Oracle Retail® Predictive Application Server
Release Notes
Release 11.0.4.18
September 2006

ORACLE

Customer Support

- https://metalink.oracle.com

When contacting Customer Support, please provide:

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

Release Notes

Current Patch – RPAS 11.0.4.18

Release Information

Application	Oracle Retail Predictive Application Server (Client, Server, and Configuration Tools)
Version Number	11.0.4.18
Code Cut Off Date	08-15-06
Release Date	09-07-06
Type of Release	Patch
Base	11.0.4
Patch	
Supported OS, Server	Sun Solaris 8, AIX 5.1/5.2/5.3, HP-UX 11i (11.11), Windows NT
Supported OS, Client	Windows NT
Required 3rd Party Software	JRE 1.4.2 (installed on client and server machines); 1.4.2_09 is recommended for IBM installations CVS (installed on server – if using integrated Source Control)
Related Documentation	RPAS 11.0 Installation Guide RPAS 11.0 Administrators Guide RPAS 11.0 Users Guide RPAS 11.0 Configuration Guide RPAS 11.0 Rule Functions Reference Guide RPAS 11.0 Calculation Engine Users Guide

Current Patch Details – Patch 18

Resolved Issue	Defect
Resolved an issue related to RDF's use of rollup attributes which was causing error messages, such as the one below, to display when building a workbook such as Forecast Maintenance. MeasureAttributeException: cannot set measure rollup because of invalid attribute [FBRT] and position [FBRTDEFAULT].	5079587
Resolved an issue related to the building of Alert Manager workbooks in RDF application. Hierarchy mappings were being mapped to the wrong hierarchy and errors, such as the one below, would appear upon workbook build. "PositionMapException: STR_001 is not a position of the item dimension"	5022218
Resolved an issue when using "Range by Alert" that caused this feature to be disabled when the user aggregates up or rolls down the product or location hierarchy in the workbook.	4884687

Patching Note

Each RPAS patch updates the domain version information and includes changes to the structure of the RPAS domain that require all workbooks to be rebuilt. Existing workbooks cannot be opened after a domain was upgraded.

Make sure that all data is committed for existing workbooks where desired before applying this patch. This includes workbooks that need to be committed manually and any workbooks waiting to be committed in the "commit later" queue.

Also note that each RPAS patch contains a new RPAS client that must be installed after upgrading a domain to the new version of the server.

Previous Patch Information

This document contains information about the latest patch of RPAS 11.0.4. This patch includes defects that have been fixed and/or other functional changes or additions that have been made to the RPAS client and server.

Each RPAS patch updates the domain version information and includes changes to the structure of the RPAS domain that require all workbooks to be rebuilt. Existing workbooks cannot be opened after a domain was upgraded.

Make sure that all data is committed for existing workbooks where desired before applying this patch. This includes workbooks that need to be committed manually and any workbooks waiting to be committed in the "commit later" queue.

Also note that each RPAS patch contains a new RPAS client that must be installed after upgrading a domain to the new version of the server.

Patch 17

Resolved Issue	Defect
The RPAS Client was updated so that an error does not occur when building a workbook from the Alert Manager.	4856450
The RPAS Server was updated so that the regmeasattr utility now has a –removepos option, which would remove a position on a given measure attribute.	4854794
The RPAS Client was updated so that when a workbook is built in the Alert Manager the dimension attribute information displays correctly.	4854879

Note:

Rerun the Alert Finder

Customers using the alert functionality in RPAS need to run the alert finder (alertmgr utility with the –findAlerts argument) after applying this patch. This is required because the RPAS client was fixed in this patch to be able to distinguish between the scenario when no alert hits are found and when the alert threshold (the maximum number of alerts) is reached.

Patch 16

Resolved Issues

Resolved Issue	Defect
An enhancement was made that allows for positions not mapped to alert hits to be filtered out in the wizard process when using the feature to range the hierarchy by alert. This feature was added only to 11.0.4 as this functionality already exists in RPAS 11.1 and later releases.	394987
The measure registration utility (regmeasure) was changed so that it outputs a return code of zero when it is successful. Previously, the utility was returning a value of one even when it was successful.	395753

Known Issues

Known Issue	Defect
RPAS does not currently allow workbook templates to be removed from a solution/configuration even though this is an option in the Configuration Tools. This functionality is currently available in version 11.1.	N/A

Note for Customers with HP Hardware

Customers that run RPAS on the HPUX operating system need to ensure that their system is updated to a specific HPUX operating system patch level to ensure compatibility with RPAS 11.0.4.16.

RPAS 11.0.4.16 is built using HP's native code compiler aCC for the HPUX UNIX operating system. Specifically what is required is the **aCC runtime patch version 3.61**, which is included with the **HPUX patch PHSS_32573**.

If necessary use the following command to determine the current version that the system is using: change directories into /usr/lib and run "what libCsup_v2.2." If the system does not return "A. 03.61" or a higher version it is necessary to update the HPUX operating system with patch PHSS_32573.

Previous Patch Information

The following section contains information about previous patches of RPAS 11.0.4. This includes defects that have been fixed and/or other functional changes or additions that have been made to the RPAS client and/or server.

A new section is provided for each RPAS patch.

Patch 15

Resolved Issues

Resolved Issue	Defect
The RPAS client was fixed to address an issue with saved styles. Specifically, the client now allows a new window to be added to a workbook built from a workbook template where the style was saved for that template. A common scenario for this condition is when a user adds an extra measure in the wizard process. If that measure is placed in a new window, this was previously a problem if the template for the given workbook has a saved style.	376843
The help files for the RPAS client were replaced with a new version, and they are now up to date.	393031, 392922
The Measure Analysis workbook template was fixed so that alert hits are now properly calculated and displayed in Measure Analysis workbooks. The hits are also updated when the workbook is refreshed.	393067
RPAS addressed an issue where alerts were being inserted on the wrong worksheets in alert workbooks.	393539
An issue was corrected in workbooks that contain position labels with parenthesis. Previously, workbooks with position names with labels were causing errors in certain workbooks.	393576
Several issues were addressed that were encountered when displaying attributes and attribute labels in the item selection wizard.	393804
The client was updated so that style formats are no longer causing errors after a domain was patched.	394150
The exportData utility was enhanced so that it now exits if one of the exported dimensions doesn't exist in one of the export arrays.	392969
A problem was fixed so that the count of alerts in alert workbooks is based on the number in the workbook and not the domain.	377080

Patch 14

Resolved Issues

Resolved Issue	Defect
The utility for loading measures (loadmeasure) was fixed so that data can be purged in configurations/solutions using different formats for the position names of day; the format is set by the dimension information. For example, now “DAY19970101” is now functional. Previously, it had to be in the format “D19970101.”	392960
It is now possible to add only scalar measures to a custom workbook using the Configuration Tools.	392858
A fix was made to the RPAS client that corrects a navigation issue when using the forward arrow keys (>>) when viewing data in the graph mode.	377864
An issue with rule groups and rule group transitions (for instance, switching from load to calc) was fixed. Under special conditions, users were seeing errors when refreshing measures that had just been committed in another workbook.	377604
The above fix (Defect 377604) also addresses the issue of calculation rules not working correctly in a workbook when they have been previously saved in another workbook.	377600
Measures no longer disappear from the wizard screens after being inserted into a workbook (the same or different). Previously, if a user selected an extra measure into a workbook, that measure would not be available for insertion in other workbooks without logging back in to RPAS.	377325
The problem in the Hierarchy Maintenance template was fixed to correctly handle using the back button to go back and choose a different dimension.	377186
Additional measures can now be included in workbooks scheduled to be built automatically.	377079
An issue was fixed for domains with registered alerts. Previously, alert workbooks were building with errors after upgrading from a previous version and inserting new alerts.	376950
An issue was addressed where alert workbooks were not building in new domains with registered alerts but when the alert finder had not been run.	376238
When setting up a measure analysis workbook in auto workbook build, the issue of blank measures appearing after the Available Positions window was corrected.	376112

Resolved Issue	Defect
The exportData utility no longer fails when both the –wide and –skipNA options are used together.	375889
A complex issue with user-defined hierarchies was addressed that was occurring when new positions are added to a domain. Note there are a number of complex issues that require the administrator/ implementer to perform some manual steps to avoid running into problems.	375837
The utility run after adding positions to a domain (reshapeArrays) was updated so that it does not leave temporary lock files (with the ".lck" extension) in the root directory of the domain after the utility is executed.	375333

Patch 12

Functionality Additions and Changes

Filtering out Values when Exporting Data

An enhancement was made to the RPAS utility **exportdata** (new parameter **–precision**) that allows a user to ignore (not export) values that differ from the NA value less than a specified value.

