Next Generation Retail Planning Application for Cloud

Submitted By

Birva Bhatt

16MCEC01

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Next Generation Retail Planning Application for Cloud

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering

Submitted By

Birva Bhatt

(16MCEC01)

Guided By

Dr. Zunnun Narmawala

DEPARTMENT OF COMPUTER ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2018

Certificate

This is to certify that the major project entitled **"Next Generation Retail Planning Application for Cloud"** submitted by **Birva Bhatt (Roll No:16MCEC01)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Engineering of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-I, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Zunnun Narmawala
Guide & Associate Professor,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Priyanka Sharma
Professor,
Coordinator M.Tech - CSE
Institute of Technology,
Nirma University, Ahmedabad

Dr. Sanjay Garg
Professor and Head,
CE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Alka Mahajan
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Birva Bhatt**, **16MCEC01**, give undertaking that the Major Project entitled "**Next Generation Retail Planning Application for Cloud**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. Zunnun Narmawala

(Signature of Guide)

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Prof. Zunnun Narmawala**, Associate Professor, Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr. Alka Mahajan**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Birva Bhatt

16MCEC01

Abstract

RPAS stands for Retail Predictive Application Server. It is used in the retail industry to forecast the current market trends based on the previously known data. It is basically a platform which provides the infrastructure needed to process and produce information based on the input provided by the retailer. In basic terms, it helps in planning of inventory by analyzing the customer needs and behaviour and minimize the stock wastage due to improper planning.

This project focuses on a Cloud based Client application for RPAS. It acts as an interface between RPAS and users, and provides various functionalities which include creation of workbook, projecting the data in graphs, alerts etc. The areas of Cloud based Client covered in this project are Online Administration Tools, Dynamic Hierarchy, Position Filtering, Export Images to Excel and Edit View Overlay. These modules have been implemented in Cloud Client for enhancing the customer experience on cloud. The other area of Cloud Client that is covered is Automation Testing of Cloud Client UI using Synergy.

Abbreviations

RPAS	Retail Predictive Application Server.
FC	Fusion Client.
ADF	Application Development Framework.
SAAS	Software As A Service.
PAAS	Platform As A Service.
OAT	Online Administration Tools.

Contents

List of Figures

Chapter 1

Introduction

1.1 Oracle Retail Planning

Retail industry is highly dependent on consumers, their demands and their choices. Along with all the different retail outlets at different locations, from remote small towns to major cities, consumers all over the world have a huge array of options from where they can buy all their stuffs. With the advent of online shopping, their options have only increased. This has increasingly made the competition among various retailers all the more fierce, as they now need to predict the consumers future demands, all the while, being in touch with current market trends.This means that retailers all over the world, need to gauge the market trends and plan accordingly. They need to plan for the profits they may generate in the future, by analyzing the current demands of the consumers, to predict their future demands.

Planning and optimizing helps the retailers to make profitable decisions based on future insight and allows effective collaboration among their different departments and their suppliers.This allows to the retailers to maintain their stock inventory, by changing supplies of different products, depending upon market trends, season and consumer interest. Optimizing their supply according to consumer demands further adds to their bottom line. In order to provide superior customer experience, they have to utilize all their data and plan accordingly to make price, inventory and assortment decisions.

Chapter 2

Retail Predictive Application Server (RPAS)

2.1 Introduction

The Retail Predictive Application Server is a calculation engine for multiple planning and optimization applications.[1] It provides the base for various Oracle Retail solutions like Oracle Retail Demand Forecasting (RDF), Merchandise Financial Planning (MFP), and Advanced Inventory Planning (AIP).[3]

In retail industry, RPAS is used to forecast the current market trends based on the data already known. It is a platform which provides the infrastructure needed to process and produce information based on the input provided by the retailer. It helps in planning of inventory by analyzing the customer needs and behaviour and minimize the stock wastage due to improper planning.

With a proven scalability for development of solutions which are based on multidimensional forecasting and planning, RPAS is a configurable platform. Multidimensional structure of database, online and batch processing, slice-and-dice user interface and calculation engine which are configurable, users security and utility functions like importing and exporting, are the capabilities provided by RPAS platform, all of them on an environment which is highly scalable and technical, that can be deployed on a range of hardware.[3]

2.2 Basic RPAS Concepts

2.2.1 MultiDimensionality

In RPAS, multidimensional framework is the basis of storing and representing information. In a database system which is multidimensional, data is stored in the form of a multidimensional array which makes individual data value of a cell to be accessible through multiple indices.

Multidimensional database systems achieve higher performance levels than the relational database systems. RPAS considers dimensions like calender,product,location and time. These dimensions contain sub-components which can be rolled up or down based on requirement.[4]

2.2.2 Dimensions

Dimensions describe a top-to-bottom relationship between the positions or levels of the dimensions in RPAS. They reflect the dimensions that are set up as per the business needs. RPAS supports many dimensions that provide various roll ups to help analyze the data from a different perspective.[4]

2.2.3 Measures

Measures represent the values that are registered, while a context for the measurement is provided by the positions in the dimensions. The business rules set in the application are used to define the measures. The base intersection of Measures is used to configure the dimensionality of a measure. It is a collection of levels that define the lowest level used to store the information for the measure. Measure names can be configured as per need. They are typically given meaningful names such that each it's meaning and use can be identified from the name.[4]

2.2.4 Domains and Workbooks

Information is stored in a multidimensional data cache which is persistent and which is optimized as per huge volumes and time series or dimensional data access requirements,normally a requirement of multidimensional solutions. This is a central repository called as domain. There can be multiple domains in one solution. Out of these, one will be a master domain, while others are called local domains.

A Global Domain environment provides a centralized view of data from all the domains present in the solution, in a single workbook. It contains all the physical local domains, and thus apart from the admin users, other users generally have no access to this domain. Local domains are organized and partitioned depending on organizational structure, reflecting user responsibilities and roles. Most users need not be aware about the global domain as they work on their local domains assigned according to their respective area of responsibilities.

A workbook is basically a subset of data stored on the domain and its visibility and scope depends on the access rights of different users. These are stored on the server. The data and metadata present on the workbook is copied from the domain. Templates are present in RPAS which defines the workbook's look and the type of data that's present. Client gets these templates via server bridge APIs and creates workbooks using those templates.[4]

2.3 RPAS Operational Architecture

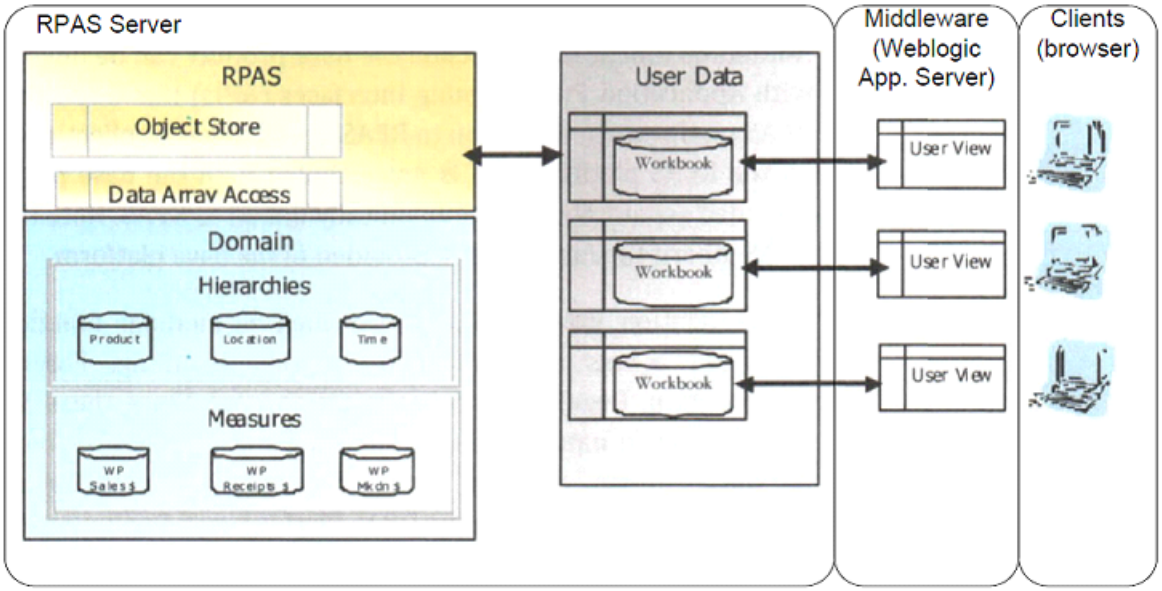

Figure 2.1: RPAS Operational Architecture[1]

RPAS server, consists of RPAS data and Domain. The RPAS data consists of object store and data array access, while the domain is built from configuration tool with the configurations provided. The configuration consists of hierarchies and measures. The hierarchies define the dimension in which the data is represented.

Whenever a user creates a workbook, it is stored on to the RPAS server. This work-

book data is connected to the view layer on the WebLogic server which hosts the FC application.[1]

2.4 RPAS and FC architecture

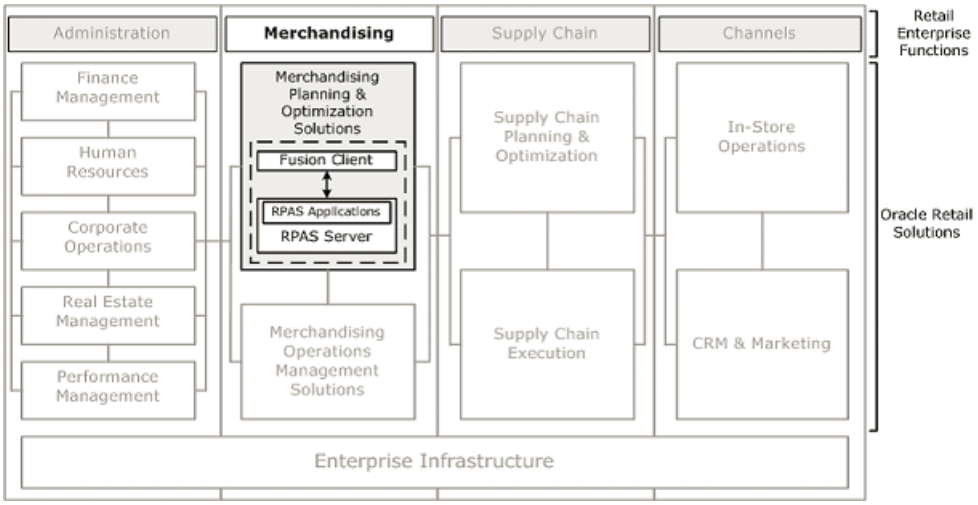

Figure 2.2: RPAS Position in Oracle Retail Product Suite[2]

The Retail Predictive Solutions comprises of a set of products that are used to generate forecasts and develop plans, taking into consideration the current consumer demands and market trends. These products use predictive technology to examine historical data and to predict future behavior. The Retail Predictive Solutions run on a common platform called the Retail Predictive Application Server (RPAS).[1]

The RPAS Client is a web-based application developed using ADF framework to run on both, desktop and mobile browsers. It provides a visual representation of the data on the server depending upon the domain configuration.Multiple taskflows have been defined at the client that describes the flow of data among different pages. This enables the user to perform multiple tasks in a single session.[1]

2.5 ADF Architecture

Model-View-Controller (MVC) architecture is followed by ADF. Any MVC application generally consists of:

- A model layer whose responsibility is to maintain and store data.
- A view layer that defines the application user interface and displays data from the model, as per controllers functionality.

