchhier.base.schema	X							
	rdft_merchhier.domain.schema	X							
	rdft_merchhier.schema	X							
	rdft_open_date.schema							X	
	rdft_orghier_issues.schema		X						
	rdft_orghier_issues_long.schema		X						
	rdft_orghier_strore.schema		X						
	rdft_outofstock.schema						X		
	rdft_outofstock_issues.schema						X		
	rdft_outofstock_sales.schema						X		
	rdft_weekly_sales.schema					X			
rfx/src	rdft_ksh	X	X	X	X	X	X	X	X
	rdft_calhier.ksh			X					
	rdft_clean_partition.ksh	X							
	rdft_close_date.ksh								X
	rdft_daily_sales.ksh				X				

Directory	Script or Schema Name	Merchandise Hierarchy	Location Hierarchy	Calendar	Daily Sales & Issues	Weekly Sales & Issues	Out of Stock Indicator	Store Open Dates	Store Close Dates
	rdft_item_loc.ksh	X							
	rdft_merchhier.ksh	X							
	rdft_open_date.ksh							X	
	rdft_orghier.ksh		X						
	rdft_outofstock.ksh						X		
	rdft_weekly_sales.ksh					X			

RDF and Merchandise Financial Planning Extract Programs

Common Programs for Extracts

config.ksh is a configuration directory that requires both the RMS version being integrated and the backup action to be defined.

The following OPTIONAL arguments are available:

- Name of the domain: Defaults to directory name
- Number of the domain: Defaults to the 2 last digits of the directory name
- Format of timestamp attached to logs and processed input files: Defaults to: (date +"%b%d%a%I%M%p") (example: Aug02Thu0111PM)
- Data Drop: Defaults to ../../to_rpas
- Data Export: Defaults to ../../from_rpas
- Log Drop: Defaults to ./logs
- Error Drop: Defaults to ./err
- Reclass Data: Defaults to ../reclass_data

functions.ksh

This script file contains ksh functions that are used by scripts in [DOM]/scripts. It should be sourced, not executed in order to preserve environment variables.

header.ksh

This script should be run at the beginning of any implementation-specific script to setup function libraries, environment, and platform-specific routines.

Extract of Forecast Data for RMS

rdf_e_rms.ksh extracts forecast demand value and standard deviation (cumulative interval) at both day and week aggregations from an RDF domain.

Arguments:

- `-t`: <Domain Type> (S for sales, I for issues)
- `-w`: <Data Width> ([7...18], defaults to 12)
- `-d`: <Domain> (defaults to current directory)
- `-n`: <Domain Number> (defaults to last two digits of domain)

Output files:

${RPAS_EXPORT}/d<s|i>demand.<Domain Number> (demand at day)

${RPAS_EXPORT}/w<s|i>demand.<Domain Number> (demand at week)

The following table provides information about the output file data format.

Field	Start	Width	Format
Day | EOW Day	1	8	Alpha
Product ID	1	25	Alpha
Location ID	26	20	Alpha
Demand	46	12	Alpha
Std. Dev. Demand	68*	12*	Numeric (floating point, 4 decimal digits with decimal)

* Width of Demand and Std. Dev. Demand may be overridden with the -w parameter; stated values Demand width and Std. Dev. Demand start and width are based on default width of 12.

Note: the following must be defined in the shell environment prior to calling this script:

RPAS_HOME

RPAS_INTEGRATION_HOME

Load of Extracted Forecast Data and Standard Deviations to RMS

rmsl_forecast.ksh pulls the daily/weekly forecast items into RMS.

During the loading of each domain file the following steps are performed:

1. Truncate the partition in the RMS forecast table which corresponds to the domain ID.

   Note: Partition names should always be in the format: [tablename]_[domainID]

2. Append a domain field and insert the domain_id into each record.
3. Load the forecast data into the RMS forecast table.

Usage: rmsl_rpas_forecast.ksh daily | weekly

Intermediate schemas which may be used (depending on configuration options) to produce the forecast data files:

- rmsl_forecast_daily.schema
- rmsl_forecast_weekly.schema

Extract of Diff Profile Data for RMS

profile_e_alloc.ksh extracts Curve diff profiles for use by Allocation.