The purpose of this enhancement is for the export utility not to export values that vary only slightly from the NA value, most commonly very small values close to zero. For example, consider a numeric measure with a NA value of zero; if the precision value is specified as 0.01 any number smaller than that is ignored, so 0.0034 would be ignored while 0.34 would be exported.

The precision value must be less than one. If a value greater than one is provided the utility returns a warning.

Return Codes for Calculation Engine Utility – mace

The calculation engine utility **mace** now returns a code of zero after successfully executing the following scenarios/actions:

- running a rule group (-run -group)
- adding a rule (-addRule)
- running rule groups in a workbook (-transit)
- validating rule groups (-validate)

RPAS Lock Timeout Variable

RPAS has introduced a new environmental variable named “RPAS_LOCK_TIMEOUT.” This variable was added to allow an administrator to set how long they want to wait before timing out when there are lock contention issues. This most commonly occurs when two users attempt to simultaneously commit/save the same data.

This environment variable is set on the back end to the number of milliseconds to wait for a file lock before returning a lock contention error. The default value is 60000 milliseconds (one minute). 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
DomainDaemon -port 55123 -start -debug &
```

Resolved Issues

Resolved Issue	Defect
A problem with alert measures was fixed so that they function correctly when alert measures are selected through an "extra measures" wizard page.	376689
The insert measure functionality was fixed to work correctly with the RDF workbook “Forecast Approval.”	376647
The calculation engine utility (mace) was fixed to return a zero return code upon successful completion.	376454
The standard RPAS administrative utility for adding a user was fixed to function properly in multilingual domains.	376115
Switching from Daylight Savings to Non Daylight Savings Time in the Central Time Zone was fixed to adjust the date/time correctly.	376025
Workbook formatting was fixed so that if the user saves the workbook format. Additional measures can now be reselected in a new workbook build wizard process.	375759
Measures that are standard within a workbook are now no longer available for insertion via the wizard that builds the workbook.	375501
A “-precision” argument was added to the exportData utility. This argument may be used to ignore exporting numeric values, which only differ from the navalue less than the precision value specified. The precision value is designed to be a number less than 1. If the precision value is greater than 1, the user will see a warning message.	375433

Resolved Issue	Defect
The Alert Manager was fixed to allow the Forecast Approval workbook to build alert workbooks correctly when hiding dimensions during the workbook build process.	374788
A new environment variable, "RPAS_LOCK_TIMEOUT" was created to allow users to set the timeout period length when writelock errors occur during simultaneous commits.	374102
The refresh functionality within the Measure Analysis workbook was corrected to accurately handle newly added measures.	376601
The Hierarchy Maintenance template was fixed to work correctly when assigning/reassigning items.	376763
Domain utilities were corrected to no longer copy alitemp files to the root folders of domains.	376659

Known Issues

Known Issue	Defect
User defined dimensions with no associated positions should have a dummy position available, but in certain circumstances, another position had been appearing with the same name. When users rolled up to that dimension, they would receive an error. The RPAS workaround is that the user has to use Acumate to open the hmaint database and type a specific command to delete the position: update dim_XXXX (hdr:posdel dim:XXXX) with XXXUNASSIGNED; replace XXX with the name, not label, of the dimension. use “print dim_XXXX;” to see if there is a duplicate position or not.	375838

Patch 11

Functionality Additions and Changes

RPAS has implemented a feature that provides the ability to export date values in a format other than the name of the position. Many customers want to see dates exported as a date value (such as week ending “20040105”) rather than the position name (such as “W01_2004”).

This is now possible with the availability of a function **indextoenddate** that returns the index of a given period. When combined with the previously available function **indexfirst/last**, the function can be used to create a mapping measure that can be referenced when exporting data.

To meet this requirement, the following steps must be followed:

- Create a mapping date measure (for instance, “mapmeas”) that has a base intersection of the CLND dimension that is being mapped (the week dimension in the above example)
- Create and evaluate an expression for that measure using the **indextoenddate** and **indexfirst** functions
    1. For example: mapmeas=indextoenddate(indexfirst([clnd].[week]))
    2. The expression can be evaluated directly by using the **mace** calculation engine utility or in a workbook
- Run the **exportdata** utility using the array mapping technique, referring to the underlying array name of the mapping measure (see the RPAS 11.1 Administrator’s Guide for additional information on exportdata)

To clarify this approach, a simple example follows.

Exporting Date Values as End of Week

Consider an example where the data to be exported is stored at sku-store-week (for instance, W01_2004 sku1 str1). Imagine that the user desires to see the value of week as YYYYMMDD instead of the position name.

First, it is necessary to create a date measure that will store the mapping data between the position names and the date of the last day in the week. For example, register the following measure:

```
regmeasure –d . –add mapmeas – type date – baseint week – navalue
2000020313200000 –db testdb
```

Next the values of the mapping measure needs to be updated with the correct date values. This can by executing an expression in the mace utility using the **indextoenddate** and **indexfirst** functions (or combinations thereof depending on how the dates should be mapped). For example:

```
mace –d . –run –expression
‘mapmeas=indextoenddate(indexfirst([clnd].[week]))’
```

After the mapping measure/array was created and values have been populated, run the **exportdata** utility such as the following:

```
exportData -d . -out test.out -array "data/pos|pos%1 %13f 0 %13f" -dim "week
testdb|mapmeas%1 %8s 1"
```

In this example the date values would be formatted as YYYYMMDD.

indextoenddate

Returns the end date of the period whose index number is supplied.

Syntax: **indextoenddate**(<index>[**,**[<clndhierarchy>].{[<dimension>] | **current**}])

Where *<clndhierarchy>* is the name of the calendar (time) hierarchy, and *<dimension>* is the name of a dimension in the calendar hierarchy. **current** is a keyword that implies the current dimension in the calendar hierarchy. If *<dimension>* is not a valid dimension in the calendar hierarchy, an ***error*** is generated. If the calendar hierarchy and dimension are not supplied, the default is the current calendar dimension.

> **Note:** This function requires that the day dimension of the calendar hierarchy be included in the workbook. If the lowest dimension of the calendar hierarchy is above the day dimension, the function will not be able to return a valid date.

<index> is an expression that returns an index number in the indicated calendar dimension. If *<index>* is non-integer, only the integer portion is used. If *<index>* is not a valid index number for the specified dimension, an ***error*** is generated. If the measure being evaluated does not have a base intersection in the calendar hierarchy, and the **current** option is used, an ***error*** is generated.

The function returns a date that is the end date of the period indicated by the dimension and index number. If the period being evaluated is at or below the day level, the end date is the date of the whole of the period. If the period being evaluated is above the day level, the end date is the date of the last child position at the day level of the period being evaluated.

Inverse: The ***indextoenddate*** function does not have an inverse.

Examples:

- **indextoenddate**(**current**)
  Returns the end date of the current time period
- **indextoenddate** (**indexfirst**([clnd].[qtr]))
  Returns the end date of the last period in the current time dimension in the current quarter.
- **indextoenddate** (**index**([clnd].[week], openweek), [clnd].[week])
  Returns the end date of the period at the week level whose name is held in the openweek measure.

Resolved Issues

Resolved Issue	Defect
The exportData utility, which exports data from an existing domain, is now exporting all records when specifying a range.	374710
The wbbatch utility now provides a return code of zero (0) in the warning message that occurs when a user runs wbbatch and has no workbooks scheduled to be committed.	374687
The help files for the RPAS client can now be correctly accessed from the Windows Start Menu.	374537
All workbook templates now accept the insertion of all upper and lower bound measures.	374207
The exportData utility was modified to export calendar data to display in the format of a week ending date: 20040101 (to be in sync with the date format in the RMS product) when used in conjunction with a new function indextoenddate (see above section in release notes).	373218
The wbbatch utility now provides a return code of non-zero when an error is encountered.	374690
Certain changes were made to the calculation engine so that calculation errors no longer occur on saved, then re-opened workbooks.	374694
A fix was made to AutoWorkbook Build. This defect is resolved. There is a new outstanding issue related to autoworkbook build that is specific to building an RDF Forecast Approval workbook in batch mode. See Defect 375637 in the “Known Issues” section of this document for details.	374637
An Extra Measures wizard page now only displays measures with the 'insertable' property set to true.	374531
All token measures are now registered as not insertable. These measures will no longer appear in an Extra Measures wizard page.	374804
The RPAS Client was fixed so that saved workbooks aren't displayed in duplicate in the "open" window. This situation was occurring under certain rare circumstances.	374741
The Alert Manager was fixed to allow the Forecast Approval Workbook within the RDF application to correctly trigger alerts when its "ignore" feature is used.	374267

Known Issues

Resolved Issue	Defect
The Security Administration Template (including Measure Rights worksheet) does not function correctly after a domain was upgraded if a style was saved for the template. The workaround is to delete the database that stores the styles (rpas_sec*.gem in the domain/styles directory).	374359
When using auto-workbook build to build an RDF Forecast Approval workbook, the user is able to put an initial value in the "# of time periods" field, but when wbbatch is executed an exception is returned due to there being no stored value being set for these fields. These fields are used to set history, forecast horizon and post-horizon time periods to include in the workbook.	375637

Patch 10

Resolved Issues

Resolved Issue	Defect
Under certain circumstances, if a user clicked "close" and "ignore changes" when shutting down a workbook, the user received an error stating "unexpected exception occurred." The issue was corrected in RPAS for the 11.0.4.10 patch.	373255
Creating a new user group and adding users to the new group resulted in an error message under certain circumstances when position level security was set. This error no longer occurs.	373676
RPAS now properly handles measures designated as read-only and read-write in the Security Administration workbook templates. Previously some measures that were set to be "read-only" could be edited and committed.	373339
Performance issues with the use of the edit/fill all commands have been improved in the RPAS client.	373748
If the user encounters an exception and the DomainDaemon does not respond, RPAS now returns non-zero values for all commands, including ping.	373836
Users will see the following error message when attempting to log into a domain that was designated as inactive (using the "deactivate" parameter with the domainDaemon): "The Domain is Temporarily Unavailable."	373802
The RegTokenMeasure utility can now be used with relative directory paths within the command line. Previously, the utility was failing unless a full directory path was provided.	374192

Resolved Issue	Defect
A performance enhancement was made to the calculation engine when running a single expression using the “mace” utility. Specifically, the calculation engine no longer creates temporary rules and rule groups when executing a single expression.	373682
A performance enhancement was made to the alert manager utility (alertmgr) that results in faster execution when finding alerts.	373692
The standard administrative workbook template Measure Analysis now provides the ability to refresh data from the domain.	374099

Patch 9

Notes about this Patch

- The behavior of position level security was changed in 11.0.4.8, but was not explicitly called out in the release notes (but was noted as a resolved issue). This new behavior will impact existing security settings where position level security is enabled. The impact of the change is that previously access granted at ANY level (user, group, or world) gave a user access to a position. In 11.0.4.8, a user must be given access to a position at ALL levels.
- If an existing domain (pre-11.0.4.9) contains one or more hypersparse measure databases all databases will be converted to hypersparse during the upgrade to 11.0.4.9 (using the upgradeDomain utility). The domain will then be marked as hypersparse, and future runs of the loadmeasure utility will create new measure databases in the appropriate mode.

Resolved Issues

Resolved Issue	Defect
Position labels in hierarchy files (“hier.dat” file name) can now be specified with parentheses. Note that the parentheses must be matching.	371292
RPAS now allows the creation of workbooks that contain only scalar measures (that is, no dimensions in the base intersection).	371434
RPAS has fixed a specific problem with load rule groups and now allows expressions to be loaded to rules that were previously empty.	371635
The alert finder (alertmgr utility with the –findalerts parameter) defaults to displaying 5000 hits during a given run of the utility. A new parameter is available “-navigationThreshold” that allows the specification of a larger maximum number of hits to display.	371719

Resolved Issue	Defect
The utility for exporting data (exportdata) was previously failing. The related issues have been addressed and the utility works as expected.	372166
Several issues with the names of aggregation types and spread methods were addressed. RPAS now correctly handles the registration of certain agg/spread combinations that previously was causing problems. The specific case that was reported involved the aggregation type of “ambig” and the corresponding spread method of “replicate.”	372484
RPAS now properly validates that the hierarchies and dimensions in the base intersection of measures is included in the workbook before trying to insert a measure. If the base intersection of the workbook is above the base intersection of the measure then the measure cannot be inserted. If a dimension of a measure is not explicitly or implicitly included in the workbook it can also not be inserted.	372523
The utility for loading measures (loadmeasure) was updated to return exit codes other than zero when it fails. The loadmeasure utility was also updated to check for a non-existent directory for the “-logdirectory” parameter.	372713
If an existing domain (pre-11.0.4.9) contains one or more hypersparse measure databases all databases will be converted to hypersparse during the upgrade to 11.0.4.9 (using the upgradeDomain utility). The domain will then be marked as hypersparse, and future runs of the loadmeasure utility will create new measure databases in the appropriate mode.	372727
The measure property “insertable” is now respected inside workbooks. Measures with this property are available for insertion into a workbook if the user has access to do so.	372754
The “reshapeArrays” utility now returns an exit code of 1 if an error is encountered.	372846
RPAS now correctly handles the checking of lower and upper bounds for measures with a data type of “real.”	373169

Known Issues

The following issue exists in RPAS 11.0.4.9.

Known Issue	Defect
For RPAS 11.0.4 NT the letter of the hard drive must always be consistently specified in either upper case or lower case both when sending add/remove/disable commands to the DomainDaemon and also when setting up the client-side configuration file.	369636

Patch 8

Notes about this Patch