- A controller that handles page flows in the entire application, handles user input and interactions between the Model and the View layers.

ADF architecture is based on four layers:

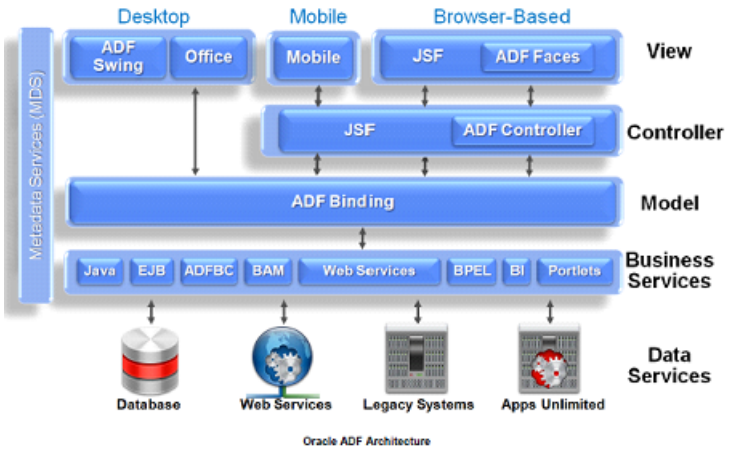

Figure 2.3: Oracle ADF Architecture[1]

- The Business Services layer provides data access from multiple sources using different implementation technologies, and handles business logic.

- The Model layer abstracts the different implementations of Business Services layer, thus allowing View and Controller modules to work seamlessly. It allows for the development of UI that is loosely coupled from the underlying business services.

- The Controller layer provides flow control for all the JSF pages, and stores the state of each operation in the flow, to enable reusability.

- The View layer provides the user interface of the application. Different UI development technologies are provided for desktop, mobile, and web browsers.

Chapter 3

Oracle Cloud

As Oracle Retail being the retail unit moving it's solutions to cloud, here is a survey regarding what drives it towards industrial cloud and why is cloud more advantageous for the organization compared to current on-premise deployments.

3.1 Advantages of Industrial Cloud

- Small scale industries can avail the cloud services and it's advantages because of it's flexible cost. The benefits will not be limited to just large scale organizations.
- Easy and quick accessibility to any software or hardware for customers, hence benefiting the market.
- Flexible capacity which can be scaled up or down depending on requirement. It helps for cut-backs and cost-reduction.
- Most cost effective for our customers as Oracle cloud provides PAAS and SAAS.Customers don't need to have their own deployment and infrastructure centres or big IT force. They just need to rent it and pay-as-you-go basis which is beneficial for the organization as well hence creating a win-win situation.
- Since retail solutions are generally desktop deployed, it is affordable in terms of reducing latency and available network bandwidth to leave the data crunching to the cloud and have the fron-end do just the rendering of processed data.

3.2 Limitations of Industrial Cloud

- Security - Security has always been an issue with cloud environments. Securing customer data is a critical need for us and that too in a multitenancy architecture, it is of prime importance to guide each of them seperately.

- Interoperability - As big as an advantage it is, it is a disadvantage at the same time. With higher scalability, it is necessary to make the platform interoperable at the same time else the application might under-perform.

- Economic Investment - No matter how cost-effective the cloud deployment might seem, initially there is a lot of investment economically as well as resources required to jump into it. Varied Short-term and long-term risks are also involved with this advent.

- IT Structure changes - There are demands for huge chance in current IT Structure for moving to cloud. Expertised team with relevant skill-set is required for cloud engineering. It is a completely fresh field which requires lot of research and development along with capital investment and time for the same. It is a risky business decision to move to cloud.

Chapter 4

Online Administration Tools

4.1 Introduction

RPAS has a client-server architecture where planning actions can be done either via FC workbooks or by running batch calculations after logging into the server. Administrators log into the server and perform the batch maintenance tasks in routine. This set-up for the online tasks work well with on-premise deployments but not with cloud environment because in case of on-premise set-up, administrators have full access to back-end servers whereas in cloud environment, there is limited access to back-end servers. This enhancement for cloud environment gives an interface that allows authorized users to login to server and launch the batch processes from FC. It also makes it easy to keep a track of all submitted and scheduled processes via dashboard like interface called Status Dashboard.

4.2 Motivation

As a part of Oracles requirement to move their product suite to the cloud and provides its services, Admin Task Manager and dashboard interface has been developed. The Software-As-A-Service (SAAS) approach demands that most of the tasks that were traditionally carried out at servers, should now be accessible from the cloud as well. The user must be able to perform all the server-side tasks and should be able to run the RPAS utility commands. RPAS Admin Task Manager is supposed to provide that interface, allowing authorized users to launch back-end processes from the Client application itself.

To display the status information of the submitted tasks to the user in the client itself, a dashboard-like interface needed to be developed. The requirement was that the

user, according to his/her access privileges, should be provided a filterable view of all the submitted tasks.

4.3 Implementation

- The entire project is divided into multiple Bean and Controller classes for both Admin Task Manager and Status Dashboard with one .jspx page as View for each. Beans store the data that is fetched from their respective Controllers at Data Access Layer using server provided APIs. XML files on the server acts as Data Stores for both the tasks.

- The Admin Tasks are defined in XML format. There can be any number of such XML files. Each XML file contains a list of admin tasks and this file is stored in the domain on the server. Any new admin task can be added to this list as per the requirement. Each task contains an argument list which may have multiple level branching.

- Each operation in the Admin Task Manager is carried out by use of certain wizard-only workbook templates defined in the server. There are four templates defined: Task Submission template, Task Modify template, Task Copy template and Task Removal template. Each Template is a multiple page template where the next page depends on the output of the previous page.

- The DB server receives the request sent by wizard and en-queues the task. It then sends an inter-process signal to Task Daemon, which controls the back-end processes. It creates the processes, monitors their status and maintains the queue.

- The various controls in the wizard pages are dynamically generated depending upon the inputs from the domain and server. The control options for each wizard pages are pre-defined at the server. Respective page control data and page layout data are received from the server.

- Page control data contains information about the current page and next page name and label. It also contains information about the buttons (Next, Previous and Finish), whether these buttons are enabled or disabled for the current wizard page.

Page layout data contains the information about the wizard page dimensions and the associated controls in it.

- Controls such as radio buttons, check-boxes, input text boxes etc. and their respective positions are rendered dynamically, by making small additions to the existing setup for wizard pages. Selection state of the controls options are also provided.

- Some components, say a list or text box, may be dependent upon another component, say radio button. This dependency comes directly from the server, and the values of the other dependent input components in the wizard page are refreshed or cleared accordingly.