Arguments:

- `-p`: <Profile Number>
- `-m`: <Mask Measure> (Optional mask; only positions for which the mask value is non-NA will be exported.)
- `-w`: <Data Width> ([7...18], defaults to 12)
- `-d`: <Domain> (defaults to current directory)
- `-n`: <Domain Number> (defaults to last two digits of domain)

Output file: ${RPAS_EXPORT}/d1<Product Level>.<Domain Number>, where Product Level is the Aggregation intersection's Prod dimension

The following table provides information about the output file data format.

Field	Start	Width	Format
Product ID	1	25	Alpha
Location ID	26	20	Alpha
Diff ID (optional)	46	36	Alpha
Quantity	82	12*	Numeric (floating point, 4 decimal digits, no decimal)*
Std. Dev. Demand	68*	12*	Numeric (floating point, 4 decimal digits with decimal)

* Quantity width may be overridden with the -w parameter.

Note: The following must be defined in the shell environment prior to calling this script:

RPAS_HOME

RPAS_INTEGRATION_HOME

Extract of Store Grade Data for RMS

grade_e_rms.ksh extracts store grades for use by RMS.

Arguments:

- `-t` <Timestamp> (YYMMDDTTTT). This value corresponds to the timestamp of the Cluster Membership measure (clpm+<Timestamp>) to be extracted
- `-d` <Domain> (defaults to current directory)
- `-n` <Domain Number> (defaults to last two digits of domain)

Output file: ${RPAS_EXPORT}/gr<Timestamp>.<Domain Number>

Output file data format:

- (a thru c) constitutes the header records
- (d thru j) constitutes the detail records
- (k thu l) constitutes the footer records
  a. 'FHEAD'
  b. Line ID Number
  c. 'GRADU'
  d. 'FDETL' (Record Identifier)
  e. Line Sequence Identifier
  f. Grade Group ID Number: This value corresponds to the first 8 characters of the Cluster Run Name measure (clnam+<user-defined name>) set by the user in the Generate Cluster wizard in Grade. For integration with RMS, the Cluster Run Name must be populated with only numeric characters.
  g. Grade Group: This value corresponds to the first "**N"** characters of the Cluster Run Name measure (clnam+<user-defined name>) set by the user in the Generate Cluster wizard in Grade. "**N"** is 20 for RMS version 11.0, and "**N"** is 120 for RMS version 12. The script determines the RMS version from the environment variable RI_RMSVERSION. RMS version 12 is assigned by default.
  h. Store ID
  i. Grade Member Name
  j. 'FTAIL' (Record Identifier)
  k. Line ID Number
  l. FDETL Line Total Number

Extract of Receipt Plan for RMS

plan_e_alloc.ksh extracts Merchandise Financial Planning measures for use by RMS

Arguments:

- `-m`: <Measure Name>
- `-f`: <Dif Dimension>
- `-w`: <Data Width>] ([7...18], defaults to 12)
- `-d`: <Domain> (defaults to current directory)
- `-n`: <Domain Number> (defaults to last two digits of domain)

Output file: ${RPAS_EXPORT}/p1<Prod Dimension>.<Domain Number>

The following table provides information about the output file data format.

Field	Start	Width	Format
Product ID	1	25	Alpha
Location ID	26	20	Alpha
Diff ID	46	36	Alpha (For future use; blank now)
EOW Date	82	8	Alpha
Quantity	90	12*	Numeric (floating point, 4 decimal digits, no decimal)*

* Quantity width may be overridden with the -w parameter.

Note: The following must be defined in the shell environment prior to calling this script:

RPAS_HOME

RPAS_INTEGRATION_HOME

Extract of Markdown Budget Data for Price

plan_e_price.ksh extracts Markdown Planning measures for use by Price.

Arguments:

- `-a`: <measure name for Markdown Budget>
- `-b`: <measure name for Planned GM Dollars>
- `-c`: <measure name for Planned GM %>
- `-e`: <month format> (The month position format in the domain, must be one of the following:
  - "PYYYYMM", "PMMYYYY", "PMM_YYYY", "PYYYY_MM", "PYYYYMMM", "MMMYYYY","PYYYY_MMM", "MMM_YYYY"
  - P represents a prefix letter required by RPAS for certain calendar formats. MM represents 01, 02, 03, ...12
  - MMM represents JAN, FEB, MAR,..DEC. YYYY represents 2005, 2006, ...
- `-d`: <Domain> (The domain path has to be a master domain or a simple domain. A subdomain path will fail)
- `-o`: <outputfile name> (including file path)
- `-m`: <Optional: space separated list of additional measures to include> ("measure_x measure_y measure_z")

Example:

```
sh plan_e_price.ksh -a mdbudget -b plgmbudget -c plgmperc -e MMM_YYYY -d . -m rsal psal csal -o ASH_BUDGET_TBL.dat
```

Data Assumptions:

1. All measures to be extracted include month in their base intersection
2. All measures to be extracted are configured at the same base intersection

The extract produces the following pipe (|) delimited data fields, each row having the format described in the following table.