- The RPAS on-line help is included with the client package for 11.0.4.8. Certain patches of 11.0.4.x did not have this help file due to a technical issue.
- RPAS provides the ability to deploy the RPAS client over an intranet. The code required to do this is referred to as the "web installation files" and is packaged separately from the RPAS client. This package was included with 11.0.4.8. Administrators should reference the RPAS 11.0.4 Installation Guide to understand this deployment option and how it is configured and used.

Resolved Issues

Resolved Issue	Defect
Position queries are now properly updated after executing a custom menu option. This occurs only if position queries have been defined. Previously the window was not getting properly refreshed under these circumstances.	367600
After inserting measures into a workbook and saving the workbook format at the template level those measures are now included with future builds of that workbook template.	367926
Values of attribute measures are now properly updated after using cut/copy/paste special operations in the RPAS client.	369270
The standard RPAS Hierarchy Maintenance administrative workbook template now performs quickly with large numbers of positions (test case 50,000 SKUs). Previously, the wizard and workbook build processes took excessive time to run.	369926
RPAS has addressed a rounding issue with real numbers. Under certain circumstances real numbers were previously being rounded to integers.	370272
A security issue was fixed that prevents users from scheduling the build of a workbook template in Auto-Workbook Build. Previously, users could see and use workbooks to which they did not have access.	370596

Resolved Issue	Defect
RPAS has fixed an issue with dynamic hierarchies when switching to outline view. Previously, this was causing the RPAS client to suddenly exit.	370675
A customer-specific issue with User Security was fixed.	370929
An issue with multi-lingual labels was addressed within RPAS.	370983
RPAS has fixed a problem that occurred when two users simultaneously delete workbooks.	371061
The RPAS on-line help file is now included with the package of the RPAS client.	371097
The “-version” parameter of the upgradeDomain utility now correctly provides the version information. Previously, calling this parameter only provided the usage of the utility.	371246
RPAS has fixed a problem that occurred when the decimal separator is set to comma in the Windows regional settings. Previously, values for formatting exceptions and scale factors did not hold.	371354
RPAS has implemented some validation and mechanics that prevent a domain from getting into an invalid state, which can happen after hitting “Control-C” on the command line when running a back-end or batch process. Transactions now complete before exiting and RPAS properly handles the update to certain pieces of meta-data.	371439
The RPAS client is now properly refreshing data after a sort was set in the Show/Hide dialog window.	371442
RPAS has fixed an issue with position level security that was not properly restricting users when they were denied access to certain positions.	371479

Known Issues

The following issue exists in RPAS 11.0.4.8.

Known Issue	Defect
The hit count for alerts is not properly calculated and displayed in the RPAS client.	371719

Patch 7

Resolved Issues

Resolved Issue	Defect
Proportional spread type now works correctly with average aggregation method. These fixes apply to the populated versions of this agg method and spread type as well.	366448
RPAS client was fixed so that it doesn't crash when RPAS encounters the same name for a hierarchy and a dimension (which is illegal).	366882
Problems with exporting date type data using RPAS utility exportdata were addressed.	367150
RPAS has addressed an issue encountered with Forecasting that gave an "Illegal Position Name" error in the RPAS administrative workbook templates Security Administration and Translation Administration.	367160
The "Find" feature no longer causes the RPAS client to lock up. This was happening in certain wizard dialogs.	367188
An issue with saving formatting for a workbook template that caused a "Connection Closed" error was fixed.	367539
The Alert Manager no longer errors when running the Alert Finder when no alerts have been registered, but rather informs the user as such.	368423
A specific problem refreshing certain types of recalc rules was addressed.	369260
A specific problem with the calculation of gross margin in a Planning solution was addressed.	369317
The DomainDaemon can no longer be used to stop a process ID (using the "stopServer" parameter) using a port number that is not valid for that ID.	369635
RPAS allows the format "YYYYmmddHHMMSSsss" when defining the na value for date type measures (using the RPAS regmeasure utility).	369748

Patch 6

Resolved Issues

Resolved Issue	Defect
Japanese characters are now properly displayed in the RPAS charting tool.	N/A
Certain RPAS functions (for instance, time series) were not properly functioning when only a single time position was selected. This was fixed for all such functions.	363199
RPAS now supports hypersparse workbook databases in hypersparse domains. Previously, hypersparse support was limited to the domain.	363203
RPAS now correctly handles invalid hierarchy updates involving multiple/invalid parents for positions. Previously, under certain scenarios, the invalid definitions were being loaded.	363227
Position-level security is now correctly handled when an administrator is defining a workbook for auto-workbook build in the standard RPAS template for another user. The administrator now assumes the role of the specified user so that the proper selections can be made.	363371
Workbook windows now correctly display all measures when there are no visible measures selected in the code. This defect was identified by Retek development.	363396
RPAS has fixed a sorting problem with the calendar hierarchy after staging.	364849
Display-only recalc measures are now properly sorted.	365376
RPAS now correctly retains the selected positions in the 2-tree wizard after aggregating the list of available positions to a higher level and removing the lower dimensions.	365507
In the RPAS Security Administration workbook, position level security windows now display the settings above the user level correctly. RPAS displays a checkbox at the group level when all users are selected, “?” when some of the users are selected, and blank when none are.	365585
Hierarchies are now correctly synchronized in the 2-tree wizards after “select rollup” and “select dimension.” Previously, the selected and available positions were losing synchronization after performing these two operations.	365633
Related to sorting defect 365376; workbook was becoming corrupted after sort errors. This issue is no longer valid after fixing the other sorting defect.	365868

Resolved Issue	Defect
Problem with copying and pasting values of zero were fixed (also with copy special and paste special). Under certain conditions, values of zero were not being pasted.	366173
Attribute labels are now properly refreshed after adding attributes to the 2-tree wizard.	366207

Functionality Additions and Changes

The following features are newly available or have been changed in RPAS 11.0.4.6:

- An Application Programming Interface was added that allows the descriptions of dimensions to be programmatically changed (feature only available for developers).
- In a multi-lingual domain, RPAS now displays the English label of a string when the string is not translated into the selected language. Previously, the string ID was displayed instead of the English string.

Patch 4

Resolved Issues

Resolved Issue	Defect
An issue with multi-lingual domains was fixed so that the operation “Select Rollup” in the wizard now correctly displays the labels for dimensions.	362341
The correct error message is now displayed for the underlying issue with this defect, which was that there was a rule without an expression.	362064
Friendlier error messages have been added to better describe problems with incompatible base intersections. The old message was "Failed to validate parse tree." The new messages are the following: • “Intersections don't conform due to a branched hierarchy. The incompatible intersections are …” • “Intersections don't conform due to different hierarchies. The incompatible intersections are …”	361334
A problem with the RPAS function “tssum” was fixed and the function is now working properly at the “all” calendar level for display-only measures.	362531
There is no longer a limit on the number of dimension attributes that can be displayed in the wizard. Previously, the horizontal scroll bar in the window could not be used when more than a few attributes were associated with a dimension.	362527
Workbook template security can now be set to three states: denied, read-only, full access. Previously it was only possible to define grant or deny (true or false) access.	361659
“Select rollup” is now consistent between wizards and workbooks. Previously, they were ordered differently.	362593
Measure rights window of Security Administration now correctly displays the Parent values. This means that the value of a Parent (checkbox since it is a Boolean measure) is now set to True when all of the children have a value of True.	361025
Previously there was a restriction with the utility for loading hierarchies (loadHier) that did not allow “soft” directory paths to be used ("soft" file paths are effectively pointers to aliases of other directories). This was fixed and it is now possible to run this utility outside of a domain.	362037

Functionality Additions and Changes

The following features are newly available in RPAS 11.0.4.4. Some of these items also related to defects that are included in the "Resolved Issues" section above.

- Read-only workbooks – Workbook template security now allows workbooks to be built as read-only for a given user. A user can now have access set as:
  1. Granted – full access depending on measure security defined in template
  2. Denied – no access to workbook template
  3. Read-only – user can build template but cannot edit any measures
- Multiple RPAS log files – RPAS now keeps the previous version of the RPAS log file

Patch 3

Resolved Issues

Resolved Issue	Defect
Alert manager was displaying different results when building a workbook through the alert manager rather than building the workbook directly with the workbook template. Now when an alert is refused for RDF, the comments field is populated.	359743