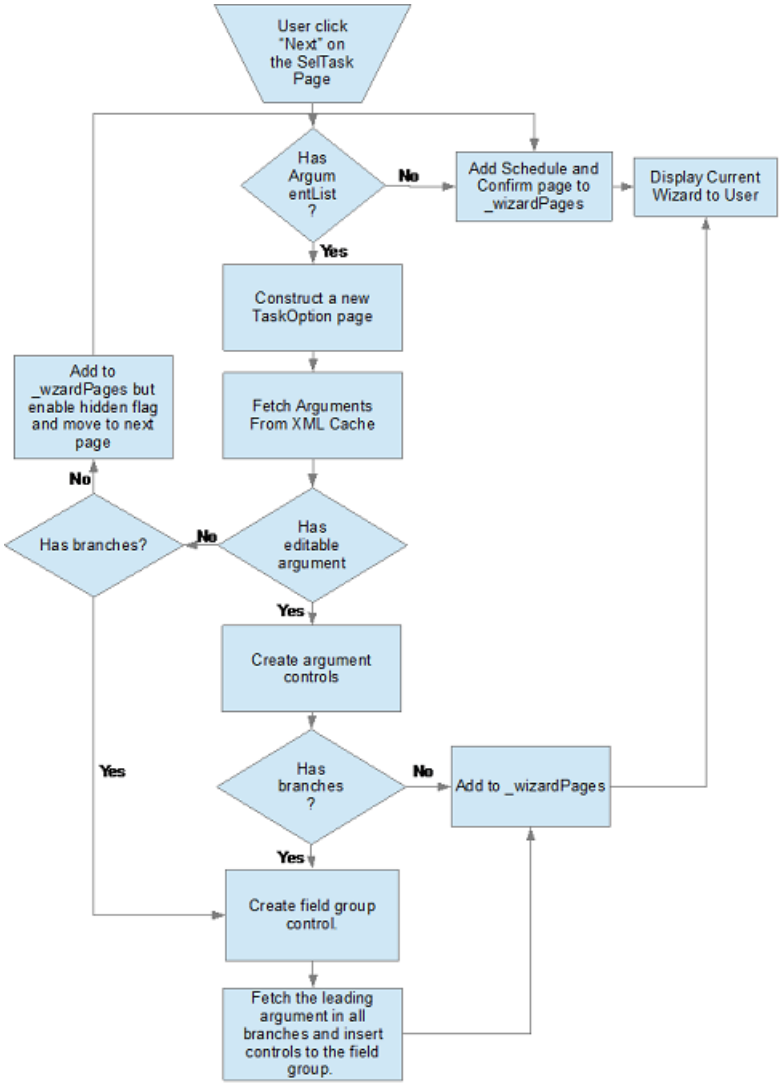

Figure 4.1: Initial Task Options Page

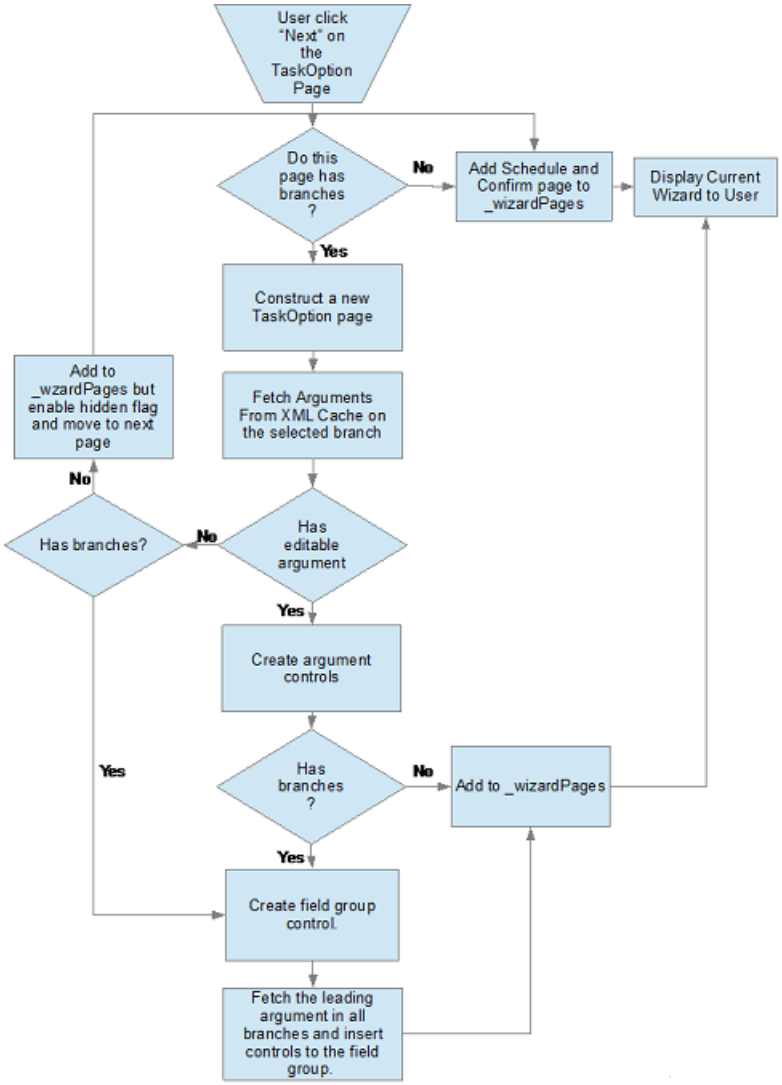

Figure 4.2: Subsequent Task Options Page

4.4 Utilities and RPAS Commands

There are many utility commands of RPAS that can be executed from the client side.. Few of them are mentioned below.

- domaininfo This utility provides miscellaneous domain information such as the current domain version, its upgrade history, type of domain (local or global) and a listing of the subdomains. Multiple options are present that give more detailed information about the domain.[2]

  domaininfo d pathToDomain -options

- checkDomain - This utility checks the validity of any existing domain, be it local or

master. It verifies whether a master domain matches its respective local domains and informs the administrator about all the discrepancies. This command checks whether the domain is local/master and is present in the mentioned directory or not. For the global domains, it checks whether the local domains that are a part of global domain in question are present in their respective directories with their type as sub. [2]

- regmeasure - The regmeasure utility is used for batch measure registration. It registers a new measure in the user-specified domain with the user-specified measure properties. Registering a measure in global domain, registers it into all the master domains and their subsequent local domains. Measures base intersection, type are some of the parameters necessary to register any measure. Any existing measure can be unregistered from the user-specified domain. If the specified domain is a global domain, this measure will be removed from the master domain and all local domains. Unregistering a measure from a domain will cause the measure definition to be removed. [2]

  regmeasure -d pathToDomain add/modify/remove measureName type typeName

- loadmeasure - This utility loads the measure data into the specified domain from the input text files. The administrator need to specify the measure names and its domain path. To load measure data, system administrators must copy or create one or more load files in the input folder of the domain directory. The administrator can then call loadmeasure to load data. A CSV file is used to load the measure (single or multiple). [2]

  loadmeasure d pathToDomain measure measureName -options

4.5 Templates for Admin Task

Following are the pages used in the above mentioned templates:

4.5.1 Task Group Selection Page

This wizard page prompts the user to choose one of the task list group stored as XML file on the server. A request to XML Parser is made which in turn requests XML Cache for the list of available groups. This request is initiated by the global Controller class but

at the lowest level, the request made at the DAO layer. If only one group is present, this wizard page is not displayed. But if, there are multiple groups, then all the groups are listed along with respective radio buttons for selection in the wizard page.

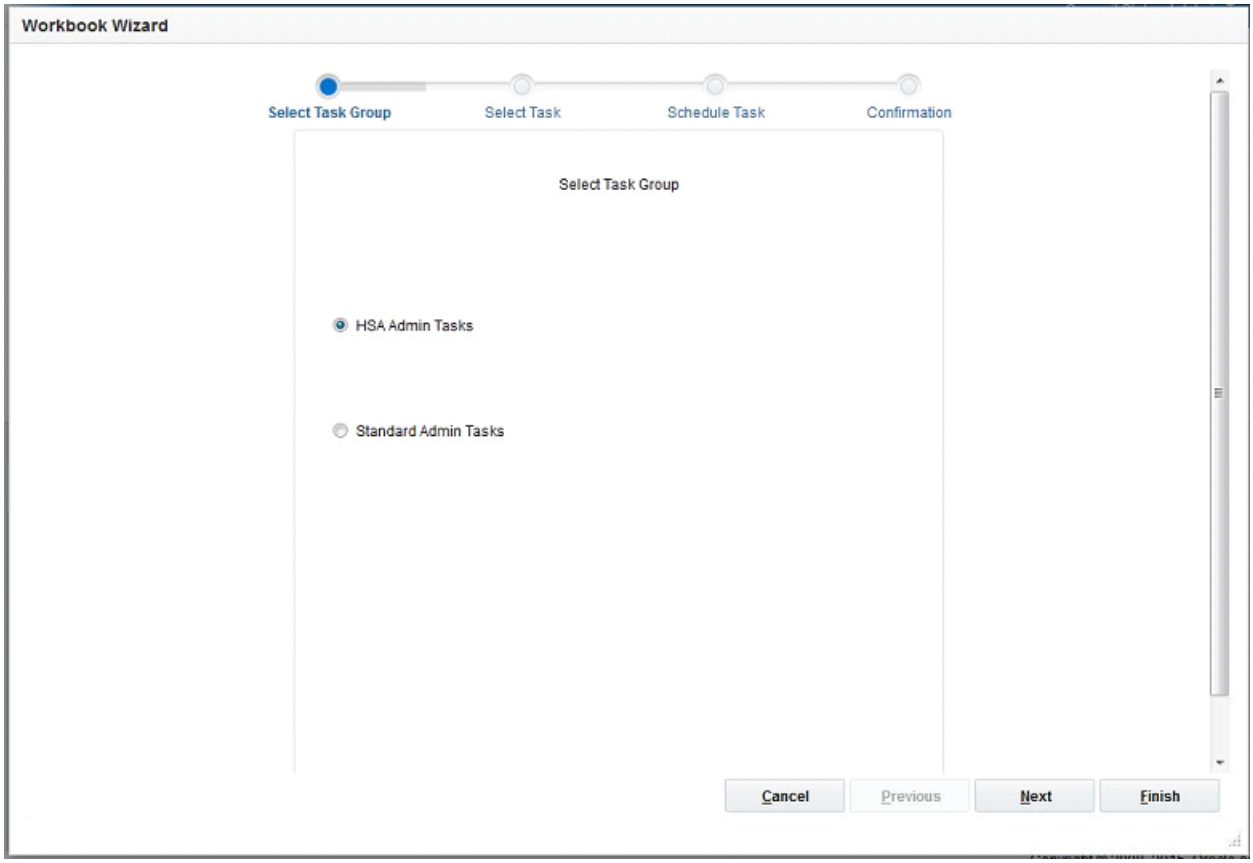

Figure 4.3: Task Group Selection Page

4.5.2 Task Selection Page

This wizard page allows the user to select one of the predefined tasks fetched based on the id of the group selected in the previous page. The user has to enter the label of the new task, which will further be used to identify the submitted task.

4.5.3 Task Arguments Page

The third page allows the user to enter the command-line arguments. The controls on the page are based on the options or arguments definition defined in the XML for the task selected in previous page. More pages may be created depending on the branching and user input until all the branches are covered, recursively. If the admin task has no options present in the XML, this page is hidden and the user is directly sent to next wizard page. This page is different for different selected tasks.

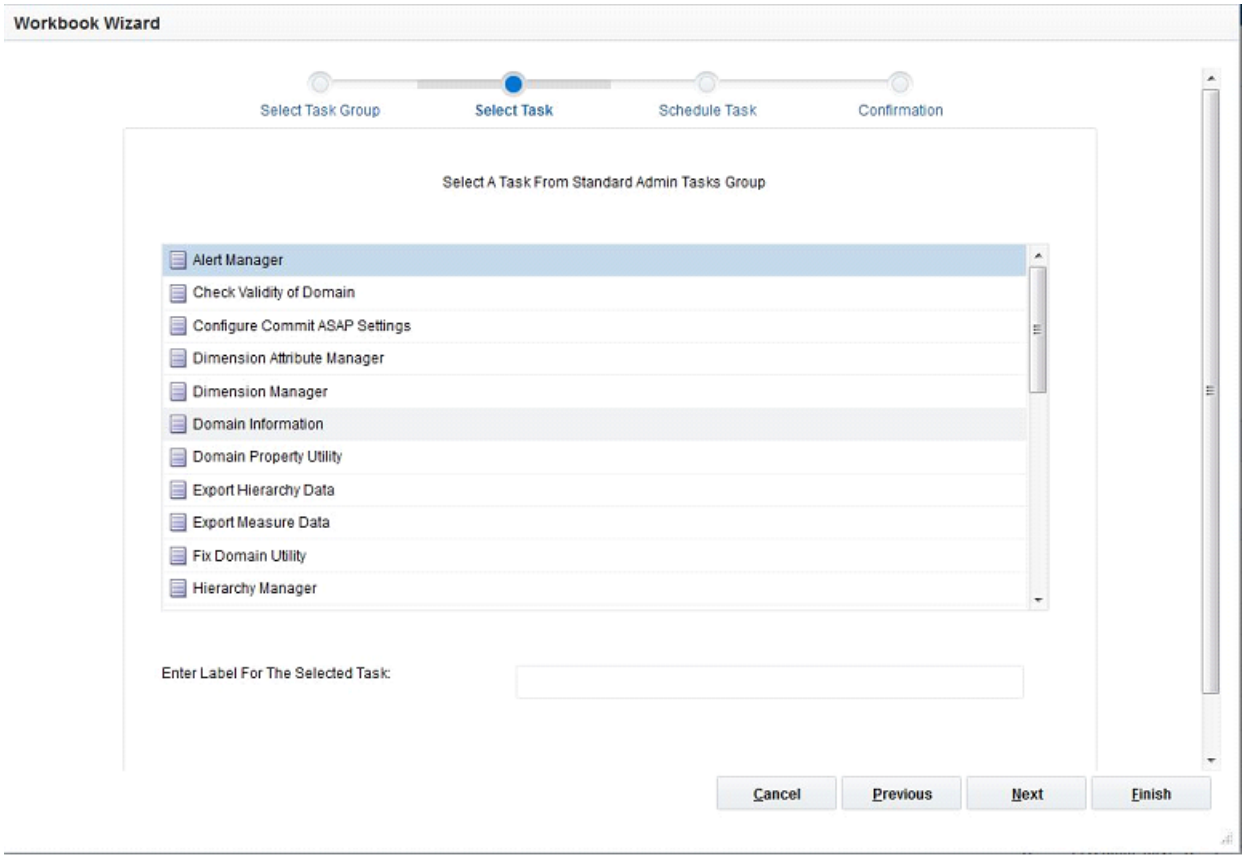

Figure 4.4: Task Selection Page

4.5.4 Task Scheduling Page

The fourth page has the scheduling options as for when the task should run. The user has the options of starting the task immediately or at some later time. The tasks can also be scheduled to run periodically. This page has the options and controls such setting task start date and time, if the task should run daily, weekly and so on. This page is displayed when all the options wizard pages have been exhausted.