Field	Max Width	Format
Product ID	25	Alpha
Location ID	25	Alpha
Year ID	4	Alpha
Month ID	2	Alpha
MARKDOWN_BUDGET	22	Numeric (floating point, 3 decimal digits)
PLANNED_GM_DOLLARS	22	Numeric (floating point, 3 decimal digits)
PLANNED_GM_PERC	22	Numeric (floating point, 3 decimal digits)
Measure Name1	22	Numeric (floating point, 3 decimal digits)
Measure Name2	22	Numeric (floating point, 3 decimal digits)

Field	Max Width	Format
Measure Name3	22	Numeric (floating point, 3 decimal digits)
Measure Name4	22	Numeric (floating point, 3 decimal digits)
Measure Name5	22	Numeric (floating point, 3 decimal digits)
Measure Name6	22	Numeric (floating point, 3 decimal digits)
Measure Name7	22	Numeric (floating point, 3 decimal digits)
Measure Name8	22	Numeric (floating point, 3 decimal digits)
Measure Name9	22	Numeric (floating point, 3 decimal digits)
Measure Name10	22	Numeric (floating point, 3 decimal digits)

Note: The following must be defined in the shell environment prior to calling this script:

RPAS_HOME

RPAS_INTEGRATION_HOME

Extract of Current Plan Data for RDW

plan_e_plcblwdm.ksh extracts Current Plan data from Merchandise Financial Planning for use by RDW.

The script expects an inputfile as an argument. The inputfile specifies the hierarchy labels and measure names that need to be extracted by the script. The user is expected to create this input file and this is used by the script to extract the appropriate measures.

Arguments

- `-d` <Domain> (The domain path has to be a masterdomain or a simple domain. A subdomain path will fail)
- `-o` <outputfile> (Including file path)
- `-i` <inputfile with field to measure mapping> (Including file path)

  Input file should contain all the fields below followed by a space and the associated measure name.

  Example of input file data format:

```
DEPT_IDNT DEPARTMENT_LABEL
CLASS_IDNT CLASS_LABEL
SBCLASS_IDNT SBCLASS_LABEL
LOC_IDNT LOCATION_LABEL
F_PLN_CURR_CLRC_SLS_QTY <MEASURE_NAME>
F_PLN_CURR_PRMTN_SLS_QTY <MEASURE_NAME>
F_PLN_CURR_RGLR_SLS_QTY <MEASURE_NAME>
F_PLN_CURR_CLRC_SLS_AMT <MEASURE_NAME>
```