Workbooks saved with "World" access can now be committed without error by any user. Previously, there was an error when a user other than the workbook creator attempted to commit data in workbooks with World access.	360341
Calculating display-only measures at aggregate levels is now working correctly.	361041
Problems that impacted load testing have been corrected. 128-user test now completes successfully.	360158
Enhancement for a utility to perform some data mapping operations that were available in previous (9.x) RPAS versions (see below).	356562
RPAS client now applies the correct numerical format for decimals based on the Windows Regional Settings. **This was fixed in 11.0.4.2 but was not included in the release notes.**	356445

Functionality Additions and Changes

New utility for mapping data between domains – mapData

Usage:

mapData -srcDomain srcDomainPath -destDomain destDomainPath

mapData -srcDomain srcDomainPath -destDomain destDomainPath -db dbName

mapData -srcDomain src DomainPath -destDomain destDomainPath -db dbName -array arrayName

-srcDomain srcDomainPath --> full path to source domain

-destDomain destDomainPath --> full path to destination domain

-db dbName --> apply mapData only on the given database

-array arrayName --> apply mapData only on the given array. Requires -db option.

Run mapData without -db and -array arguments to map the entire domain.

To set the logger verbosity level, use -loglevel:

Possible values: all, profile, debug, information, warning, error, or none.

Patch 2

Resolved Issues

Resolved Issue	Defect
RPAS now correctly represents recalc measures at intersections where it cannot be calculated. Previously, this was failing with an error.	360308
Administrators will now receive an error message when attempting to delete a user that is currently in use. This fix was addressed for all platforms.	357101
The sort/display attribute in the wizard process now correctly displays attributes that have been changed (for instance, changing a measure label after the measure was registered).	359508
Measure rights for non-administrative users are now properly respected in workbooks based on the measure security settings (based on settings made when the measure is registered or in a workbook template). Note that this access is NOT reflected in the RPAS Security Administration template for measure rights. Previously all measures were forcing read-only access for non-administrative users.	359724
Customer-specific problem where workbooks could not be built with all “classes” was fixed.	360363
Performance issue with the RPAS client from 11.0.4.1 (patch 1 of 11.0.4) was fixed.	360805

Resolved Issue	Defect
Problem with running a forecasting batch operation from the RPAS client on an AIX server has now been fixed and exceptions are correctly caught.	360029

Functionality Additions and Changes

Utility for Changing the Sparseness of Databases - changeDomainSpareseness

The following utility can be used for changing the database type from sparse to hypersparse for all databases in a domain.

General usage of the utility is:

changeDomainSparseness -d domainPath [-tohypersparse | -tosparse] {-nobackup}

changeDomainSparseness -version

To make a domain hypersparse you run:

changeDomainSparseness -d /path/to/domain -tohypersparse

If the -nobackup switch is specified, the old, sparse files will be deleted after a hypersparse copy is made. Otherwise, a directory named data/backup in the domain will be created and the old files will be put there.

Patch 1

Resolved Issues

Resolved Issue	Defect
Workbooks built with all elapsed time periods will now be properly protected. Before this fix users could edit time periods that should be elapsed, and thus protected, when building a workbook with time periods that were all elapsed.	359581
The feature "hiermods" can now take a 2-dimensional rollup specification for a given hierarchy. Before this fix, RPAS required that three or more dimensions be specified using hiermods. This feature is used to limit the number of dimensions of a particular hierarchy that are included and displayed in a workbook.	359270
The "not" mathematical operator (!) can now be correctly used in calculations (for instance, measure x != measure y).	359607
A performance enhancement is available that will improve the build times of workbooks. The improvements resulted in substantial reductions in the time required for loading data and the general overhead of a workbook build. Overhead is defined as the overall workbook build time minus load and calculation times.	Not related to defect

Release Information Overview

Functional Summary

The purpose of this document is to communicate the enhancements and changes that were implemented in RPAS 11.0.4. The features and changes described herein are available in the base platform and most are supported in the RPAS 11.0.4 Configuration Tools. Certain features might require some configuration or administration outside the RPAS user interface and outside of the Configuration Tools.

Due to some additions to the internal metadata definitions, an upgrade process is required to upgrade 11.0.3 domains to 11.0.4. Note that configuration changes will be required to take advantage of many of these new features (such as position queries in windows, non-materialized measures, new functions, procedures, and aggregation types), but that in general the performance enhancements will not require configuration changes.

The features and changes in RPAS 11.0.4 were implemented to support a number of customer requirements and Retek initiatives, including:

- Features to support the port of Retek Demand Forecasting to RPAS 11
- Features and performance enhancements to support Assortment Planning
- Features and performance enhancements to support the development of Retek Advanced Inventory Planning

Performance Summary

Significant performance enhancements were made in RPAS 11.0.4. Improvements made include faster workbook builds and many enhancements to the performance of the RPAS calculation engine.

Technical Summary

The RPAS platform consists of the RPAS server application and client user interface. The RPAS server was tested and verified on three Unix platforms (Sun Solaris, IBM AIX, and HP-UX) and Windows NT. See the RPAS Installation Guide for additional technical information.

Summary of Changes Made Since 11.0.3.1 Release

New Features

- Position queries in workbook windows – position queries provide the ability to define, save, and reuse the definition of a ‘query’ that results in a selection of positions in a hierarchy.
  1. Implementation of infrastructure to support position queries.
  2. Support for position queries in workbook windows for automatic ranging in the z-axis. This allows the user to change the position in the z-axis (or the slice) and RPAS automatically executes the “query” such that the set of positions that are relevant for another dimension for that slice are displayed in the client. Thus, for example, the warehouses that are visible in the down orientation will vary according to the current product in the slice area.
  3. Positions that are not selected through the query will be treated as hidden in the UI. The behavior of ‘hidden’ positions in the UI has not changed.
  4. The definition of the positions that should, or should not, be visible in a given context (that is, the slice position in another hierarchy) will be held in a normal binary measure. RPAS intends to support more sophisticated queries over time.
  5. Position queries are not available in the measures dimension.
- Sanity check – a mechanism that allows dynamic lower and upper limits to be placed on measures such that values are validated on edit prior to being passed into the calculation engine. The limits are held in measures that may have base intersections that are higher up the hierarchies with the values being ‘replicated down’ if required. Thus, for example, an upper limit at department may be used to perform a sanity check on, say, an IMU at class/week.
- Non-materialized measures, display-only – non-materialized measures are measures that are temporary or transient in nature, and are not persisted to the domain or workbook. This first implementation is for recalc measures that are only displayed in the client, although RPAS intends to support more sophisticated queries over time. These recalc measures (typically variances, and so on) are only calculated on demand (that is, when required to display on the UI), not during a normal calculation cycle, which for workbooks with many such measures provides a significant performance boost.
- Mechanism for kicking users off the system – the “domainDaemon” utility was upgraded so that users are kicked off the system when domains are deactivated (existing parameter). This parameter can be set and executed prior to beginning batch processes to remove users that are logged into domains.
- Changing descriptions of positions on the UI – a feature to allow the descriptions of positions to be changed through the normal calculation of rule expressions. This affects the descriptions in just the local workbook, not the domain. This is especially valuable in applications, such as ‘grading’, that require position descriptions to vary according to the context, such as the active class/assortment period.

- Aggregate procedure –This procedure allows a recalc measure to be calculated at aggregated levels by aggregation. The aggregation type can vary by hierarchy. Therefore, the procedure supports the calculation of measures where the required aggregations may be. For example, total up calendar and product, but average up location.
- flookup function – A function (not a procedure) that may therefore be used in complex expressions with other functions. A limited version of the lookup procedure, that looks up against specified slices (positions). The specification of the positions must be explicit in the function parameters (that is, it cannot be indirect through a measure).
- Several enhancements for functions with multiple results. See a later section in this document for additional details.
- New aggregation method for determining the median value (and median of populated values).
- Tokenized measures – string measures used as variables in an RPAS expression whose values refer to other measures. This functionality is provided to support the generation of configurations of applications such as RDF, and it is not supported as general configurable functionality through the tools for planning applications.