4.5.5 Task Confirmation Page

The final page is the confirmation page which displays the user inputs entered previously and asks for user confirmation. On clicking finish, the task gets submitted and the wizard lifecycle ends.User can cancel the entire process at any point of time. Also, user can navigate between the pages sequentially, back-and-forth. For any particular task, the user can go back and the inputs entered are retained only until the any other task is selected in the process flow.

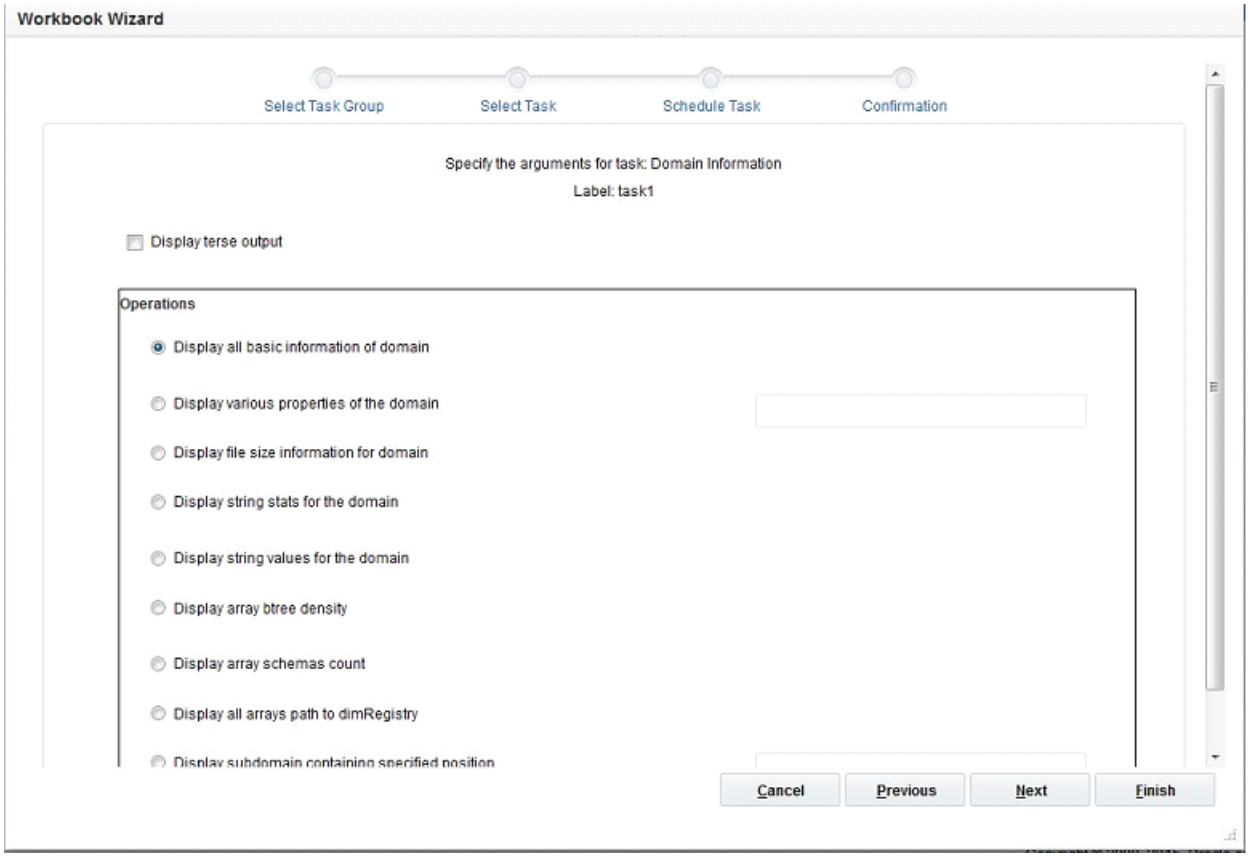

Figure 4.5: Task Arguments Page

4.6 Admin Task Submission

- It allows submission of new tasks by the admin user. It consists of all the pages mentioned above in sequential order. The wizard pages are generated for all the five types of pages sequentially and their lifecycle ends when the wizard is cancelled or finished.

- The user selects the task group, task name and enters a Task Name for it. According to the task selected multiple wizard pages may be generated depending on the level of branching and whether there are editable arguments.If the entered argument value is associated with another branch, a new options page is generated. If there are no options, then the wizard page remains absent.

- The user then schedules the task and confirms it on the next page. The user can go back to the previous page to edit any values entered. But if the user changes the selected task, then the entered values will be cleared. The subsequent wizard pages generated will be based on the new selection.

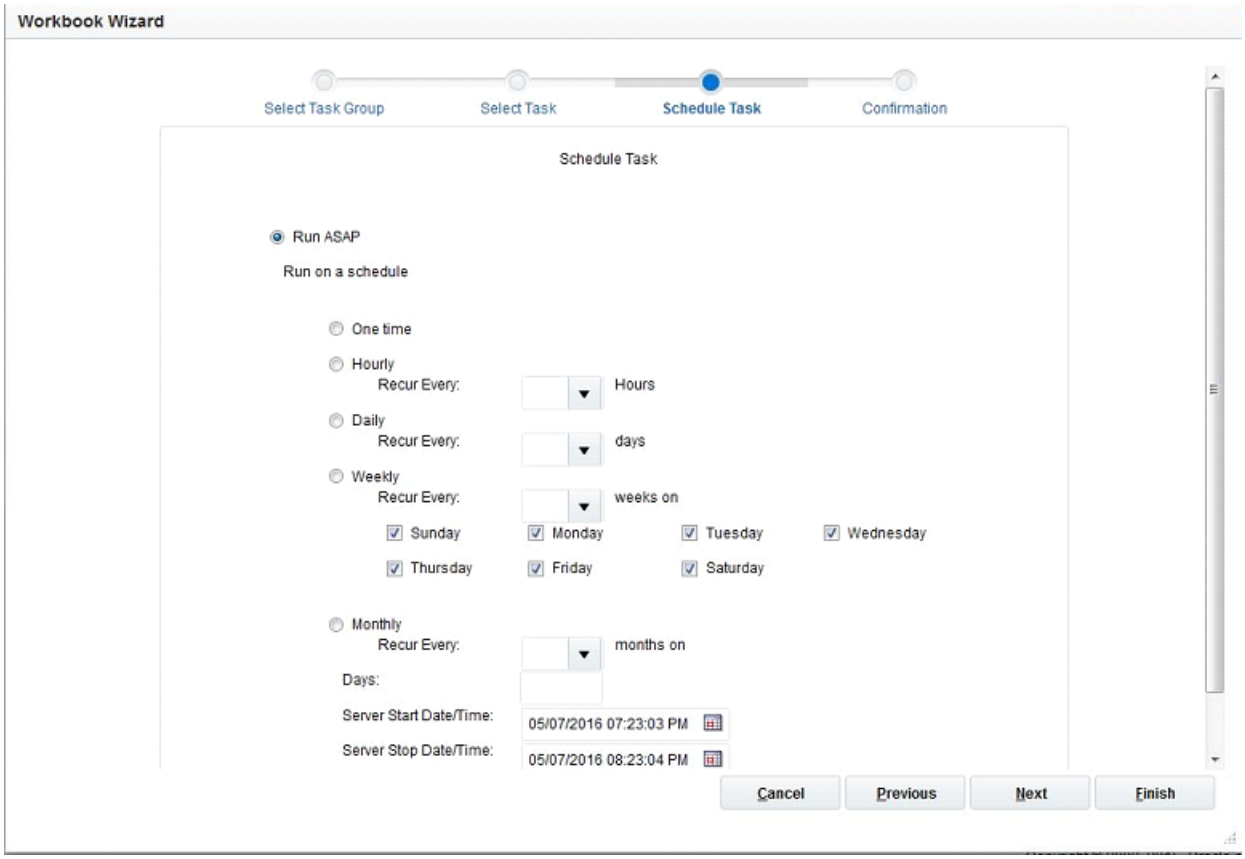

Figure 4.6: Task Schedule Page

- When the task gets submitted in FC, it propogates instantaneously to the server, where it executes, depending upon the entries in Scheduling wizard page.

4.7 Admin Task Copy

- It allows the user to copy a previously submitted task and submit it as a new task. Some tasks may be used repeatedly, much more than others. Or one user might want to copy a certain task submitted by other user. This template is useful in such scenarios where duplicate tasks need to be submitted.

- It consists of for wizards: Task Selection Page, Options Page, Scheduling Page and Confirmation Page.

- The inputs to Options Page are also prefilled with the values from the task being copied. Any task whose definition is not present in the XML file is not shown in the task list.User, also, has the option to modify the values of certain parameters as required, to submit a new task essentially.

