Retek® Predictive Application Server™ 11.1.7

Administration Guide

Corporate Headquarters:

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

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

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

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

Fax:
+44 (0)20 7563 46 10

Printed in the United States of America.

Customer Support

Customer Support hours

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

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

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

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

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

Mail

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

When contacting Customer Support, please provide:

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

Contents

Chapter 1 – Introduction

Overview

All Foundation-based products require setup and 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 Retek Predictive Solutions. System administrators use the administration workbooks to:

- 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 groups
- Security Administration Workbook – a workbook for setting up and maintaining user/template, user/measure, and template/measure access rights, as well as setting position-level security (if enabled)
- 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 is the most common, and it allows a user to make multiple edits in a workbook before recalculating the data. In this mode, the edits are effectively “queued” and executed once the user hits “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 Retek recommends that users operate in the “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, or hit the Calculate button on the tool bar

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

1. From the File menu, select Export Sheet. The Save As dialog box is displayed.
2. In the Save In field, select a directory on your computer where you want to save the export file.
3. In the File Name field, type a name for the export file.
4. Click the Save As Type drop-down list, and select a file type for the export file.

5. Make a selection for each of the following:
   - 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.
   - Labels – Specify the format of the label headers across the top of 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 at the top of each cell
     - Descriptions – specifies whether to identify dimensional positions in your output file with concise system names (for example, SKU00012) or the descriptive labels (for example, Cashmere Sweater – L – beige) that is assigned to each position
6. Select Save to export the file.
7. 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 in the context of establishing access to alerts.

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

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

1. From the Edit menu, select Insert Measure.
2. 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. Then you can toggle the display between the metric name (the system name) and the metric label as it appears in the workbook.

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

Global domain

Overview

"Global Domain" is a type of domain structure that provides 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.

There are two primary functional benefits in using a Global Domain environment. The first feature is the ability 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 responsibilities and roles. 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.

Centralized configuration and administration are the other primary features of Global Domain. 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 it centrally so that the local domains reference it in the master.

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: across the local domains or in the master domain.

Measure data that is stored in local domains is split across the 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 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. In this scenario, data for measures that have a base intersection in the Product hierarchy at or below Department (other hierarchies are irrelevant for this discussion) is stored in the local domain. This is based on which Department the underlying position in the Product hierarchy belongs to.

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, no Product Hierarchy) cannot be split across the local domains. These measures will reside in the master domain, and they will be accessed from there when these measures are required in workbooks.

All measures will be registered in the master domain and they will be 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.

Chapter 2 – Domain administration

Overview

The RPAS DomainDaemon is a process that enables the communication between the RPAS client and server when users initiate an on-line session.

The DomainDaemon runs on the server on a specified port and waits for requests from RPAS clients that are looking for that port. Once the DomainDaemon receives a request from a client, it starts a server process (RPAS Database Server or RpasDbServer) to which the client connects. From this point on, the client communicates directly with the RPAS Database Server. The system administrators may choose to have one single DomainDaemon process for all users, or they may choose to have separate processes per domain, per enterprise, etc.

The DomainDaemon is installed in the [RPASDIR]/bin directory where [RPASDIR] stands for the full path to the directory in which the RPAS server is installed. The system administrators can start, stop, and monitor the DomainDaemon processes by using a variety of DomainDaemon commands.

Starting the DomainDaemon

In order to start the DomainDaemon, execute the DomainDaemon utility in the sub-directory “\bin” where the RPAS server is installed. The port number on which 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 in which the RPAS server is installed.

For 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 a user’s computer crashes or if the user’s network connection is lost.

If this situation occurs, the user should notify their system administrator if they 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 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.

The user can then log back into the RPAS client.

showActiveServers

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

For each active server, the DomainDaemon shows the process ID, the domain, and user the 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 on a regular basis 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. Being 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 (60000 milliseconds) 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 two lines below indicate 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 and can only be run in the master domain of a Global Domain environment. See the individual sections for additional information.

- Alert Manager dialog window – results of the alert finder run on the global domain are collated and displayed
    - Applies for all alerts registered in the global domain
    - Results will be based on data from all the individual local domains
    - Results are consolidated (added together) to display a single result per measure
- Alert manager workbook template – used to build alert workbooks from the Alert Manager dialog window; data will be retrieved from the local domains
- Measure Analysis – for analyzing measure data from local domains
- Security administration – ability to set security by template, measure, and positions; this workbook template can only be used in the master domain and has been disabled for use in local domains
- User administration – user information will be set up and maintained in the global domain, but will be replicated to the local domains; updates will be effective immediately after the changes are committed; this workbook template can only be used in the master domain and has been disabled for use in local domains
- Translation Administration – template used for modifying the labels of translatable data in RPAS; this workbook template can only be used in the master domain, and it has been disabled for use in local domains
- Hierarchy maintenance – for setting up and maintaining positions of user-defined dimensions (user-defined dimensions must be registered in the Global Domain using utility reguserdim)

RPAS utilities

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

- Alerts (alertmgr)
  - 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.

  - 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) – it is required that hierarchy information be centrally administered in the Global Domain and replicated to the local domains
- Reshaping arrays (reshapeArrays) – execution of this administrative utility in the global domain will perform this function on all local domains
- 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
- 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
  - Measures (regmeasure)
  - Templates (regtemplate)
  - Functions (regfunction)
  - Token measures (regTokenMeasure)
  - User-defined dimensions (reguserdim)

Chapter 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 to perform (and possible to perform) the administrative activities included in this section in the “master” domain.

User logon security

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

This will prevent the user 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.

A user account may be marked as **must change password**.

This is useful for brand-new accounts. The user will be allowed to logon with the current password, but will immediately be prompted to select a new password.

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

Account Lockout may be enabled for a domain.

Once the domain administrator selects a number of failed logon attempts, 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 using the –lockAccount flag.

Password expiration may be enabled for a domain.

The domain administrator selects a certain number of days, which determine when passwords expire. When a user logs in, if the configured number of days have passed since this user entered a new password, the system requires a new password to be entered.

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 a user attempts to change passwords, the system will not permit the use of any password that is already stored in the password history.

Password history may be enabled through the domainprop utility by 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 that 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 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. 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 all users, for users in a group, or for individual users.

Position level security can be defined at levels (dimensions) that are at or above the base (such as class in the product hierarchy) in any hierarchy other than calendar. As positions are added at a level/dimension that is 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. Basically, if security is maintained at the subclass level, users are automatically granted access to all of the SKUs in a given subclass if they have access to that subclass. This includes those that were added after security has been 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 **inithier** utility. See the RPAS Configuration Tools User Guide or the section for Defining hierarchy settings – **inithierr** for more information.