- Ability to set an environment variable (TODAY_RETEK) to a specified date that will override the system date when the 'today' keyword is used in expressions. This enables batch jobs to be run with an effective date different to the actual date, for example when updating actuals an hour or two before midnight.
- Ability to delete rules and rule groups using RPAS-Tools interface and calc engine utility (mace) – delete rules that do not belong to any rule group, delete specific rule groups and its rules, and delete all rules and rule groups.
- Function that returns the na value of an expression (function navalue).

Updated RPAS Utilities

The following utilities were changed in RPAS 11.0.4. Some of these utilities might be used in external, implementation-specific batch scripts. These scripts might require an update depending on the utility used and if the parameters have changed.

- Updated utility for loading measures (loadmeasure) – utility rewritten in C++ (all MSPL removed), requiring command line arguments to change
- Updated utility for registering measures (regmeasure)– utility updated to allow the update/modification of the position information for a measure along each dimension in the measure hierarchy
- New parameters for calculation engine utility (mace)
  1. Validate rule groups – the new parameter (–validate) takes one of three possible values (general, calc, refresh) to specify which type of rule group to validate
  2. Purge rules and rule groups – it is now possible to delete rules not referenced in any rule group, delete rule groups and its rules (that are not used in other rule groups), and delete all rules and rule groups
- Updated utility for loading hierarchies (loadHier) – functionally equivalent to hierarchy load utility in 11.0.3 but developed in C++ (all MSPL removed); changes to command line options

- Updated utilities for performing certain administrative tasks on hierarchies (refreshHier & checkParents) – rewritten in C++ (all MSPL removed); these utilities are not normally called by an administrator

New RPAS Utilities

The following is a list of new utilities in 11.0.4.

- Adding and removing users in a domain; this utility can be run from a command line or in batch (utility usermgr)
- Listing and removing workbooks; this utility can be run from a command line or in batch (utility wbmgr)
- Creating databases and indicating if they are in normal/spare or hypersparse mode (utility createDb)
- Determine the version of RPAS for a specified domain (utility domainversion)
- Determine the version of RPAS running in a specified location (utility rpasversion)

Other Changes and Additions

- Changes and additions to data in RPAS that can be translated
  1. Domains can be designated as single or multi-language domains; this is set during the domain-build process (use –m flag when running createRpasDomain or a flag set in the Configuration Tools)
  2. The position name and label of any dimension can now be translated
  3. Measure descriptions can be translated – previously only measure names and labels could be translated
  4. The format of measure names for loading translated data was changed; new format is R_{dimname}{property} (for example, R_MEASLABEL)
- Domain daemon was enhanced to allow domains to be accessed from multiple servers
- Ability to print on legal paper

Definition of New Features

The following section describes some of the larger new RPAS features in more detail.

Non-Materialized Measures – display-only

Definition

Non materialized measures are measures that are temporary or transient in nature, and are not persisted to the domain or workbook. In the initial implementations, they are always assumed to be of recalc aggregation type. They are designed to be performance-enhancing for two major reasons:

1. Non materialized measures are calculated only when required.
2. Non materialized measures are not persisted to the domain or workbook.

Expressions for normal (materialized) measures are always calculated when the rule is affected and the expression is selected. This is not the case with non-materialized measures. A non-materialized measure that is not used on the rhs of any expressions does not get calculated during the calculation cycle even when measures on the rhs of the expression to calculate the non-materialized measure are changed. Such measures only get calculated when they are required. Processes that may require such measures include viewing data through windows in workbooks, extracting data, and so on.

The materialized status of a measure is a measure attribute, which is maintainable through the tools. Validation ensures that all measures flagged as non-materialized have a recalc aggregation type. Any rule group that does not have a rule with a single expression to evaluate every non-materialized measure is invalid. Any rule group that includes any expression with a non-materialized measure on the right hand side is invalid.

Aggregate

Procedure that returns the value of a measure aggregated from the base intersection to the current level, using the supplied aggregation type.

Syntax

aggregate (<cachemeasure>, <hierspec1> [, <hierspec2> … , <hierspecn>])

Where <hierspec1-n> is [<hierarchy>].<aggtype>

The aggregate procedure provides a mechanism that is the functional equivalent of an aggregation type that varies by hierarchy. Such variable aggregation types cannot be implemented as aggregation types because of the difficulties of spreading changes. The aggregate procedure avoids that difficulty, because the measure being calculated will be of recalc type, and thus is spread indirectly through a mapping rule.

The rule writer specifies a <cachemeasure> which holds the base intersection values to be aggregated, and is also the source of values for ‘recalc’ aggregation.

The rule writer also specifies the aggregation type for each hierarchy and the priority sequence to be used. The priority sequence is required because at levels that are aggregated in more than one hierarchy (for instance, Department/Region/Month for a measure with a base intersection of Class/Store/Week), different results would usually be obtained by aggregating up each of the hierarchies. Thus, for example, if the requirement is to aggregate up the product hierarchy by using the total aggregation type, up the location hierarchy using the average aggregation type and the calendar hierarchy by using the first aggregation type, there are three potential ways to calculate a value at Department/Region/Month. We could total from Class/Region/Month, average from Department/Store/Month or first from Department/Region/Week. These would almost certainly generate three completely different values. By providing a priority sequence, the rule writer explicitly determines which of these values are required (See worked example, below).

Note that the effect of a series of aggregations of the same type up a single hierarchy may return different results from those of a measure with the same aggregation type. 'Normal' aggregation for a measure driven by its aggregation type is performed from all base intersection cells descended from the cell being evaluated. Thus, for example, for a measure with a base intersection of Class/Week and an 'average' aggregation type, the value calculated for a cell at Department/Month is the average of all values for all Class/Week cells for the Department/Month. If the measure is a recalc measure, calculated at aggregated levels from a rule with an aggregate function such as aggregate(x, [prod].average, [clnd].average), the value for the Department/Month will be the average of all the Class/Months (not Class/Weeks) that belong to the Department/Month. Other than coincidentally, this would generate a different value.

<cachemeasure> is the measure to be aggregated, and the value of <cachemeasure> is also the value used for cells that are at the base intersection (that is, bottom levels), and at aggregated levels when the required aggregation type is recalc. <hierarchy> is the name of a valid hierarchy. Each hierarchy may only be specified once in the procedure, but hierarchies may appear in any order. The sequence that the hierarchies are specified in is used to determine which hierarchy to aggregate up if the cell being evaluated is at an aggregated level in more than one hierarchy. In this circumstance, aggregation is performed up the first specified hierarchy that the cell is at an aggregated level in, and the other hierarchies are ignored. <aggtype> specifies the aggregation type to be used. The <aggtype> must be one of the standard aggregation types (see Appendix in back). If any hierarchy that is in the scope of the measure being calculated is not explicitly specified, the aggregation type of that hierarchy is assumed to be total. Such hierarchies are assumed to be sequenced after all hierarchies that are explicitly referenced, and ordered from innermost to outermost.

Note that the value of the <cachemeasure> is used at the base intersection of the measure being calculated. If, for a given cell, the aggregation type to be used is recalc, the value is also obtained directly from the <cachemeasure> at that level, which will normally have an aggregation type of recalc.

Note that aggregate is a procedure and thus cannot be combined with functions, modifiers, or other procedures in any manner. As a procedure it requires a different syntax: "<-" instead of "=" when being assigned.

Inverse: The aggregate procedure does not have an inverse.

Examples

- **aggregate**(x, [clnd].recalc) For cells at the base intersection, the value is calculated from the measure x. For cells at an aggregated level in the calendar hierarchy, the value is also obtained from the measure x (which we can assume has an aggregation type of recalc, and thus the result of the procedure is as if the aggregation type were recalc, using the usual expression to calculate measure x). If the cell is not at an aggregated level in the time hierarchy, and assuming in this example that the other hierarchies are product and location, in that priority, the value for a cell at an aggregated product level is calculated as the total of all cells for products descended from that product, for the same location and time. Otherwise, the value for the cell is calculated as the total of all cells for locations descended from the cell's location, for the same product and time.
- **aggregate**(x, [loc].average) In a similar manner to the previous example, cells at aggregated levels in the location hierarchy will be calculated by averaging the values of cells for all descendent locations, otherwise the value will be totaled up the product or time hierarchy as appropriate.