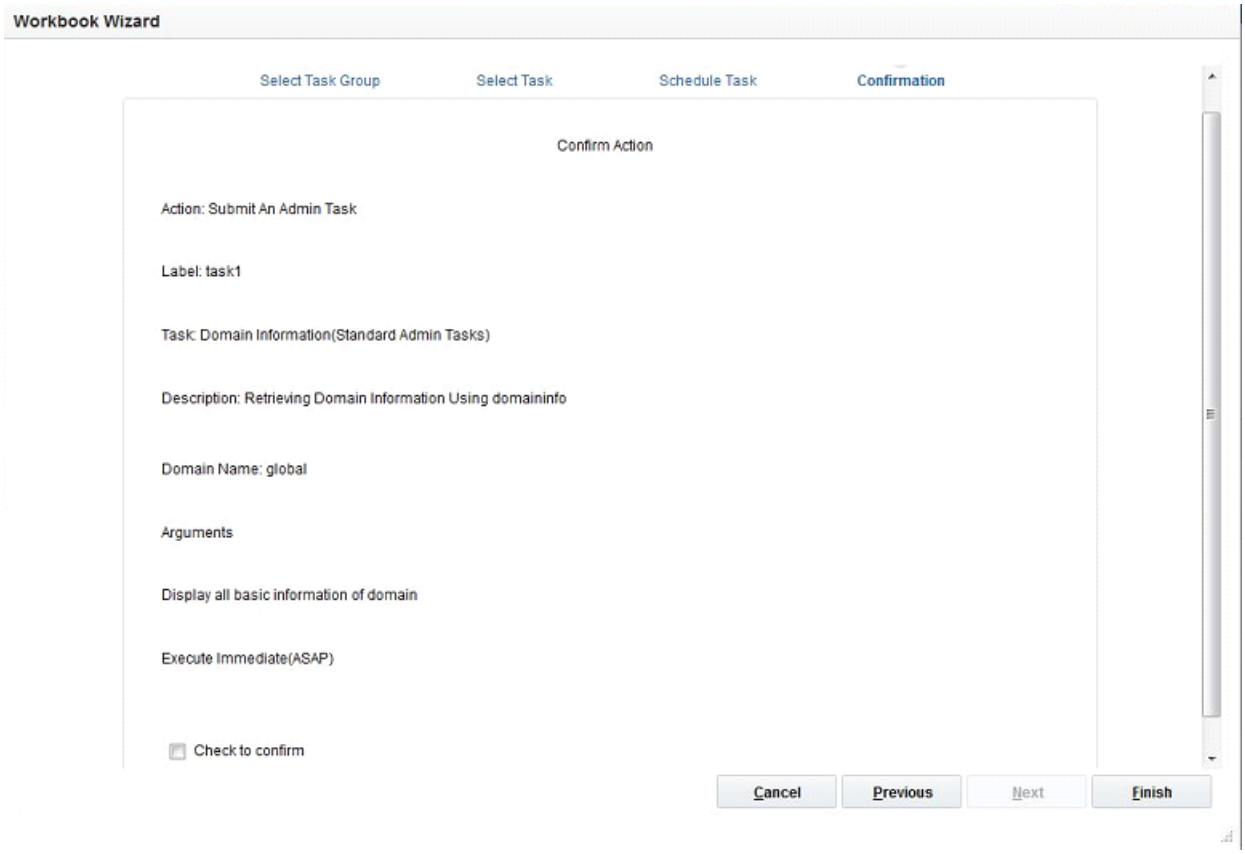

Figure 4.7: Task Confirmation Page

4.8 Admin Task Modify

- It has the same pages as that of Task Copy Template except that the Task Selection Page presents a list of tasks that have not started which is fetched from another XML file. The user can rename the label in another wizard page and modify the task by selecting different options in the Options Page or by scheduling it for some other time.

4.9 Admin Task Removal

- It consists only of Task Selection Page and Confirmation Page. Again, The Task Selection Page fetches the list from different XML file containing a list of all the pending tasks. It removes the task from the execution queue

4.10 Admin Status Dashboard

- Status Dashboard displays the status of tasks that were run / scheduled to run, by admin Task Manager. It has details of each task. The user can see the status and

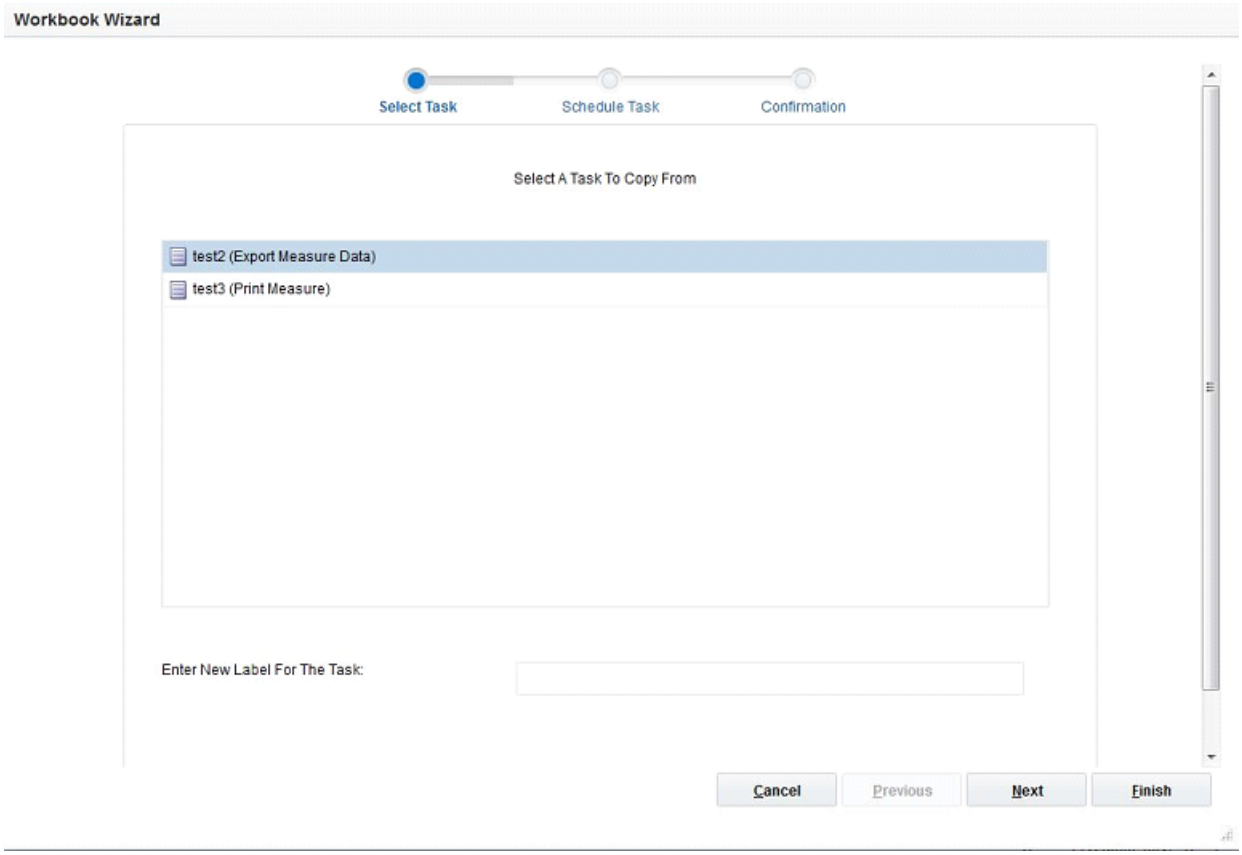

Figure 4.8: Task Selection for Copy Task

obtain log for debugging purpose from this dashboard. The dashboard provides the user with task related details such as: Submitter of the task, when the task was started and completed, on which Solution/Domain the task was run and so on.

- The default screen has all tasks submitted in the last seven days, sorted by Submission Time in descending order.A link is provided on the home page to make this dashboard easily accessible.

- The status dashboard contains rows, each row displaying details of a particular task. This data will always be fetched from the RPAS by using a bridge API. Thus the data in the Status Dashboard is on request basis.

- Following columns are displayed:

  – Solution/Domain It displays the solution and domain combination of the submitted task. There can be multiple solutions attached to each client and there can be multiple domains in each of these solutions.

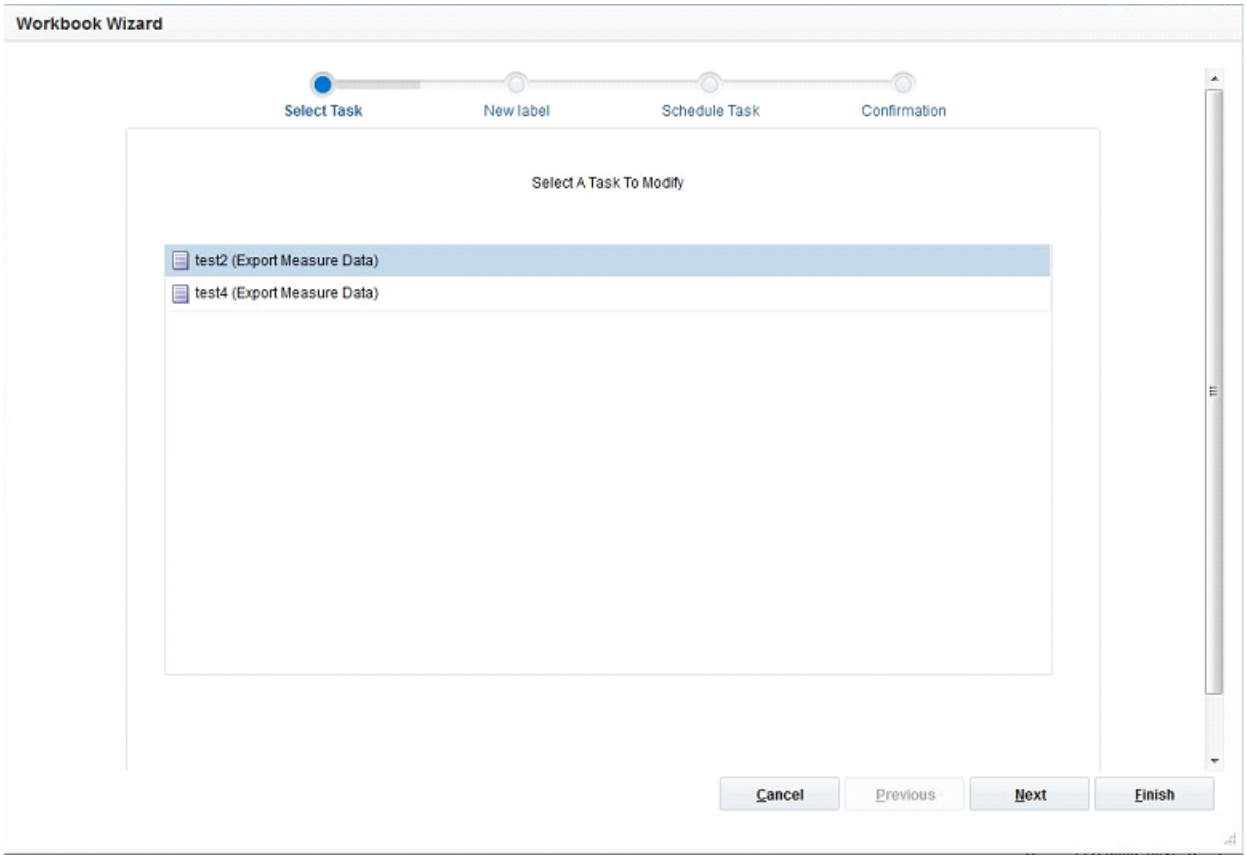

Figure 4.9: Task Selection for Modify Task

- Task Name It displays the task name that was entered by the user during task submission. This same name can be used for multiple tasks.
- Task ID It displays the unique task id generated at the server side when the task was submitted. This helps to differentiate between the tasks with same names.
- Submitter IT displays the name of the user who submitted the task.
- Submission Time It displays the time of task submission by user.
- Start Time It displays the time at which the submitted task started/will start. This is the same time value that was entered by the user in Scheduling Page wizard while submitting the task.
- Completion Time The time at which the submitted task finished execution. This field has an entry only when the task has completed, else it remains blank.
- Status This field gives the status of the submitted task. The field may contain following values: Success, Failure, In Progress, Scheduled, Pending.