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. One sheet controls access to positions for all users (default) (for instance, Prod Security Default); one controls access to positions by user group (for instance, Prod Security Group); and the third controls access to positions by individual user (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 table below demonstrates how access is granted or denied based on all combinations of settings:

Security setting by Position **(Denied = False, Granted = True)**			**Based on settings user is Granted or Denied access**
User	**User Group**	**World**	**Resulting access for a given position**
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 a 2-tree wizard, which ultimately limits which positions a user can include in the workbook build.

Workbook security

Currently, workbook access is either granted or denied. If users have been granted access to a workbook, they 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 entire 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, this means that administrators have access to all workbooks that are saved with world access. If a workbook is saved with group access, administrators can access the workbook only 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 a user begins the workbook build process. If the user’s limit has been reached, an error message appears that informs the user that the workbook build process cannot be completed because the user has reached their limit. The message also informs the user what that limit is. The wizard process then terminates.

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 to measures within workbooks through Security Administration. Additionally, you can 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, and click OK.
4. 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, must contain no spaces (the underscore character is acceptable), and must not exceed a total of eight characters.

5. In the User Label field, type a label that describes the user. For example, the user’s full name. This identifying label will appear 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 they appear on the Forecast Approval worksheet to specify which user approved a given forecast.
6. In the Default Group field, select the user group to which the user will belong.
7. If a user will belong to more than one group, select the additional groups from the list in the Other Groups field.
8. In the Password field, type a password for the user.
9. In the Password Verification field, type the same password.
10. If the user should have Admin status, which allows the user to create system-wide default styles for workbook templates, 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, note that a user’s Admin status can later be modified on the User’s worksheet of the User & Template Administration workbook.

Note: Granting Admin status to a user gives the user access to all workbook templates, but it does not automatically give the user access to all workbooks.

11. If the user must change his/her password when logging on for the first time, check the Force Password Change box.
12. If you want to temporarily disable the user’s account, check the Lock User Account box.
13. 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 & 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 you assign new users 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 Retek 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, and click OK.
4. In the Group Name field, type a name for the group.
5. In the Group Label field, type a descriptive label for the group. This label is displayed when referring to the group throughout RPAS.
6. 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, and click OK.
4. Select the name of the user that you want to delete.
5. 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.

Note: 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, and click OK.
4. Select the user group that you want to delete.
5. 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, and click OK.
4. Select the user that you want to edit, and click Next.
5. Make the necessary changes to the user’s profile. You can change anything, except the User Name. See “Add a user” for details.
6. Click Finish to save the changes you have made.

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. This means that you can determine which templates individual users can access, as well as the measures that users can access while manipulating workbooks in the system. Additionally, you can specify and restrict the measures that are available to be added to a given workbook template. The ability to set 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.

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/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 used to set up and maintain 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 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 before they can 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 that are 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 through the Insert Measure command. See "Insert measures into an open worksheet" on page 3 for more information.

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 that is located at that template/measure intersection.

After changes are made to a template's profile, they must be committed before they can 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, which integrates the current user/template and template/measure settings and applies 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 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 the user from accessing a given measure even though that measure is normally permitted to be displayed in a particular workbook template to which the user does have access.

After changes are made to a user’s profile, the changes must be committed by the administrator before they will take effect for that user.

Note: The Measure Rights worksheet contains only public measures. That is, measures that can be optionally included in a worksheet depending on the 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, the changes must be committed by the administrator before they will take effect for that user.

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, the changes must be committed by the administrator before they will take effect for that user.

Workbook template limits worksheets

The Workbook Template Limit worksheets are used to limit the number of workbooks that a user can have saved. Limits can be set for a given template and are defined per user, user group, or all users (all for a given template). Defined limits (any value other than 1 billion) are evaluated in the above order, which means that a limit that is defined at user-template will override any values that are defined at group-template or template. If the above limits are not defined, the default value is one billion.

The limits are checked when a user begins the workbook build process. If the user's limit is reached, an error message appears that informs the user that the workbook build process cannot complete because the user has reached their limit. The message also tells what that limit is. 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. Select the Security Administration workbook template, and click OK.

Set or modify users' access to workbook templates

1. From the File menu, select New.
2. Click the Administration tab.
3. Select Security Administration, and click OK.
4. On the Workbook Template Rights worksheet, select each template for which a user needs access rights. For templates that the employee should not have access rights to, ensure that there is no check mark.
5. Changes must be committed to the master database for them to take effect. To commit your changes, select Commit Now from the File menu.
6. If you desire, save your workbook by selecting Save from the File menu.
7. 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, and click OK.
4. 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, ensure that there is no check mark.
5. Changes must be committed to the master database for them to take effect. To commit your changes, select Commit Now from the File menu.

6. If you desire, save your workbook by selecting Save from the File menu.
7. 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, and click OK.
4. On the Measure Rights worksheet, select either Read Only or Read/Write from the drop-down list for each measure that a user should have access to. For measures that the user should not have access to, ensure that Denied is selected.
5. Any changes made must be committed to the master database before they will take effect. To commit your changes, select Commit Now from the File menu.
6. If you desire, save your workbook by selecting Save from the File menu.
7. 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, and click OK.
4. 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, ensure that there is no check mark.
5. Any changes made must be committed to the master database before they will take effect. To commit your changes, select Commit Now from the File menu.
6. If you desire, save your workbook by selecting Save from the File menu.
7. 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, and click **OK.**
4. Select the worksheet for which security needs to be set or modified: User, User Group, or World.
5. By default the dimension (level) at which position level security has been enabled will be displayed. If you want to manage security at a level that is above the designated level (only levels above are possible), right-click and “Select Rollup” to view the available dimensions.
6. 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 in order to have access to a position.

7. For changes to take effect, they must be committed to the domain before exiting.

Limit the number of workbooks that a user can have saved

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

Chapter 4 – Hierarchy maintenance

Overview

Hierarchy maintenance workbook

Retek 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 Retek Predictive Solutions are installed, implementation scripts define the dimensions and hierarchical structures that are specific to your organization. For example, the system can be built to recognize that SKUs roll up into styles, and styles roll up into product classes within the product hierarchy. Occasionally, you might want to group products according to some ad hoc personal design in order 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, which mean that they allow you to roll data up from lower levels of aggregation along the hierarchical paths that you define.