- **aggregate**(x, [clnd].average) Totals up all hierarchies except time, which uses an average aggregation type.
- **aggregate**(x, [prod].average, [clnd].first) Averages up the product hierarchy if possible, otherwise takes the first child value up the calendar hierarchy, otherwise totals up the other hierarchies.
- **aggregate**(x, [prod].average, [clnd].last) Averages up the product hierarchy if possible, otherwise takes the last child value up the calendar hierarchy, otherwise totals up the other hierarchies.

Multi-Level Calculation Example

Consider a measure calculated from the expression **aggregate**(x, [prod].total, [loc].avg, [clnd].first). The measure is assumed to have a base intersection of Class/Store/Week. Examples of the calculations that would be applied at various levels are:

Class/Store/Month: first from Class/Store/Week

Class/Region/Month: avg from Class/Store/Month

Department/Region/Month: total from Class/Region/Month

Flookup

'Fixed lookup' function which returns the value of a measure for an explicitly named fixed intersection.

Syntax

flookup*(*<measure>, <posspec1> *[*, <posspec2> … , <posspecn>*])*

Where <*posspec1-n*> is: [<*hierarchy*>].[<*dimension*>].[<*positionname*>]

<*measure*> is the measure to be looked up. This <*measure*> must conform with the measure being calculated as follows. Some hierarchies may be present in the base intersection of both measures, and these are handled by normal 'non-conforming' logic. For any hierarchies that are only in the base intersection of the measure being calculated, all positions will lookup the same value. For any hierarchies that are only in the base intersection of the <*measure*>, the position to be used must be explicitly named through a <*posspec*>. Note that if the position to be used can only be specified indirectly (for example, if it is held in a measure), the *flookup* function cannot be used, and the more powerful *lookup* procedure should be used instead.

flookup can be used to return a constant or a slice. In case of a constant, the NA value of the flookup function will be the value of the constant. In case of a slice, the NA value of the flookup function will be the NA value of <measure>.

For each <*posspec*> that is specified, the <*hierarchy*> must be the name of a valid hierarchy, the <*dimension*> must be the name of a valid dimension in that hierarchy, and the <*positionname*> must be the name of a valid position in that dimension. If <*hierarchy*> is not a valid hierarchy or <*dimension*> is not a valid dimension in that hierarchy, or <*positionname*> is not a valid position in that dimension an *error* is generated.

Additionally, <*dimension*> must be a dimension in the base intersection of <*measure*>. If a level modifier is used, the aggregated <*measure*> will be used in the *flookup* function, and any <*posspec*> that refers to that hierarchy must conform to the new level.

There is no special 'cycle breaking' logic for the *flookup* function. This means that a measure may never be calculated from the *flookup* of the same measure. The *flookup* function returns the value of the expression from the specified fixed intersection.

Inverse:The *flookup* function does not have an inverse.

Examples

- **flookup**(perc, [flvl].[flvl].[flvla])

  Returns the value for the measure perc for the position 'flvla' in the flvl dimension of the flvl hierarchy

- **flookup**(leadtime, [prod].[cls].[class1], [loc].[whse].[whseA])

  Returns the value for the measure *leadtime* for the class 'class1' for the warehouse 'whseA'

Enhancements for Functions with Multiple Results

Several enhancements to the supporting infrastructure for writing functions with multiple results were added to RPAS 11.0.4. Below is a summary of the specifications and assumptions about this type of function, full details are in the developer's guide:

- Functions can contain multiple outputs/results.
- Each output can be a different type (string, integer, real, and so on).
- Parameters and results can be named.
- Results can be optional either by omitting the rightmost results or by using names to denote which results are desired. The function is aware of which results are desired and can use this information to skip unnecessary computations.
- Arguments can be optional but not named.
- Multiple output functions integrate with the protection processing algorithms.
- Arguments passed to multi-result functions can be computed using normal calculation features. But the result of the function cannot have any further calculations applied to it - it must be assigned directly to the LHS.

Position Queries

Overview and Global Design

Position queries provide the ability to define, save, and reuse the definition of a 'query' that results in a selection of positions in a hierarchy, or in the measures dimension. The intention is that, over time, position queries will become all-pervasive in RPAS applications, and may be used at any point where the application requires the selection of positions. 11.0.4 includes the first support for position queries in worksheets only.

Position Queries in Spreadsheet Windows

Position queries in 11.0.4 support the specific case where a context is supplied in one hierarchy that dictates the positions to be shown in another hierarchy. For example, the selected position in a warehouse dimension may be used to range down the store dimension of the location hierarchy to only show those stores 'served by' the selected warehouse. The information that defines which positions should be visible for a specific context is held in a Boolean measure (dimensioned on the context and the dimension being queried), which is calculated normally.

The positions to be shown for a dimension on a view are defined as being the result of a position query, using a specific measure, through the configuration tools. Currently, therefore, there is no mechanism to dynamically configure a worksheet window to use a position query. For such worksheets, the current position of the dimension in the slice area is the context. When the user changes this current position, the worksheet is refreshed to show just the positions required for the new context.

Sanity Check

Sanity check provides a mechanism that allows dynamic lower and upper limits to be placed on measures such that values are validated on edit, prior to being passed into the calculation engine. The limits will be held in measures that may have base intersections that are higher up the hierarchies with the values being 'replicated down' if required. Thus, for example, an upper limit at department may be used to perform a sanity check on, say, an IMU at class/week.

Specifically the following functionality is available:

- Upper and lower bound measures can be defined for any registered measure
- This feature will only function with numeric measures (non-numeric measures will be ignored); however, real and integer types can be mixed
- Bounding measures must be materialized (must be stored in the domain)
- The same hierarchies must be present in the measure being evaluated and the bounding measures
- If a bound measure is defined at an intersection that is not an ancestor of the current intersection, and cannot be aggregated to the current intersection, no sanity checking will apply

navalue

Function that returns the na value of an expression

Syntax: navalue(<expression>)

<expression> can be a constant, a measure, or an expression.

The navalue function does not directly generate errors, but it can propagate errors generated by <expression>.

Inverse:The navalue function does not have an inverse.

Examples:

- **navalue**(<meas>) This returns the NA value of <meas>.
- **navalue**(<meas1> + <meas2>) This returns the NA value of the expression "<meas1> + <meas2>". In this example, if the NA value of <meas1> is 2 and the NA value of <meas2> is 5, then the result of the navalue function will be 7.

RPAS Utilities

The following section contains the syntax of the new RPAS utilities and describes changes made to existing utilities. Only utilities that are intended to be directly run by an administrator or implementer are included in this section.

Loading Measures – loadmeasure utility

Use the admeasure utility to load a measure or apply staged loads for a measure. You must specify the measure name and the path to the domain containing the measure. Usage of the utility is:

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

Argument	Description
–d pathToDomain	Specifies the domain in which to load the measure.
–measure measureName	Specifies the name of the measure to load. The name must be lowercase.
–logdirectory directoryName	Specifies the location of the output error log. The default location is pathToDomain/scripts/err.
–applyloads	Use this argument to apply any staged loads for the named measure.
–loglevel level	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Calculation Engine – mace utility

Purging Rules and Rule Groups

Mace utility now supports the following:

mace –d $TEST_DOMAIN purgeRules

This deletes all rules that are not contained in any rule group.

mace –d $TEST_DOMAIN removeGroup -groupName

This removes the specified rule group. It also removes all rules within the rule group that are not shared by other rule groups.

mace –d $TEST_DOMAIN removeAllRuleData

This removes all rule groups and all rules.

Validating Rule Groups

New parameter (–validate) that takes one of 3 possible values (general, calc, refresh) to specify which type of rule group to validate.

All of these also require the -ruleGroup <ruleGroupName> parameter be given as well. In addition, if you select the refresh option, you must also specify the -calcRuleGroup <calcRuleGroup> parameter. Each refresh rule group has a corresponding calc rule group.

The usage is:

mace -d $TEST_DOMAIN -validate general -ruleGroup MyRuleGroup

mace -d $TEST_DOMAIN -validate calc -ruleGroup MyCalcRuleGroup

mace -d $TEST_DOMAIN -validate -refresh -ruleGroup MyRefreshRuleGroup -calcRuleGroup MyCalcRuleGroup.

Administering Users – usermgr utility

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

Usage:

usermgr –d domainPath –add userName –label label –password psw –group grp{–admin} {–loglevel level}

usermgr –d domainPath –remove userName {–loglevel level}

usermgr –d domainPath –list {–loglevel level}

usermgr –d domainPath –print –user username {–loglevel level}

usermgr –d domainPath –print –group groupname {–loglevel level}

usermgr –version

Argument	Description
–d domainPath	Specifies the path to a domain that you want to add, remove or get information about a user.