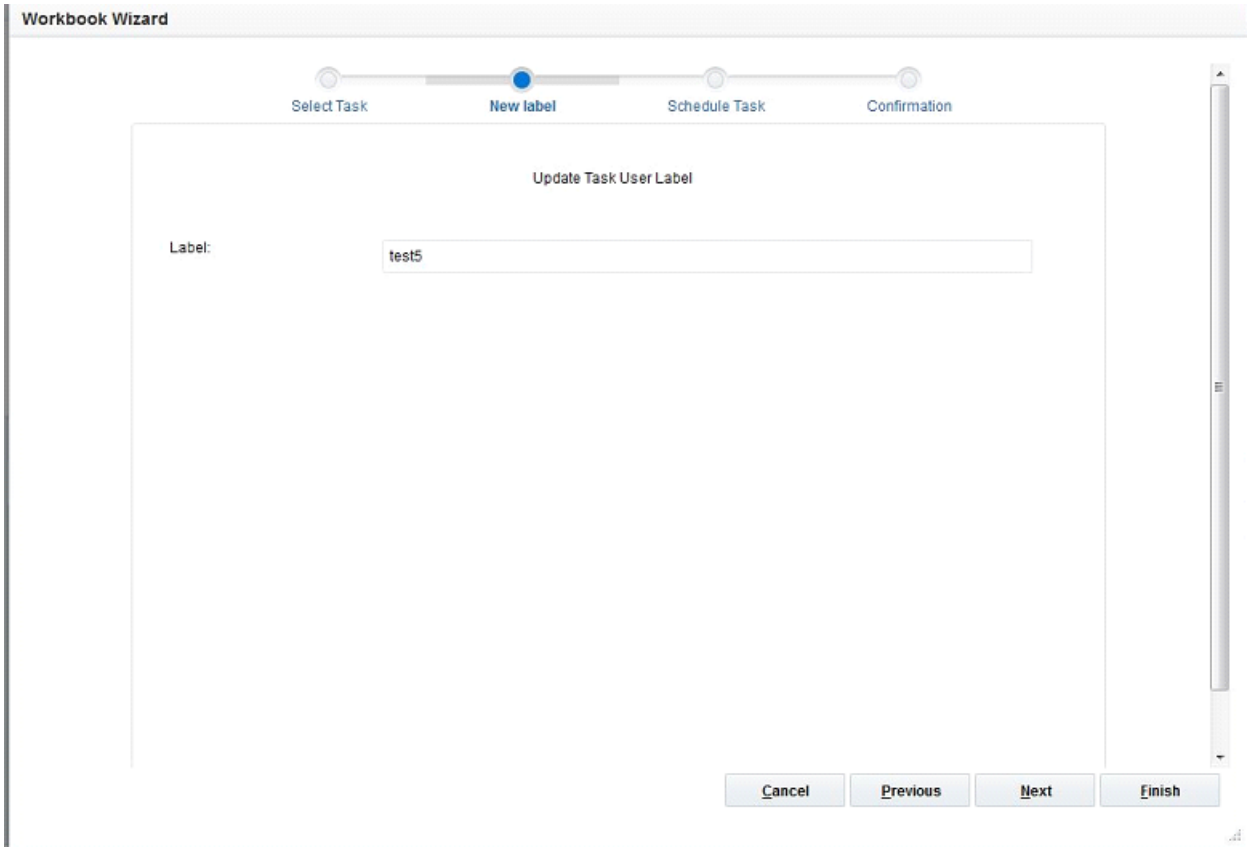

Figure 4.10: Task Label for Modify Task

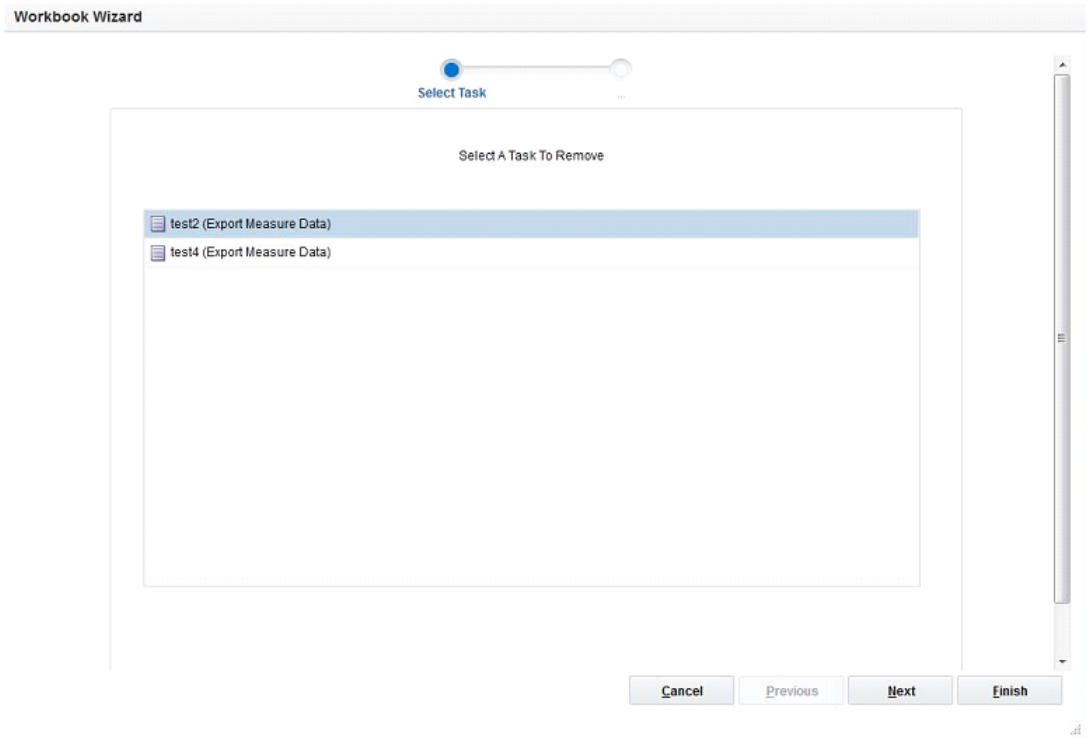

Figure 4.11: Task Selection for Delete Task

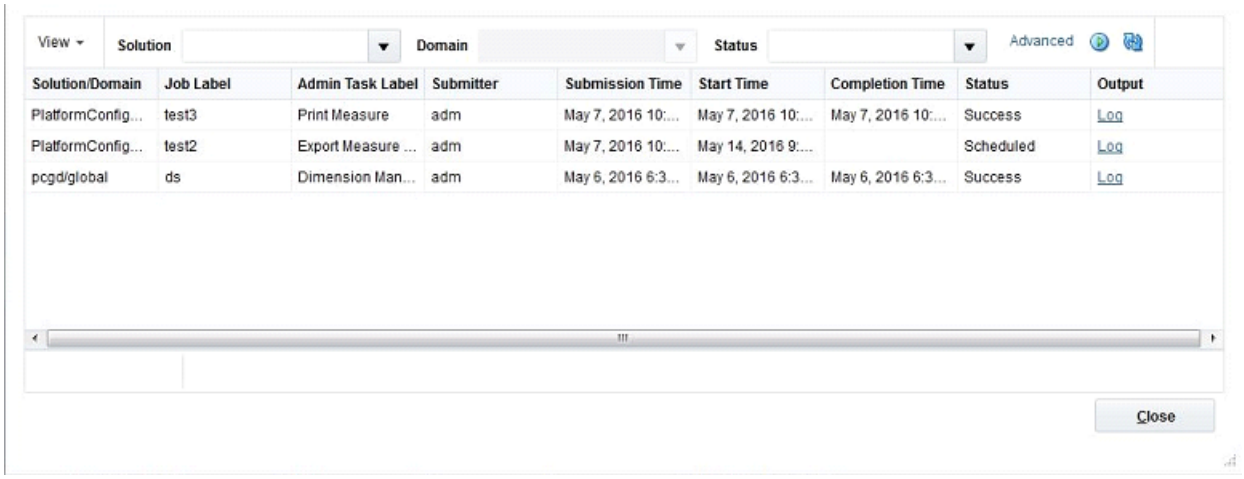

Figure 4.12: Status Dashboard

- Various filters are provided to the user for filtering of visible task list in the dashboard. The user can filter the task on the basis of following: Submitter, Solution, Status, Domain, Date. On initial load of the dashboard, default filters are applied, wherein, only the task submitted in the last 7 days are displayed. This has been done to optimize the performance.

- Logs and messages can be downloaded for all the tasks by the user by clicking on the link provided in the status dashboard in each row for each task. Message link only appears for the failed tasks.

- During the initial loading of the dashboard, task status list is initialized. First the call is made for obtaining the filter values and then to obtain task status entries with default filter values. Since the dashboard gets loaded each time it opens, all data and filter values are cleared and new data is fetched.

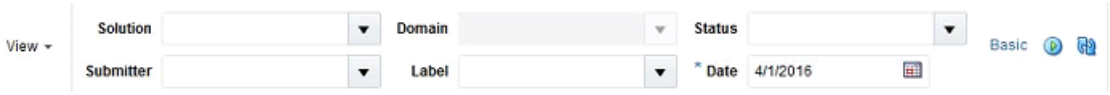

Figure 4.13: Status Dashboard Filtering

Chapter 5

Dynamic Hierarchy

5.1 Introduction

A hierarchy is a dimension within a workbook whose relationship is dynamic based on the context of the workbook and can vary from one workbook to another. The positions within a hierarchy are built using measure data during the workbook build process. There are two types of dynamic hierarchies: domain modified dimensions and workbook only dimensions.

5.2 Methodology

For rolling up to a particular level within a dimension, previously, the whole workbook had to be re-created. To avoid this overhead of recreation of the workbook, Dynamic Hierarchy feature is introduced, where even if the levels attributes changes, the user can successfully roll-up to the required level.So currently the dynamic rollup information gets setup at the time of creating the workbook. The requirement from Assortment Services is to have the ability to refresh the dynamic rollup information without having to close and open the workbook again.

To support this feature a new RPAS function dynHierRefresh is being implemented which will refresh the positions of the dynamic dimensions (both domain and workbook only). It will return either a Success or Failure string based on the execution of the refresh function.