Hierarchy maintenance workbook example:

Assume that your experience has shown that the accuracy of forecasts for your top 50 products (A products) reflects the relative accuracy of all forecasts, so you would like to group elements within a user-defined dimension as the top 50 products by designating them ‘A Products.’ When you select products in a wizard or you 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 that 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 it is set up by your company at the time of installation. The examples in this section refer to the Product hierarchy, but other hierarchies can be maintained in the same way.

Hierarchy maintenance example

Suppose that 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.

In order 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 that is used in the following example is named Product Status. To maintain this user-defined dimension, you use the Hierarchy Maintenance Wizard.

Hierarchy maintenance wizard

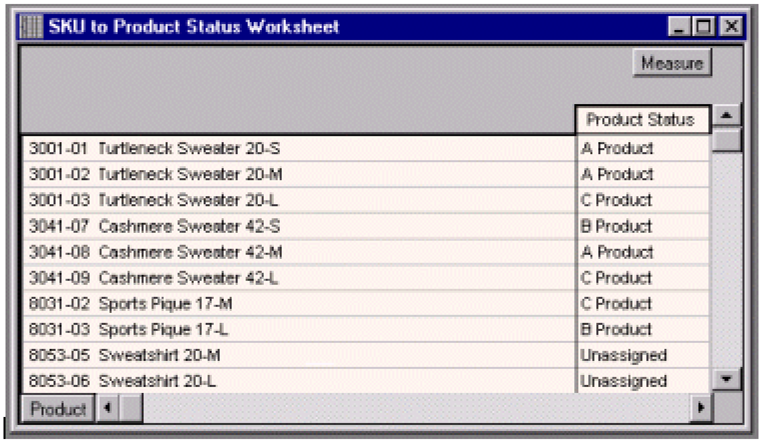
The first step in maintaining hierarchies is to access the Hierarchy Maintenance Wizard. In this wizard, you select the SKUs that will be mapped to the various positions of your user-defined dimension. Responses to prompts in the wizard are used to format a new Hierarchy Maintenance Workbook.

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 only provides three positions (or values) in the Product Status dimension; however, you can enter any character string in an individual SKU's Product Status cell, and 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.



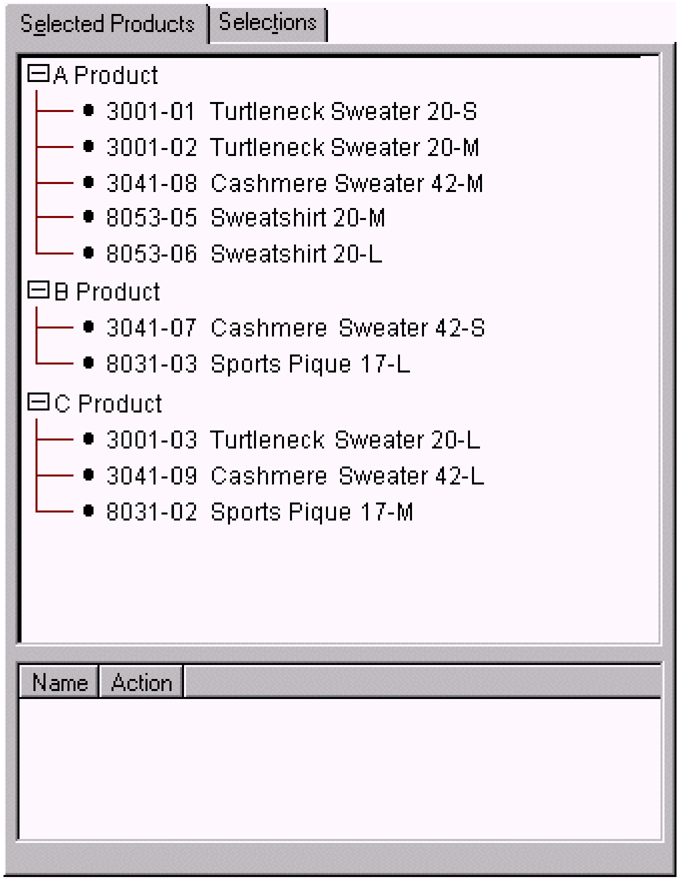
	Product Status
3001-01 Turtleneck Sweater 20-S	A Product
3001-02 Turtleneck Sweater 20-M	A Product
3001-03 Turtleneck Sweater 20-L	C Product
3041-07 Cashmere Sweater 42-S	B Product
3041-08 Cashmere Sweater 42-M	A Product
3041-09 Cashmere Sweater 42-L	C Product
8031-02 Sports Pique 17-M	C Product
8031-03 Sports Pique 17-L	B Product
8053-05 Sweatshirt 20-M	Unassigned
8053-06 Sweatshirt 20-L	Unassigned

Note: The Retek Predictive Solutions system is case-sensitive when a new position name (label) is entered in the Hierarchy Maintenance Workbook. After the workbook has been 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 figure below 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.



⊟A Product
- 3001-01 Turtleneck Sweater 20-S
- 3001-02 Turtleneck Sweater 20-M
- 3041-08 Cashmere Sweater 42-M
- 8053-05 Sweatshirt 20-M
- 8053-06 Sweatshirt 20-L

⊟B Product
- 3041-07 Cashmere Sweater 42-S
- 8031-03 Sports Pique 17-L

⊟C Product
- 3001-03 Turtleneck Sweater 20-L
- 3041-09 Cashmere Sweater 42-L
- 8031-02 Sports Pique 17-M

Procedures

Access hierarchy maintenance

Note: If a solution is built in a Global Domain environment, it is only required (and possible) to perform all of the administrative activities in this section in the “master” domain.

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

   OR

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

Maintain a user-defined dimension within a hierarchy

Use this procedure to create and map positions for user-defined dimension to positions of existing dimensions (defined during configuration, most commonly in product or location hierarchies). User-defined dimensions are distinct from those defined during the initial configuration of a solution and must be registered by the administrator.

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

8. 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.
9. To close the workbook, select Close from the File menu.

Chapter 5 – Measure analysis

Overview

Measure analysis workbook

The Measure Analysis Workbook template allows you to view data that is associated with any registered measure in the Retek Predictive Solutions applications (for instance, 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 Retek 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. Just choose the measures to be displayed in the new workbook.