F_PLN_CURR_PRMTN_SLS_AMT <MEASURE_NAME>
F_PLN_CURR_RGLR_SLS_AMT <MEASURE_NAME>
F_PLN_CURR_GRS_PRFT_AMT <MEASURE_NAME>
F_PLN_CURR_RGLR_MKDN_AMT <MEASURE_NAME>
F_PLN_CURR_CLRC_MKDN_AMT <MEASURE_NAME>
F_PLN_CURR_PRMTN_MKDN_AMT <MEASURE_NAME>
F_PLN_CURR_SHRK_QTY <MEASURE_NAME>
F_PLN_CURR_SHRK_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_BOP_QTY <MEASURE_NAME>
F_PLN_CURR_BOP_COST_AMT <MEASURE_NAME>
F_PLN_CURR_BOP_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_OTB_QTY <MEASURE_NAME>
F_PLN_CURR_OTB_COST_AMT <MEASURE_NAME>
F_PLN_CURR_OTB_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_RCPTS_QTY <MEASURE_NAME>
F_PLN_CURR_RCPTS_COST_AMT <MEASURE_NAME>
F_PLN_CURR_RCPTS_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_CMTS_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_ORD_CNCLLD_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_ORD_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_RECL_IN_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_RECL_OUT_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_RTV_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_CMTS_QTY <MEASURE_NAME>
F_PLN_CURR_ORD_CNCLLD_QTY <MEASURE_NAME>
F_PLN_CURR_ORD_QTY <MEASURE_NAME>
F_PLN_CURR_RECL_IN_QTY <MEASURE_NAME>
F_PLN_CURR_RECL_OUT_QTY <MEASURE_NAME>
F_PLN_CURR_RTV_QTY <MEASURE_NAME>
F_PLN_CURR_EOP_RTL_AMT <MEASURE_NAME>
F_PLN_CURR_WOS_AMT <MEASURE_NAME>
F_PLN_CURR_EOP_COST_AMT <MEASURE_NAME>
F_PLN_CURR_ORD_CNCLLD_COST_AMT <MEASURE_NAME>
F_PLN_CURR_ORD_COST_AMT <MEASURE_NAME>
F_PLN_CURR_CMTS_COST_AMT <MEASURE_NAME>
F_PLN_CURR_CUM_MKUP_PCT <MEASURE_NAME>
F_PLN_CURR_EOP_QTY <MEASURE_NAME>
F_PLN_CURR_WOS_QTY <MEASURE_NAME>
F_PLN_CURR_COGS_AMT <MEASURE_NAME>
F_PLN_CURR_EXCL_SLS_VAT_AMT <MEASURE_NAME>
F_PLN_CURR_EMPLY_DISC_AMT <MEASURE_NAME>
F_PLN_CURR_FRGHT_COST_AMT <MEASURE_NAME>
F_PLN_CURR_WRKRM_COST_AMT <MEASURE_NAME>
F_PLN_CURR_RTRNS_SLS_AMT <MEASURE_NAME>

The following table provides information about the output file data format.

Field	Start	Width	Format
DAY_DT EOW Day	0	8	Date
DEPT_IDNT	6	4	String
CLASS_IDNT	10	4	String
SBCLASS_IDNT	14	4	String
LOC_IDNT LOCATION_LABEL	18	10	String
MEASURE	28		Float

Note: The following must be defined in the shell environment prior to calling this script:

RPAS_HOME

RPAS_INTEGRATION_HOME

Extract of Original Plan Data for RDW

plan_e_ploblwdm.ksh extracts Original Plan data from Merchandise Financial Planning for use by RDW.

The script expects an inputfile as an argument. The inputfile specifies the hierarchy labels and measure names that need to be extracted by the script. The user is expected to create this input file and this is used by the script to extract the appropriate measures.

Arguments:

- `-d` <Domain> (The domain path has to be a masterdomain or a simple domain. A subdomain path will fail)
- `-o` <outputfile> (Including file path)
- `-i` <inputfile with field to measure mapping> (Including file path)

  Input file should contain all the fields below followed by a space and the associated measure name.