The only way to trigger this refresh is via a custom menu. So the configurator will have to configure a custom menu if the ability to refresh dynamic hierarchy is needed

for a specific workbook. Before triggering the refresh, the mapping measure that stores the rollup information must be updated. It can either be loaded into the domain or committed by a previous workbook operation. So when we execute the custom menu, the dynHierRefresh function gets the new roll-up information from the mapping measure in the domain and updates the dynamic dimensions in the workbook.

5.3 Implementation

The dynHierRefresh function must be added to a rule and rule group. This rule group must then be added to the custom menu of the required workbooks. The LHS measure of the rule should be a scalar string measure("Success" or "Failure"). The rule group and custom menu can be configured as shown in the fig.[3]

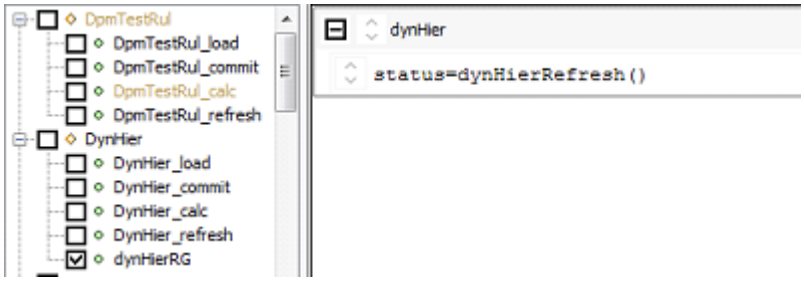

Figure 5.1: Rule Definition for Dynamic Hierarchy

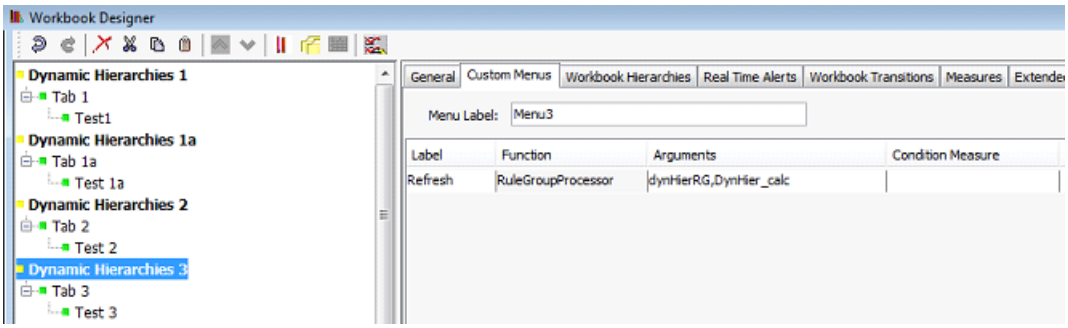

Figure 5.2: Working defining the Dynamic Hierarchy

This function does the actual work of refreshing the dynamic hierarchies. This function must only be configured in a rule group via the custom menu. It should not be run against the domain. If it is run against a domain, it will log an error message. Before starting the refresh, we check if there are any dimension attributes defined on the dynamic dimensions. If yes, then we cannot proceed with the refresh. Even if a dynamic dimension position is stored as a slice information of a dimension attribute, the refresh is

not allowed. If there are no dimension attributes then we can proceed with the refresh. Since we will be updating the rollup information, we make a copy of the dimension registry and workbook meta DB and work on the copy version. Once we refresh successfully, we rename the working copy.

The dynHierRefresh function only updates the rollup information in the workbook database. This change must be propagated to the UI. Currently once we execute the custom menu, only the grid/measure data gets updated. Once we execute the Refresh custom menu, we also want the Tree (positions) to be updated.

To support this, the client needs to call an existing bridge API ServerGetPositionContext to update the tree. However, we want the tree to be refreshed only in this specific case when we are executing the dynamic hierarchy custom menu. So an additional parameter has been added to the existing Menu Event bridge API which will be used to determine whether we need to refresh the tree or not.

Chapter 6

Position Filtering

6.1 Introduction

When a workbook is created initially, a large amount of data, or positions are included. But when user needs to perform operations on only a set of data, it was cumbersome. The Position Filtering feature helps segregate only a set of positions on which the user can perform the required action. This avoids the overhead of data management.

Position Filtering is the functionality to use the set of selected data cells in one worksheet view to filter another worksheet view. For example, a user selects a set of cells related to a group of items, locations and measures which helps in clear visualization of subset of data. This feature is available through a context menu from a page edge.[2]

6.2 Types of Position Filtering

When position filtering is applied, the cells visible in the view are reduced to those specified by the selected positions. Position filtering can be

- Progressive
- Regressive

6.2.1 Progressive Position Filtering

In progressive position filtering, a set of positions are selected in the current view and position filtering is applied. The view generated from it contains only the selected positions from the parent view. [4]

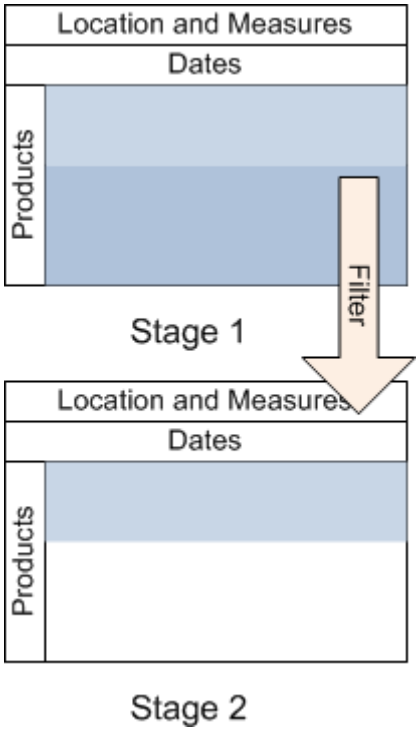

Figure 6.1: Progressive Position Filtering[2]

6.2.2 Regressive Position Filtering

In regressive position filtering, a set of positions are selected in the current view and position filtering is applied. The view generated from it contains only the selected positions from the parent view. Now positions in second view are selected and filtered to match the positions in the parent view.Position filtering can be applied to multiple views progressively or regressively-as in a sequence or flow of four-five views. [4]

6.3 Initiating Position Filtering

Position Filtering can be applied from

- Row or Column
- Page Edge
- Selecting Continuous/Discontinuous Cells

6.3.1 Position Filtering from Row/Column

Position Filtering can be triggered for current view or a child view by selection one or more rows or columns. After a series of rows or columns are selected, right click to get

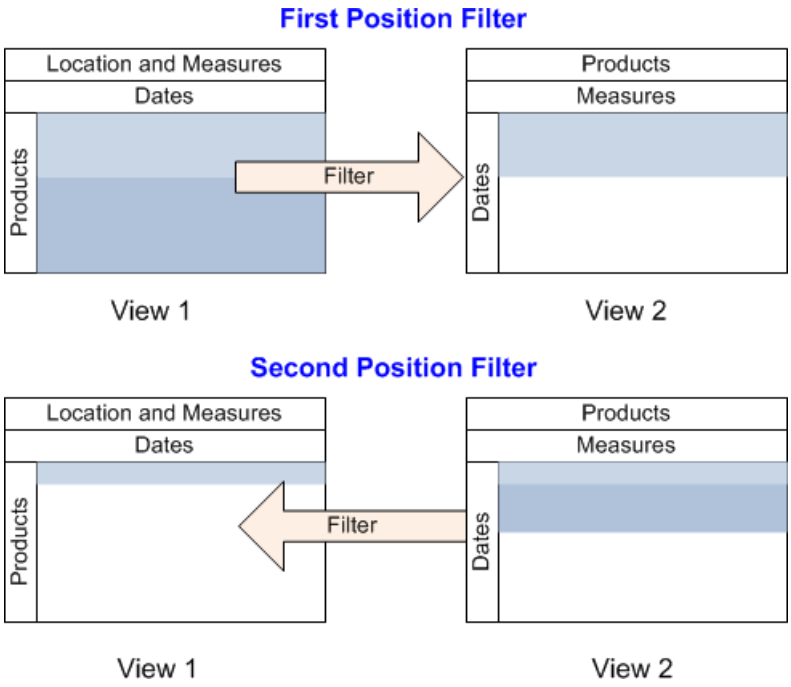

Figure 6.2: Regressive Position Filtering[2]

the context menu and initiate Position Filtering. It can be applied to Current view or to other visible views of the current worksheet.

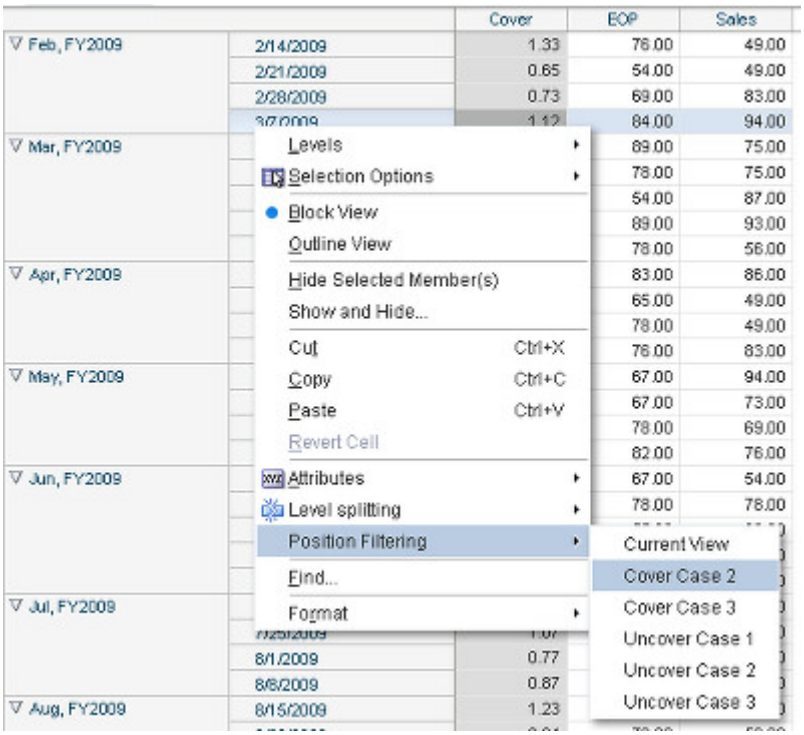

Figure 6.3: Position Filtering from Row/Column

6.3.2 Position Filtering from Page Edge

Position Filtering menu appears under context menu after right clicking on page edge as shown in the figure below. It can be applied to Current view or to other visible views of the current worksheet.