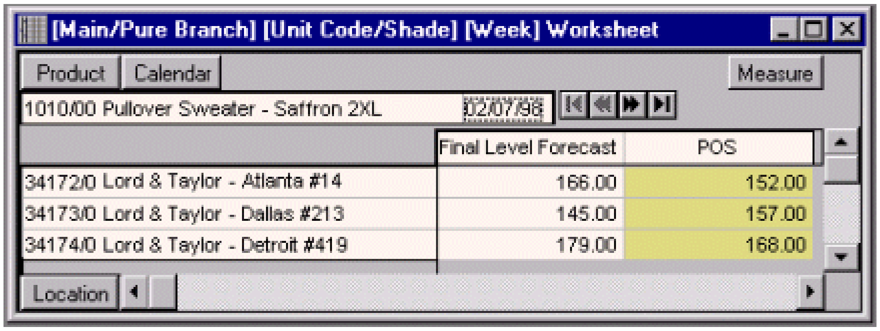
Measure analysis wizard

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

Measure analysis worksheet

The Measure Analysis workbook displays data that is 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, which depends on the measure selections that were made in the wizard. This dimensional intersection is shown in the worksheet title bar.



[Main/Pure Branch] [Unit Code/Shade] [Week] Worksheet

1010/00 Pullover Sweater - Saffron 2XL 02/07/98

	Final Level Forecast	POS
34172/0 Lord & Taylor - Atlanta #14	166.00	152.00
34173/0 Lord & Taylor - Dallas #213	145.00	157.00
34174/0 Lord & Taylor - Detroit #419	179.00	168.00

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 is registered as a read/write measure, so it 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 in order to bypass the Measure Analysis wizard and open an existing Measure Analysis workbook.

   OR

2. Select New from the File menu.
3. On the Analysis tab, select Measure Analysis, and 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
2. Double-click on the workbook to be opened
3. Go to step 15.

   OR

4. To open a new workbook, select New from the File menu.
5. On the Analysis tab, select Measure Analysis, and click OK.
6. The Measure Analysis Wizard opens and prompts you to select the registered measures to be displayed in the new workbook.
7. Use Ctrl-Click and/or Shift-Click to select multiple measures.
8. Click Next.
9. Select the dimensional level where you want to view measure data.
10. 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.

> **Note:** 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.

11. Select all calendar periods for which you want to view data. Click Next.
12. Select the locations whose measure data you want to view. Click Next.
13. Select the products whose measure data you want to view. 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 that you selected. Make any changes as required.
16. Commit your changes to the master database by selecting Commit Now from the File menu. 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.

Chapter 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. The workbook build process is automated, so users are spared the processing time that is required to regularly enter the same wizard selections each time a new workbook is built. Since the build process occurs overnight, users are spared the wait time that is 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 workbook auto build maintenance workbook

1. Select New from the File menu.
2. Select the Administration tab.
3. Highlight Auto Workbook Maintenance, and 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, and click OK.
4. From the task list, select Add Workbook, and click Next.
5. Select a workbook template type, and click Next.
6. Select an owner for the workbook, and click Next.
7. Fill in the workbook Build Label, the Build Frequency (in days) with which the workbook should be built, and the Next Build Date.
8. Specify the Saved Access for the workbook by selecting User, Group, or World.
9. Select the group that owns the workbook. You can choose from the list of groups to which you belong.

10. Click Next to initialize the wizard for the workbook template that was selected in step 5 above. The choices that you make are saved under the name that 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, and click OK.
4. From the task list, select Delete Workbooks, and click Next.
5. Select the workbook or workbooks to delete from the auto build queue.
6. Click Finish to delete the workbooks from the Auto Workbook Build queue.

Chapter 7 – Translation administration

Overview

All of the Retek Predictive Solutions can be presented in multiple languages. Every product, location, and calendar position can be displayed in multiple languages; just like messages presented through the client are displayed in multiple languages. 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 that is used in measure labels, workbook template names, template group names, user group labels, and general areas (such as, 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 that are found throughout RPAS. Translations into each of the system’s alternative languages are supported for objects such as wizard instructions, pick list options, system warnings, error messages, and menu options.

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 instance (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 that were set up by the system administrator. For products in the Retek 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, and 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, and click OK.
4. 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.
5. 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.
6. On the File menu, select Commit Now.
7. 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 enter 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 enter 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 enter 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 enter 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 enter the user group label to be translated.

Chapter 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. The procedures for using Commit ASAP are provided in the “RPAS 11.1 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 by using Commit ASAP (File\Commit ASAP), a user will see a message in the client that the workbook has been scheduled for a commit; and the user can continue with their work. The system will then try to commit the workbook as soon as it can, and it takes 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 that are 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 see only their 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 only see the processes with the specified statuses. The window can be updated by using the ***Refresh*** button.

Note: The dialog remembers the settings based on the last use.

Important notes:

- 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 be one pending commit ASAP in the queue for a given workbook/user/template name combination.
- 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 the entries will be removed. This option also requires specifications 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]*

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 that 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 Retek 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

Argument	Description
–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 settings are set by 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 the following day

Usage:

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

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	Display 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 it is 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. In general, these files are 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.

Chapter 9 – Batch processes and RPAS utilities

Overview

A collection of stand-alone executables and scripts that are used for a variety of operations are included with an RPAS installation. 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 that are 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.

Centralized utilities

In a Global Domain environment most of the RPAS utilities can only be run on the master domain. These changes are propagated to the local domains, or they are 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**)
    - 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.

-
    - 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**) – It is required that hierarchy information be centrally administered in the Global Domain and replicated to the local domains.
- Reshaping arrays (**reshapeArrays**) – Execution of this administrative utility in the global domain will perform this function on all local domains.
- Users (**usermgr**) – The back-end utility for managing users will propagate changes from the global domain to local domains. Ultimately this will be updated for the RPAS Administrative Template “User Administration.”
- Domain properties (**domainprop**) – Manipulating properties, such as specifying password properties, and locking user accounts.
- Defining hierarchy settings (**inithier**) – For setting position-level security and setting the width of positions for a particular dimension.
- 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.
- Measure attributes (**regmeasattr**) – Registering attributes of measures defined in the domain.
- Dimension attributes (**dattrmgr**) – Registering attributes of existing dimensions.
- Change domain sparsity (**changedomainsparseness**) – Change the sparsity setting of an entire domain (between sparse and hypersparse). This process should only be run with guidance from Retek Services of Retek Customer Support.

Common information & parameters for RPAS utilities

A number of standard parameters are available for most RPAS utilities. 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 (e.g. RPAS 11.1.0). It does not require –d domainPath.
–d *pathtodomain*	This specifies the path to the domain against which the utility will run.
–loglevel	Use this argument to set the logger verbosity level. Possible values are none, error, warning, information, or all. See additional information below regarding logger verbosity settings.
–n	Certain utilities contain this parameter to perform a dry run. Using this option will show the administrator what would change, but it 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.