  Example of the input file data format.

```
DEPT_IDNT DEPARTMENT_LABEL
CLASS_IDNT CLASS_LABEL
SBCLASS_IDNT SBCLASS_LABEL
LOC_IDNT LOCATION_LABEL
F_PLN_ORIG_CLRC_SLS_QTY MEASURE_NAME
F_PLN_ORIG_PRMTN_SLS_QTY MEASURE_NAME
F_PLN_ORIG_RGLR_SLS_QTY MEASURE_NAME
F_PLN_ORIG_CLRC_SLS_AMT MEASURE_NAME
F_PLN_ORIG_PRMTN_SLS_AMT MEASURE_NAME
F_PLN_ORIG_RGLR_SLS_AMT MEASURE_NAME
F_PLN_ORIG_GRS_PRFT_AMT MEASURE_NAME
F_PLN_ORIG_RGLR_MKDN_AMT MEASURE_NAME
F_PLN_ORIG_CLRC_MKDN_AMT MEASURE_NAME
F_PLN_ORIG_PRMTN_MKDN_AMT MEASURE_NAME
F_PLN_ORIG_SHRK_QTY MEASURE_NAME
F_PLN_ORIG_SHRK_RTL_AMT MEASURE_NAME
F_PLN_ORIG_BOP_QTY MEASURE_NAME
F_PLN_ORIG_BOP_COST_AMT MEASURE_NAME
```

```
F_PLN_ORIG_BOP_RTL_AMT MEASURE_NAME
F_PLN_ORIG_RCPTS_QTY MEASURE_NAME
F_PLN_ORIG_RCPTS_COST_AMT MEASURE_NAME
F_PLN_ORIG_RCPTS_RTL_AMT MEASURE_NAME
F_PLN_ORIG_CMTS_RTL_AMT MEASURE_NAME
F_PLN_ORIG_ORD_CNCLLD_RTL_AMT MEASURE_NAME
F_PLN_ORIG_ORD_RTL_AMT MEASURE_NAME
F_PLN_ORIG_RECL_IN_RTL_AMT MEASURE_NAME
F_PLN_ORIG_RECL_OUT_RTL_AMT MEASURE_NAME
F_PLN_ORIG_RTV_RTL_AMT MEASURE_NAME
F_PLN_ORIG_CMTS_QTY MEASURE_NAME
F_PLN_ORIG_ORD_CNCLLD_QTY MEASURE_NAME
F_PLN_ORIG_ORD_QTY MEASURE_NAME
F_PLN_ORIG_RECL_IN_QTY MEASURE_NAME
F_PLN_ORIG_RECL_OUT_QTY MEASURE_NAME
F_PLN_ORIG_RTV_QTY MEASURE_NAME
F_PLN_ORIG_EOP_RTL_AMT MEASURE_NAME
F_PLN_ORIG_EOP_QTY MEASURE_NAME
F_PLN_ORIG_ORD_COST_AMT MEASURE_NAME
F_PLN_ORIG_ORD_CNCLLD_COST_AMT MEASURE_NAME
F_PLN_ORIG_CMTS_COST_AMT MEASURE_NAME
F_PLN_ORIG_CUM_MKUP_PCT MEASURE_NAME
F_PLN_ORIG_COGS_AMT MEASURE_NAME
F_PLN_ORIG_EXCL_SLS_VAT_AMT MEASURE_NAME
F_PLN_ORIG_EMPLY_DISC_AMT MEASURE_NAME
F_PLN_ORIG_FRGHT_COST_AMT MEASURE_NAME
F_PLN_ORIG_WRKRM_COST_AMT MEASURE_NAME
F_PLN_ORIG_RTRNS_SLS_AMT MEASURE_NAME
F_PLN_ORIG_EOP_COST_AMT MEASURE_NAME
```

The following table provides information about the output file data format.

Field	Start	Width	Format
DAY_DT EOW Day	0	8	Date
DEPT_IDNT	6	4	String
CLASS_IDNT	10	4	String
SBCLASS_IDNT	14	4	String
LOC_IDNT LOCATION_LABEL	18	10	String
MEASURE	28		Float

Note: The following must be defined in the shell environment prior to calling this script:

RPAS_HOME

RPAS_INTEGRATION_HOME

RDF and Merchandise Financial Planning Extract Matrix

The following matrix identifies the extract scripts and schemas used for each the data files produced for either RMS or Price:

Directory	Script or Schema Name	Forecasts and Standard Deviations	Diff Profiles	Receipt Plan	Markdown Budget	Store Grades	Current Plan	Original Plan
common	config.ksh							
	functions.ksh	X						
	header.ksh	X	X	X		X		
curve	profile_e_alloc.ksh		X					
grade	grade_e_rms.ksh					X		
plan	Plan_e_alloc.ksh			X				
	Plan_e_price.ksh				X			
	Plan_e_plcblwdm.ksh						X	
	Plan_e_ploblwdm.ksh							X
rdf	rdf_e_rms.ksh	X						
	rmsl_forecast.ksh	X						
	rmsl_forecast_daily.schema	X						
	rmsl_forecast_weekly.schema	X						

B

Appendix: Curve Administration Guide

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]
```

Arguments

- `-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:
      i. The Profile Type must NOT be diff(8).
      ii. The Aggregation Intersection must be above the Profile Intersection.
      iii. The Aggregation Intersection must be above the Stored Intersection.
   h. If Aggregation Intersection does not conform to Profile Intersection:
      iv. The Profile Type must be Diff (8).
      v. There must be at least one common hierarchy between the Aggregation Intersection and X in {Profile Intersection, Stored Intersection}.
      vi. 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:
      i. The Profile Type must NOT be diff(8).
      ii. The Aggregation Intersection must be above the Profile Intersection.
      iii. The Aggregation Intersection must be above the Stored Intersection.
   f. If Aggregation Intersection does not conform to Profile Intersection:
      iv. The Parent Profile Type must be Diff (8).
      v. There must be at least one common hierarchy between the Aggregation Intersection and X in {Profile Intersection and Stored Intersection}.
      vi. 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.

Curvebatch

Usage

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

Arguments

- `-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.