–add userName	Use this argument to add a user with a specified name. Use the other arguments specified in the usage to add those attributes for that user.
–label label	Use this argument to specify the label of the user that you are adding to the domain.
–password psw	Use this argument to specify the password of the user that you are adding to the domain.
–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.
–version	Use this argument to get the version information. It does not require –d domainPath.
–loglevel level	Use this argument to set the logger verbosity level. Possible values: all, profile, debug, information, warning, error, or none.

Inspecting and Managing Workbooks – wbmgr utility

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
–d pathToDomain	Specifies the domain containing the workbooks.
–list –all	Lists all workbooks in the domain.
–list –user userName	Lists all workbooks belonging to the user.
–print –wbList wb1,wb2,...	Prints detailed information about workbooks in the list.
–remove –all	Removes all workbooks from the domain.
–remove –user userName	Removes all workbooks from the domain belonging to the user.
–reomove –user userName –wbList wb1,wb2	Removes all the workbooks in the specified list.
–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. It does not require –d domainPath.

Loading Hierarchy Structure – loadhier utility

The loadHier utility can be used to load and refresh a hierarchy. Usage of the utility is:

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

Domain Version Information – domainversion utility

Use the domainversion utility to get version information for a specified domain.

Usage:

domainversion –d pathToDomain

domainversion –code

Argument	Description
–d pathToDomain	Specifies the path to the domain for which you want to get version information.
–code	Use this argument to get version information that RPAS Libraries expect.

RPAS Version Information – rpasversion utility

Note: This usage information was changed in the 11.0.4.6 release notes to include a minor correction.

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

Usage:

rpasversion {–version} pathToLibrary

Argument	Description
–version	Specifies the path to the domain for which you want to get version information.
–pathToLibrary	Use this argument to get version information for a specified library.

Upgrading

The following section describes the process for upgrading all the components of an RPAS solution. This upgrade process is only possible with solutions built on the GA version of RPAS and Tools 11.0.3 with patch 1, and for solutions built on pre-releases of RPAS 11.0.4. The process involves upgrading the RPAS platform, the Configuration Tools, configurations, and domains.

Important Notes about Upgrading

Note the following about the upgrade process:

- The upgrade process is not reversible for configurations or domains.
- All existing 11.0.3 workbooks should be committed, if desired, and deleted prior to running the upgrade utility. The upgrade utility does not upgrade workbooks within the domain, so trying to use an 11.03 workbook after the upgrade will result in an error.
- All the wizard page user selections are deleted – specifically, users/*/selections directories in the domain are deleted, meaning that all stored wizard page selections will no longer exist and users will have to make selections again.
- The domain-upgrade component of the upgrade process performs several upgrade functions. Before performing any of these upgrades, it first validates that the domain has an 11.0.3 format for each component it's trying to upgrade. If any of the components is in an unexpected format, the utility reports an error and leaves the domain unchanged. Otherwise, it will proceed with the upgrade.

Upgrade Process

1. Make a backup copy of your configuration, your domain, and the installs directory for the domain.
2. Install the new Tools release on your computer. Starting with this release of the Configuration Tools, we have made some changes with the installation process of the Tools. These changes are particularly important if you are ***not*** using the NT server installation of RPAS to build domains. If you are not using the NT server installation of RPAS on your laptop or desktop, refer to the RPAS Installation Guide, *Chapter 6: Installing the configuration tools in Windows*, regarding the changes for this release of the Tools. Even if you have already installed a previous release of the Tools, you ***must*** make these changes for this release.

   Specifically, we have separated the RPAS NT-specific files from the Tools release into their own directory called **RPASlibs**. The RPASlibs directory is packaged as part of the Tools release. You need to setup the **RPAS_HOME** environment variable to point to the path where you install the RPASlibs directory. Even if you are not using the RPAS NT server installation to build domains, you need to setup the RPAS_HOME variable to point to the RPASlibs directory to use the Tools Workbench. Once you have setup the RPAS_HOME environment variable, you need to add *%RPAS_HOME%/lib* directory to your **PATH** environment variable.
3. If you want to build domains on NT or Windows 2000, you need to install the new RPAS NT release on your computer. This installation process has not changed. Refer to the RPAS Installation Guide for complete details. Change or create your **RIDE_HOME**, **RPAS_HOME**, and **PATH** environment variables as necessary to point to the new Tools and RPAS installations respectively. If you are not building domains using the RPAS NT installation, you can skip this step.

   Follow the process below depending on your respective scenario, either Scenario 1 or Scenario 2. Then proceed to the remaining instructions for Both Scenarios.

Scenario 1: Upgrade from RPAS/Tools 11.0.3 to 11.0.4

This scenario is only for configurations that have not been previously upgraded to a pre-release of RPAS 11.0.4. Otherwise, go to Scenario 2.

Upgrade the configuration:

- Open your configuration using the new Tools installation.
- You will be prompted to convert the configuration. Select Yes.
- As part of the conversion process, rule set, rule groups, and rule names may be truncated in order to meet new validation requirements. Specifically, rule sets will be truncated to the first nine characters. Rule groups will be truncated to the first ten characters. Rule names will be renamed to be the new rule group name plus a rule index autogenerated numeric identifiers. The conversion process will also convert rule group names used in workbook custom menus to reflect the new rule group names. The converted rule set, rule group, rule names, and custom menus should be manually inspected. Note that there should be no impact to rules and expressions as a result of the conversion.
- To assist you in updating any external scripts that reference rule group names, a log file will get created called RuleRenamer.log located in your configuration directory.

- We strongly recommend that you do not make any changes to your configuration during the upgrade process. After upgrading your configuration, save and exit your configuration.

Scenario 2: Pre-existing RPAS/Tools 11.0.4 configuration and domain

This scenario is only for configurations that have been previously upgraded to a pre-release of RPAS 11.0.4.

Upgrade the configuration:

- Double-click on the RuleRenamer.bat file located in the bin subdirectory of your Tools installation. This will open up a new utility that will convert your configuration. Note that you should not have your configuration open in the Tools Workbench during this scenario.
- Use the Browse button to open up your configuration .xml file. Select the main .xml file of your configuration the same as you would as if you were opening the configuration in the Tools Workbench.
- As part of the conversion process, rule set, rule groups, and rule names may be truncated in order to meet new validation requirements. Specifically, rule sets will be truncated to the first nine characters. Rule groups will be truncated to the first ten characters. Rule names will be renamed to be the new rule group name plus a rule index autogenerated numeric identifiers. The conversion process will also convert rule group names used in workbook custom menus to reflect the new rule group names. The converted rule set, rule group, rule names, and custom menus should be manually inspected. Note that there should be no impact to rules and expressions as a result of the conversion.
- The renamed rules will be displayed in the RuleRenamer. We recommend that you copy and paste this into a text file in order to save this information for updating any external scripts.
- Exit the RuleRenamer.
- Go to step 6 below. We recommend that you do not make any changes to your configuration during the upgrade process. You do not need to open and save your configuration in the Tools Workbench for the rule conversion to take effect. The RuleRenamer modifies your configuration directly.

Both Scenarios (Except Step 10):

1. Change any external batch scripts to use the new rule group names and to reflect parameter and/or name changes to certain RPAS utilities (including **loadmeasure**, **loadHier**, and so on)
2. On the server, move or rename your old configuration. Zip up your new configuration and transfer it to the server. Do ***not*** unzip the new configuration over the old configuration.
3. If using a UNIX server, change your **RIDE_HOME** path in your .profile to point to the new Tools installation.
4. If using a UNIX server, change your **RPAS_HOME** path in your .profile to point to the new RPAS release. Restart your session for these changes to take effect.
5. For **Scenario 1** only (upgrading from 11.0.3 to 11.0.4), run the upgrade utility on your domain using the command below:
   ```
   domainUpgrade1104 –d domain_path
   ```
   > **Note:** This script will call the RPAS upgrade1104 utility. This script will remove all rule groups except the RPAS ones.
6. Rebuild all custom libraries using the new RPAS release. Move the custom libraries Into the $RPAS_HOME/**applib** directory.
7. Since RPAS no longer references libraries from the **lib** directory of the domain, remove this directory from the domain.
8. Run a **patchinstall** on your domain using the new configuration. The installs directory must be located in your domain home path at the same level as your domain for the patchinstall to work.

This will complete the upgrade process for the RPAS platform, Configuration Tools, and existing domains.