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 – This outputs error messages only.
- warning – This outputs warnings in addition to error messages.
- information – This outputs status and progress of the operation in addition to the error and warning messages.
- all – This outputs all available information that was generated by the utility. This includes error, warning, and informational messages.

Each output line of the above feedback types 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 that the message is an error. "W" indicates the message is a warning. "I" indicates that the message is informational.

Logging batch processes

RPAS batch processes may run for a long time, so it is often difficult to know the step that are currently being performed. Fortunately, the RPAS server code provides a large amount of detailed output messages from its processes in order 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) that are 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 Retek 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 that are followed by the total number of records read and related information. Here is part of the output from a typical data loading batch job.

```
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
```

Chapter 10 – Data management

Hierarchy management

Loading hierarchies – loadHier

The loadHier utility can be 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 is set by a configuration file before a domain is built. The width of positions can be increased after a domain has been built using the **inithier** utility.

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.***, where **hier** is name of a registered hierarchy. The “**.***” indicates that all extensions of a hierarchy file are loaded for a given hierarchy. 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 –version

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

Argument	Description
–version	Use this argument to get the version information. It does not require –d domainPath.
–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.

Argument	Description
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Updating domains with hierarchy changes – reshapeArrays

The **reshapeArrays** utility is used to make arrays conform to the current hierarchies in the domain. Any positions added to dimensions as a result of running loadHier will be added to arrays that have this dimension. Any positions that are removed from dimensions will also be removed from the arrays that have this dimension. **reshapeArrays** will update the arrays to reflect these changes that were made in the hierarchies.

The purge option can be used to avoid creating backup copies of all the databases. For large environments, this might be desired to avoid the effect of excessive backups. Also, if a customer has a separate backup/restore process (for instance, another application that regularly backs up the system data) the backup gem files likely are not needed, and the purge option should be used.

Usage:

reshapeArrays –version

reshapeArrays –d *domainPath* –registered {–purge} {–processes *max*}

reshapeArrays –d *domainPath* –db *dbName* {–array *arrayName*} {–purge}

Argument	Description
–version	Use this argument to get the version information.
–d *domainPath*	This argument specifies the domain that contains the arrays to be reshaped.
–registered	Use this argument to reshape all registered arrays. You must specify either –registered or –db, but not both.
–processes *max*	In a Global Domain environment this will run the utility with the specified number of parallel processes on the master domain and local domains. If this parameter is used with the –db option or in a simple domain, –processes will be ignored.

Argument	Description
–db *dbName*	If you specify –db, all arrays in the specified database are reshaped. You must specify either –registered or–db, but not both. dbName can specify either a full or relative path to the database. If the path is relative, it is relative to the root of the domain.
–array *arrayName*	This argument specifies which arrays in a database are to be reshaped. This argument can be repeated to reshape multiple arrays. If –array is not specified, all arrays in the specified database are reshaped. –array can be repeated on the same command line to reshape multiple arrays.
–purge	If –purge is not specified, the original database will be copied to dbName.bak.

Reconfiguring global domain partitions – 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 with the loadhier utility, a special process must be followed for positions at the partition level of a Global Domain environment.

The following information is only relevant if an environment has been built as a Global Domain:

Consider a Global Domain environment that is partitioned at the “department” dimension/level in the product hierarchy. This environment could handle the addition or removal of SKUs, classes, or divisions without having to follow a special process; however, if a new department is to be added to the environment, an administrator must run a special RPAS utility to add this position before it can be loaded into the hierarchy.

The following process must be followed to add or remove a position at the partition level of a Global Domain environment:

1. The administrator must be notified in advance that a new position is being added to or removed from the system.
2. Run the utility **reconfigGlobalDomainPartitions** to add/remove the positions by specifying the local domain to which the positions do belong or will belong.

3. Load the updated hierarchy file (that either contains the new positions or does not contain the removed positions) using the **loadhier** utility. The next run of **loadhier** following the addition or removal of positions by using the **reconfigGlobalDomainPartitions** utility must contain the updated hierarchy file or **loadhier** will fail.
4. Run **reshapearrays** utility to update the domain to reflect the changed positions.

Note: This only applies to positions at the partition level. Positions can be added or removed below or above the partition level by being added or removed from the hierarchy file.

reconfigGlobalDomainPartitions –d *pathToMasterDomain* –add *posName1,posName2*, ... –sub *pathToSubDomain*

reconfigGlobalDomainPartitions –d *pathToMasterDomain* –remove *posName1*, *posName2*, ..

Argument	Description
–d *pathToMasterDomain*	Specifies the path to the master domain in a Global Domain environment
–add *posName1*, *posName2*,	Adds one or more positions to a specified local domain The path to the local domain must follow the list of positions to add
–remove *posName1*, *posName2*, ...	Removes the designated positions from the local domain to which the positions belong
–sub *pathToSubDomain*	Specifies the path to the local domain to which new positions are being added

Notes:

- Position names are separated by commas, and they must be valid **external** position names without the prefix of a dimension.
- The path to the local domain does not need to be specified if removing positions.
- Assumptions and limitations:
  - Positions can only be added or removed in EXISTING local domains.
  - The user must specify the local domain to which the positions are being added.
  - The user must call this utility before loading the hierarchy.
  - The user must run reshapeArrays utility after loading the hierarchy.
  - The remove option will NOT remove the local domain if the local domain does not have any positions.
  - Multiple positions can be added to a local domain in a single call to the utility.

- If new positions are included in the hierarchy load file prior to running the loadhier utility, RPAS will reject the entire load for that hierarchy file, generate an error message, and use the previous load.

Defining hierarchy settings – inithier

The inithier utility should be used for the following activities:

- Turning on or off position-level security – This can also be controlled in the Configuration Tools.
- Extending the width of positions for a dimension – The default value is 24. This value can be increased if necessary.

For information about position-level security, refer to the write-up on page 14.

🕮 **Note:** Position-level security can also be established in the RPAS Configuration Tools.

Position-level security must be set for a specific dimension of a hierarchy. Use the following command to enable or disable this setting:

inithier –d *pathToDomain* –h *hierarchyName* (–securityDim *dimName* |
–noSecurity) {–loglevel *level* }

The other use of the **inithier** utility is to extend the length of position names for a dimension to a value greater than the default value of 24 characters. RPAS 11.1 provides the ability for any dimension to have position names longer than 24 characters.

To extend the length of position names, use the following command:

inithier –d pathToDomain –dim dimensionName –width width

🕮 **Note:** The designated length of the position names must correctly match the associated data of those positions names in the input data files.

🕮 **Note:** The length can only be increased and cannot be decreased after a change has been made.

The list of all arguments for **inithier** is provided below.

Argument	Description
–version	Use this argument to get the version information.
–d *domainPath*	This argument specifies the path to the domain.
–h *hierarchyName*	This argument specifies the name of the hierarchy for which position-level security is enabled or disabled. Use –h with either –securityDim or –noSecurity.
–securityDim *dimName*	This argument specifies the name of the dimension for which position-level security is enabled.

Argument	Description
–noSecurity	Use this argument to disable position-level security for the hierarchy specified using the –h argument.
–dim *dimensionName*	This argument is used to specify the dimension for which the width of position names is to be increased.
–width *width*	This argument specifies the new width of positions for the specified hierarchy. This must be an integer value greater than the previous setting (default is 24 characters).
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Renaming positions – renamePositions

RPAS provides the ability to change the name of a position by 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 with the following usage:

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

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 (e.g. 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 (default file name is hierRename.log)
–n	Do not apply changes but generate report that identifies changes that would be applied

Note: This note is about the following about the input file:

- There will be three columns of data in each input line that correspond to dimension name, old position name, and new position name. The three fields will be tab-delimited. Any line that is not formatted this way will be ignored. Empty lines are also ignored.
- Old position names must be an existing position name.
- New position names cannot be an external name that is already used or 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 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.

Note: This note is about the following about the utility:

- 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.

Measure data

Loading measure data – loadMeasure

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

Usage:

- **loadmeasure** –d *pathToDomain* –measure *measureName* {–logdirectory *directoryName*} {–applyloads} {–loglevel *level*}

Argument	Description
–d *pathToDomain*	This specifies the domain in which to load the measure.
–measure *measureName*	This specifies the name of the measure to load. The name must be lowercase.

Argument	Description
–logdirectory *directoryName*	This 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.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

The loadMeasure utility also provides functionality to automatically run scripts before and after the utility is executed. These are referred to as pre 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.

Splitting measure data input files – splitMeasureData

In a Global Domain environment, RPAS provides the **splitMeasureData** utility to load a single input (measure) file into the master domain, and it splits it into multiple input files for each of the local domains. These files can then be loaded by using the **loadMeasure** utility. This feature reduces the effort in administering measure input files, and it avoids duplicating files in the local domains.

The **splitMeasureData** utility splits the input file into multiple files, which are based on the assignment (mapping) of positions to the local domains that are included in the base intersection of the measure. The utility can process multiple input files for the same measure in a single call (for example, loads “measurename.*” such as sales.ovr, sales.ovr.1, and sales.inc). Measures with a base intersection above the partition level are not split, and they are stored only in the master domain.

The **splitMeasureData** utility should be run in the master domain prior to running **loadMeasure** in the local domains.

📖 **Note:** This utility is only applicable in a Global Domain environment.

splitMeasureData –d *pathToGlobalDomain* –measure *measureName*

Argument	Description
–d *pathToDomain*	This specifies the domain in which to load the measure.
–measure *measureName*	This specifies the name of the measure to split. The name must be lowercase.

Exporting measure data – exportData

Use exportData to export measure data from RPAS into text files. Each exported line 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.

The parameters specify the arrays and dimensions that 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 the cell value and the 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. The order that the arrays appear in the –array parameter is the order they will appear in the export file.

After the arrays have been specified, you must 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 it lines 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 it will 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.

The –wide 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.

The –skipNA parameter is used for the NA suppression option. This option controls whether data is exported based on it having NA’s in the export row. The choices for the –skipNA parameter are ANYNA, ALLNA, ALWAYS, or ARRAYNA. The ANYNA value will cause a row of data not to be exported if any cell contains a NA value. The ALLNA value will cause a row of data not to be exported only if all values in the row are NA. The ALWAYS value will export data regardless of whether it contains NAs. The ARRAYNA value will cause a row of data to not be exported if the value in the given array name is NA. If the ARRAYNA value is set, the –naArray parameter must be given. This parameter is followed by the array name to be used with the ARRAYNA value.

The –precision 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. The utility returns a warning if a value greater than one is provided.

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 the parameters needed for the export.

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

Usage:

exportData –version

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}

Argument	Description
–version	This prints the RPAS version, revision, and build information.
–d *domainPath*	This specifies the domain containing the data you wish to export.

Argument	Description
–out *outputFile*	This 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*\”	This 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.
–array \”*arraySpec*\”	This 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 that arrays are listed is the order that they will be exported. 🕮 **Note:** This argument cannot be used in a Global Domain environment, and it can only be used in simple domains.

Option	Description
–dim \"*dimSpec*\"	This 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, the dimension is not exported. The –dim parameter can be repeated.
–skipNA always|allna|anyna|arrayna	This controls whether a line of data is exported based on having NAs in a cell. *always* exports data regardless of NAs. *allna* does not export if all columns are NA (default). *anyna* does not export if any column is NA. *arrayna* does not export if cell value in an array is NA (requires –naArray).
–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*	This 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.

Option	Description
–range *start:end*	This is used in conjunction with the –wide parameter, and it specifies a range along the innermost dimension. Only values in the range are considered for export.
–append	This specifies that output is appended at end of output file. The default is to overwrite output file.
–wide	This exports data in innermost dimension across instead of down.
–time	This specifies a YYYYMMDD format for dates.
–precision *precisionValue*	This skips the export of values that vary from the NA value less than the specified precision value.

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 into 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 will be performed outside of this utility.

The usage of this utility is as follows:

- **mapData** -srcDomain *srcDomainPath* -destDomain *destDomainPath*

  {-db *dbName* {-array arrayName}} {-loglevel}

Argument	Description
-srcDomain *srcDomainPath*	This specifies the full path of the source domain.
-destDomain *destDomainPath*	This specifies the full path of the destination domain.
-db *dbName*	This instructs **mapData** to map data only on the given database.
-array *arrayName*	This instructs **mapData** to map data only on the given array within the given database specified by **–db**.

Argument	Description
-loglevel	This sets the logger verbosity level.

- Run **mapData** without the **-db** argument to map the entire domain.
- To set the logger verbosity level, use **-loglevel**. Possible values include **all**, **profile**, **debug**, **information**, **warning**, **error**, or **none**.

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 that is 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*}

OR

updateArray –argFile *filename* {–version} {–loglevel *level*}

Argument	Description
–destArray *dbPath.arrayName*	Required argument. Specifies the destination array. dbPath is relative to destDomain.
–srcArray *dbPath.arrayName*	Optional argument. Default is no source array. **Note:** This parameter cannot be used with –srcScalar scalarCell.

Argument	Description
–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.
–loglevel *level*	Use this argument to set the logger verbosity level.Possible values: all, profile, debug, information, warning, error, or none.
–version	Use this argument to get the version information.

Chapter 11 – Miscellaneous RPAS utilities

Operational activities

Alerts

Overview

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 registered in the domain. 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 that are 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, the user can view the identified alerts in the Alert Manager in the RPAS client.

The following is a high level description of the process for defining an alert.

- Register an alert measure – This must be a Boolean measure (values are true-false, yes-no, or on-off), and they must be registered with the RPAS utility **regmeasure** or in the RPAS Configuration Tools.
- Register the alert and the expression for which the alert should be evaluated. This flags the registered measure as an alert so that it is recognized when the "alert finder" is run.
- The "alert finder" is run on the domain, and it determines when the value of the alert expression is true.

Registering alert measures – regmeasure

Alert measures can be registered through the RPAS Configuration Tools or through the RPAS utility **regmeasure**. The type of this measure must be Boolean, which allows for values of true/false, yes/no, or on/off.

Note: This utility is centralized in a Global Domain environment, and it can only be run on the master domain.

regmeasure –d *pathtodomain* –add *alertMeasName* –type boolean –baseint *baseintersection* –db *databasename*

Argument	Description
–d *pathdodomain*	This specifies the path to the domain where the alert measure is to be registered.
–add *alertMeasName*	This adds the new measure to the domain using the designated name. **Note:** This name must be less than or equal to 15 characters.
–type boolean	This registers the alerts as a Boolean type measure.
–baseint *baseintersection*	This specifies the base intersection of the measure.
–db *databasename*	This specifies the name of the database where the measure will be stored.

Designate alert and define alert expression – alertmgr

Once the measure has been registered, the measure must be identified as an alert measure, and it must have an expression defined for which the alert condition will return a value of true or false.

This process is completed using the RPAS utility **alertmgr**. See the section below for the usage.

Find alerts on the domain – alertmgr

The “alert finder” is run in batch to evaluate all registered alert measures in the domain. This process is most commonly added to a batch process that is regularly run (either nightly or weekly).

alertmgr –d *pathtodomain* –findAlerts

alertmgr utility

The following is the complete usage of the RPAS utility **alertmgr.** Use the **alertmgr** utility to:

- find alerts
- register and unregister alerts
- add expressions to alerts
- create and label alert categories

alertmgr –version

alertmgr –d *pathToDomain* –findAlerts {–loglevel *level*}

alertmgr –d *pathToDomain* –register *alertName* {–category *catName*} {–categoryLabel *label*} {–expression \"*expr*\"} {–loglevel *level*}

alertmgr –d *pathToDomain* –unregister *alertName* {–loglevel *level*}

alertmgr –d *pathToDomain* –addExpression *alertName* {–expression \"*expr*\"} {–loglevel *level*}

Argument	Description
–d *pathToDomain*	This specifies the directory in which to run the utility.
–version	Use this argument to get the version information. It does not require –d domainPath.
–findAlerts	This finds all of the alerts in the specified domain.
–register *alertName*	Use this argument to register an alert. Requires –category, –categoryLabel, and –expression. **Note:** alertName must be the name of a registered Boolean measure.
–unregister *alertName*	Use this argument to unregister an alert. alertName must be the name of a registered Boolean measure.
–addExpression	Use this argument to add an expression to the alert. Use –expression to specify the expression.
–category *catName*	Use this argument to specify a category for the alert.
–categoryLabel *label*	Use this argument to specify a label for the alert category.

Argument	Description
–expression \"*expr*\"	Use this in conjunction with –addExpression and –register. expr can be any valid RPAS 11 expression.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Copying domains – copyDomain

The **copyDomain** utility is used to copy a simple domain or all domains that are included in a global domain implementation.

For a standard, simple domain (for instance, non-global), **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 arrays so that the domains are still properly connected together.

Relative paths are supported with this utility, and they are used when creating the new copies of all of the underlying arrays. Relative paths are based on the full pathname of the domain’s root directory.

Usage:

copyDomain –d *pathToSrcDomain* -dest *pathToDest* { -f }

copyDomain –version

Argument	Description
–d *pathToSrcDomain*	This specifies the path of the domain to be copied.
–dest *pathToDest*	This specifies the path to where the domain is to be copied.
–f	This forces the deletion of the existing destination path before copying.

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 –version

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

Argument	Description
–d *pathToDomain*	This specifies the domain that you wish to manipulate.
–version	Use this argument to get the version information. It does not require –d domainPath.
–expirePassword {*days*}	This is used to set or view the number of days a password is valid. If a number follows the argument, it sets the number of days that a password is valid. Otherwise, it prints the current setting.
–passwordHistory {*oldPasswordCount*}	This is used to set or view the number of previous passwords that are kept in order 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.
–property *propertyname=value*	This is used to specify the property to be changed. See the list of properties below that can be set with this utility.

Argument	Description
–lockAccount {*failedLogins*}	This is 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	This 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.

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 command causes an entire rule group to be evaluated. The second command evaluates 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 –version

mace –d *domainPath* –find *string*

mace –d *domainPath* –newGroup *groupName*

mace –d *domainPath* –newRule

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*

Argument	Description
–version	Use this argument to get the version information. It does not require –d domainPath.
–d *domainPath*	This specifies the domain in which to load the measure.
–find *string*	Use this argument to search all expressions for the specified string, and to print all of the rules and rule groups that have these expressions.
–newGroup *groupName*	Use this argument to create a new rule group with the specified name.
–newRule	Use this argument to create a new empty rule.
–addRule *groupName:ruleName*	Use this argument to add the specified rule to the specified rule group.

Argument	Description
–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 by using the –run argument.
–resolve *groupName*	Use this argument to order (does not evaluate) expressions within rule group. This 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. This 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.

Argument	Description
–print	Use this argument to print all of 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.
–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 a specified group and the non-shared rules within it.
–purgeRules	Use this argument to remove all rules that are 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.

Argument	Description
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

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 –version

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*}

Argument	Description
–version	Use this argument to get the version information. It does not require –d domainPath.
–d *domainPath*	This 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 that are 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.
–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.

Argument	Description
–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.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Managing the workbook batch queue – wbbatch

The **wbbatch** utility is used to manage workbooks in the **workbook batch queue**.

- Build workbooks that have been scheduled to be automatically built by using the “Auto-Workbook Build” wizard in the RPAS client. Workbooks in the build queue can be executed in parallel using the **–processes** parameter.
- Commit data from workbooks that have been scheduled for deferred commits by using the “Commit Later” option in the RPAS client (File menu).
- Refresh workbooks that have been scheduled to be refreshed by using the workbook’s default refresh rule group. Workbooks in the refresh queue can be executed in parallel using the **–processes** parameter

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*}

Argument	Description
–d *pathToDomain*	This specifies the domain that contains the workbooks.
–version	Use this argument to get the version information. It does not require –d domainPath.
–build {*workbookName*}	This builds workbooks that are 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 by using the specified number of parallel processes.
–commit {*workbookName*}	This 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.
–refresh {*workbookName*}	This utility refreshes workbooks that are scheduled to be refreshed. To refresh a single workbook in the queue, specify the name of the workbook. If no name is provided, all workbooks that are scheduled to be refreshed will be completed.

Argument	Description
–refresh –processes *number*	This refreshes workbooks in the auto-workbook refresh queue in parallel by using the specified number of parallel processes.
–scheduleRefresh	This schedules a workbook to be refreshed at a later time by adding it to the workbook batch queue.
–unscheduleRefresh	This removes a workbook from the workbook batch queue.
–scheduleCommit	This schedules a workbook to be committed at a later time by adding it to the workbook batch queue.
–unscheduleCommit	This removes a workbook from the workbook batch queue.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Workbook manager – wbmgr

Use the wbmgr utility to inspect or remove the existing workbooks. Do not assume that manual removal of the workbook directories will remove the workbook metadata in the domain.

Usage:

wbmgr –version

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*}

Argument	Description
–version	Use this argument to get the version information. It does not require –d domainPath.
–d *pathToDomain*	This specifies the domain that contains the workbooks.
–list –all	This lists all workbooks in the domain.
–list –user *userName*	This lists all workbooks that belong to the user.
–print –wbList *wb1,wb2,...*	This prints detailed information about workbooks in the list.
–remove –all	This removes all workbooks from the domain.
–remove –user *userName*	This removes all workbooks from the domain that belong to the specified user.
–remove –user *userName* –wbList *wb1,wb2*	This removes all of the workbooks in the specified list for the specified user.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Informational RPAS 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.

Checking the validity of a domain – checkDomain

This utility checks the validity of an existing domain. Currently, its primary purpose is to verify that a master domain matches its respective local domains. All discrepancies are written to the terminal.

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

Argument	Description
–d *pathToDomain*	The path to the domain that needs to be validated.
-type *expectedType*	The expected type of domain: **simple**, **master**, or **sub**.
-q	quiet mode – do not display progress messages.

Valid domain types:

- **simple**: domain is a standard, non-partitioned domain
- **master**: domain is the 'master' of a global domain set
- **sub**: local domain is one partition of a global domain set

When **checkDomain** is run on a **simple** domain (for example, non-Global Domain), the following two items get validated:

1. The domain directory exists.
2. It is of type "simple."

If **checkDomain** is run on a **Global Domain**, it verifies that the global domain exists, that it is of type "master," and that it checks all of the sub-domains for:

1. The sub-domain directory exists and is of type "sub."
2. If the master domain has a repos directory, the sub-domain also has a repos directory.
3. 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 –version

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

Argument	Description
–version	Use this argument to get the version information.
–db *dbName*	This specifies the database for which you want to list the contents.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Printing data from arrays – printArray

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

Usage:

printArray –version

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}

Argument	Description
–version	This prints the RPAS version, revision, and build information.
–array *db.array*	This specifies the array that you want to print. You must specify the full path to the database that contains the array. Do not specify the .gem suffix.
–specs	This prints the specifications of the array and the positions along each dimension.

Argument	Description
–cell	This identifies a single cell. It must not contain spaces. Specify it using the format “dim1:pos1,dim2:pos2,…”
–slice	This identifies a single 1-D slice. It must not contain spaces. Specify it using the format “dim1:pos1,dim2:pos2,…”
–allpopulatedcells	Prints all populated cells including the navalue of the array.
–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	This is used in conjunction with –specs. It only prints the first *n* positions for each dimension.
–cellsperrow	This specifies how many cells should be printed in each row.
–noposnames	This suppresses the output of position names.

Printing data from measures – printMeasure

Use the printMeasure utility to print measure information.

Usage:

printMeasure –version

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

printMeasure –list

Argument	Description
–version	Use this argument to get the version information. It does not require –d domainPath.
–d *pathToDomain*	This specifies the domain that contains the measure that you want to print.
–m *measure*	This specifies the measure that you want to print.
–wb *workbookName*	This specifies the workbook that is associated with the measure that 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*	This prints out the nobs, nods, and sparse format of the measure array at the specified intersection and agg type.
–loglevel *level*	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Retrieving domain information – domaininfo

The **domaininfo** utility provides basic information about a given domain. Specifically, it can identify whether a domain is a simple domain (not created in global domain environment), a master domain (parent of a global domain), or a local domain (child of global domain).

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

The **-domainversion** command gives the version of the domain pointed to by the **-d** parameter. The **-expectedversion** command prints the domain version that the current code expects to find.

The **-type** command returns which of three types of domain is being inspected:

- Simple – domain is a standard, non-partitioned domain
- Master – domain is the 'master' of a global domain set
- Sub – domain is one partition of a global domain set

The **-sparsity** command reports whether a domain has been designated as containing sparse or hypersparse measure databases.

If a Global domain is being inspected, the **-listsubdomains** and **-subdomain** commands may be useful. The **-listsubdomains** command will display a list of all subdomains of this domain, and will tell which positions of the partitioned dimension are in each subdomain. The **-subdomain** command allows searching for a particular position of the partitioned dimension or any dimension that rolls up to the partitioned dimension. For example, if the global domain environment is partitioned on DEPT, one may query which local domain contains a particular DEPT position or a particular SKU position.

The **-history** command lists the different versions and upgrade actions that are taken on the domain.

The **-all** command reports all relevant information on the domain.

Usage:

domaininfo –d *pathToDomain* [*commands listed below*]

domaininfo –expectedversion

domaininfo –version

Argument	Description
–domainversion	Display version of referenced domain
–type	Display domain type (see types above)
–xnames	
–sparsity	Display domain sparsity (see types above)
–listsubdomains	List all subdomains of a global domain
–subdomain *dim,pos*	Find subdomain containing given position
–history	Display version history of domain
–all	Display combined details of domain
–expectedversion	Display domain version expected by this code
–version	Display version of this utility

Index