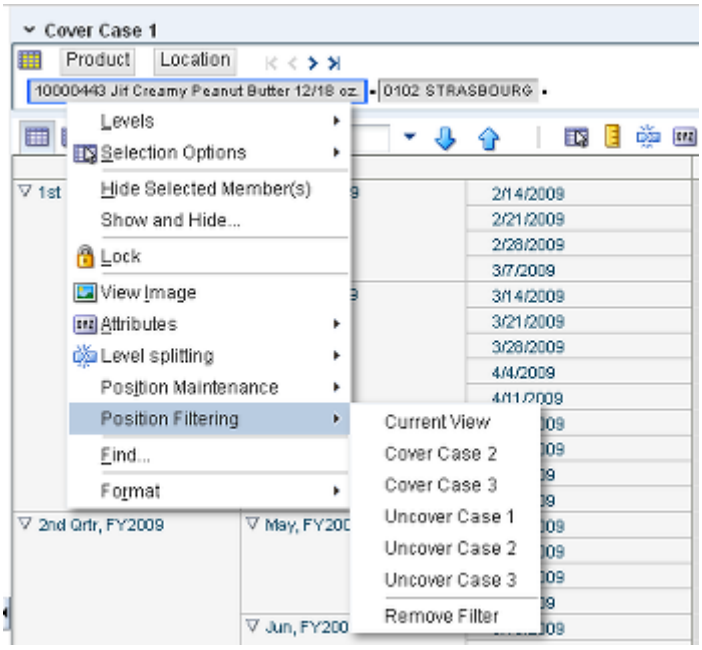

Figure 6.4: Position Filtering from Page Edge

6.3.3 Position Filtering from Cells

Position Filtering can be applied after selection a number of continuous/discontinuous cells in the view and later right clicking to get the context menu. It can be applied to Current view or to other visible views of the current worksheet.

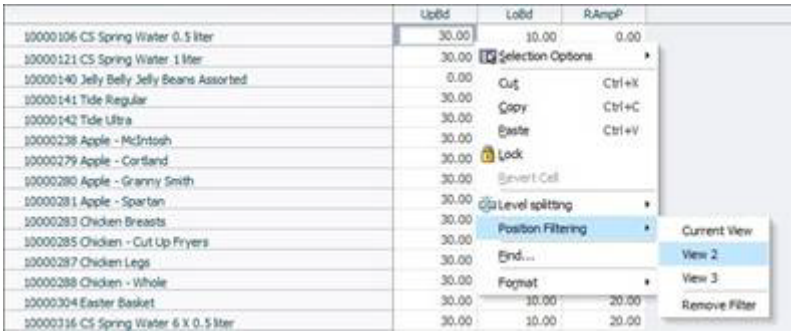

Figure 6.5: Position Filtering from Cells

6.4 Implementation

The Worksheets that are configured for Position Filtering shall be grouped by Step, with Current View at the top followed by Worksheets of the same Step. Other Worksheets belonging to the same task but a different step will be cascaded and grouped under the Step name.The format for Step name is a configurable. Users can specify the format for the installation in a configurable property file which can be localized.This pattern is not solution specific instead it is specific to an installation. This property is configured in rpasBundle.properties.[1]

Customer/user can configure as which all worksheets should be listed. This can be done by adding/deleting worksheets in the Taskflow_Multisolution.xml. Customer can configure which worksheet names should be displayed in step by making changes in ⟨*candidate_worksheets*⟩. Worksheets which come under one task are grouped together and thereby increasing the accessibility.

```
<step>
  <name>PCGD.Activity1.Activity1.Task7.Step6</name>
  <description>PCGD.Activity1.Activity1.Task7.Step6.Desc</description>
  <order_num>1</order_num>
  <instructions>PCGD.Activity1.Activity1.Task7.Step6.Instructions</instructions>
  <worksheet label="PCGD.Activity1.Activity1.Task7.covercase1" >
    <name>covercase1</name>
    <candidate_worksheets>
    covercase1,covercase2,covercase3,uncovercase1,uncovercase2,uncovercase3
    </candidate_worksheets>
  </worksheet>
  <worksheet label="PCGD.Activity1.Activity1.Task7.covercase2" >
    <name>covercase2</name>
    <candidate_worksheets>
    covercase1,covercase2,covercase3,uncovercase1,uncovercase2,uncovercase3
    </candidate_worksheets>
  </worksheet>
  <worksheet label="PCGD.Activity1.Activity1.Task7.covercase3" >
    <name>covercase3</name>
    <candidate_worksheets>
    covercase1,covercase2,covercase3,uncovercase1,uncovercase2,uncovercase3
    </candidate_worksheets>
  </worksheet>
</step>
```

Figure 6.6: Candidate worksheets declared in Taskflow_MultiSolution.xml

6.5 Tiling in Position Filtering

In order to work with Position Filtering conveniently, Tiling of Views is introduced. Here, the user can have multiple views simultaneously and can work on them at a time. Filter can be applied on one view and the resultant positions are visible on adjacent view selected. In this manner, the current view remains undisturbed and filtered out positions are visible in the adjacent view selected by user.[2]

The position filtering that is applied is based on four dimensions of the worksheet i.e. column, row and also two dimensions along the page edge. First priority is for row and column and if nothing is found common along row and column, page edge is taken into consideration.

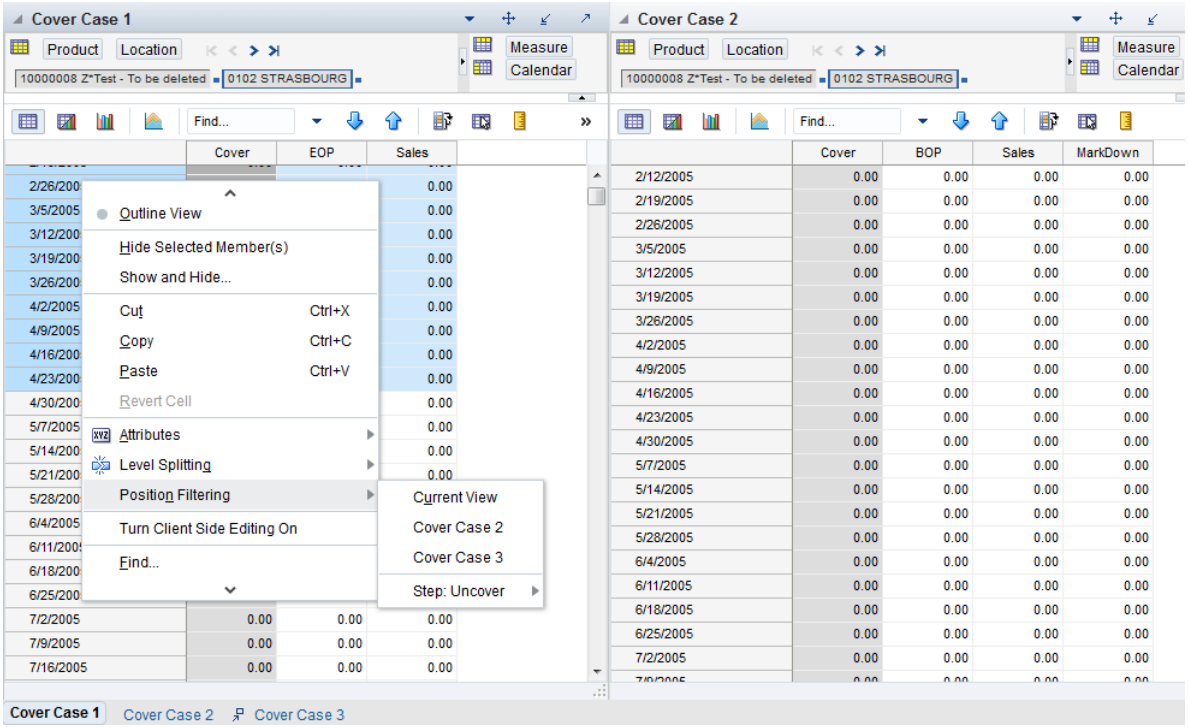

Figure 6.7: Position Filtering Stage 1

As shown in figure,a few positions are selected in Cover Case 1 and position filtering is applied to Cover Case 2. Later, Cover Case 2 shows only the filtered out positions. Hence it makes convenient for user to work just with the filtered out positions in the second view rather that scrolling through hundreds of positions. The applied position filtering can be removed anytime to display all positions as before.

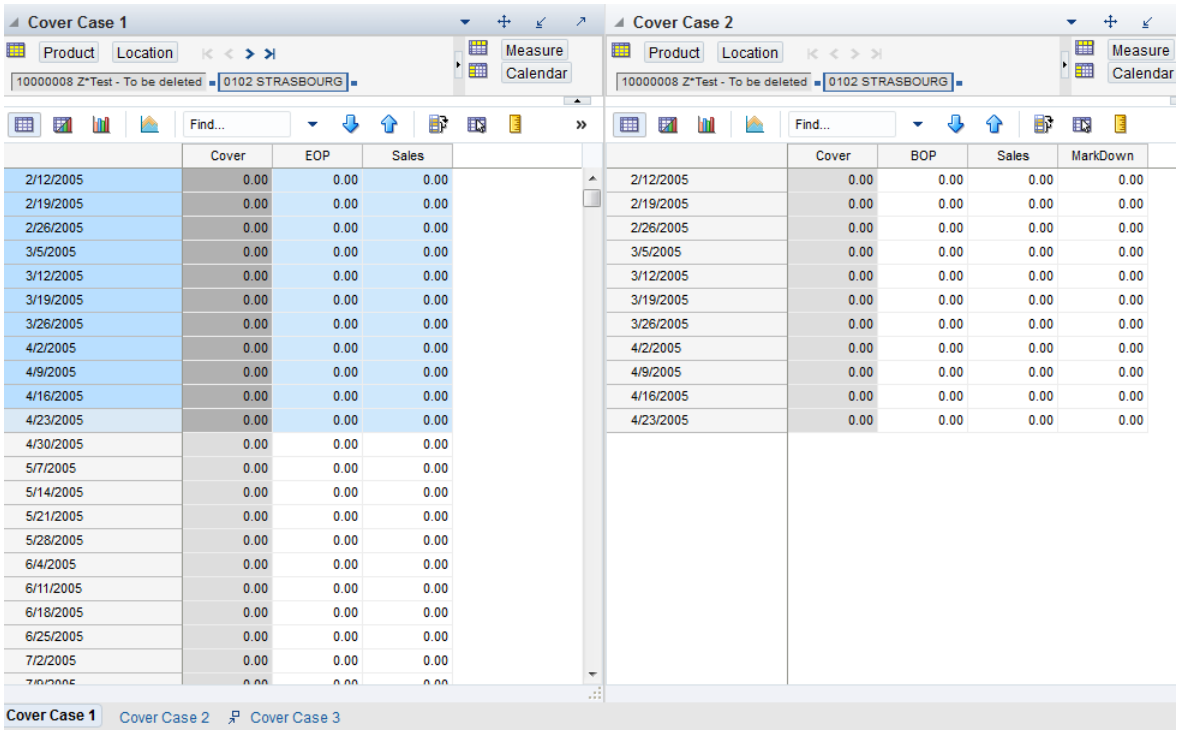

Figure 6.8: Position Filtering Stage 2

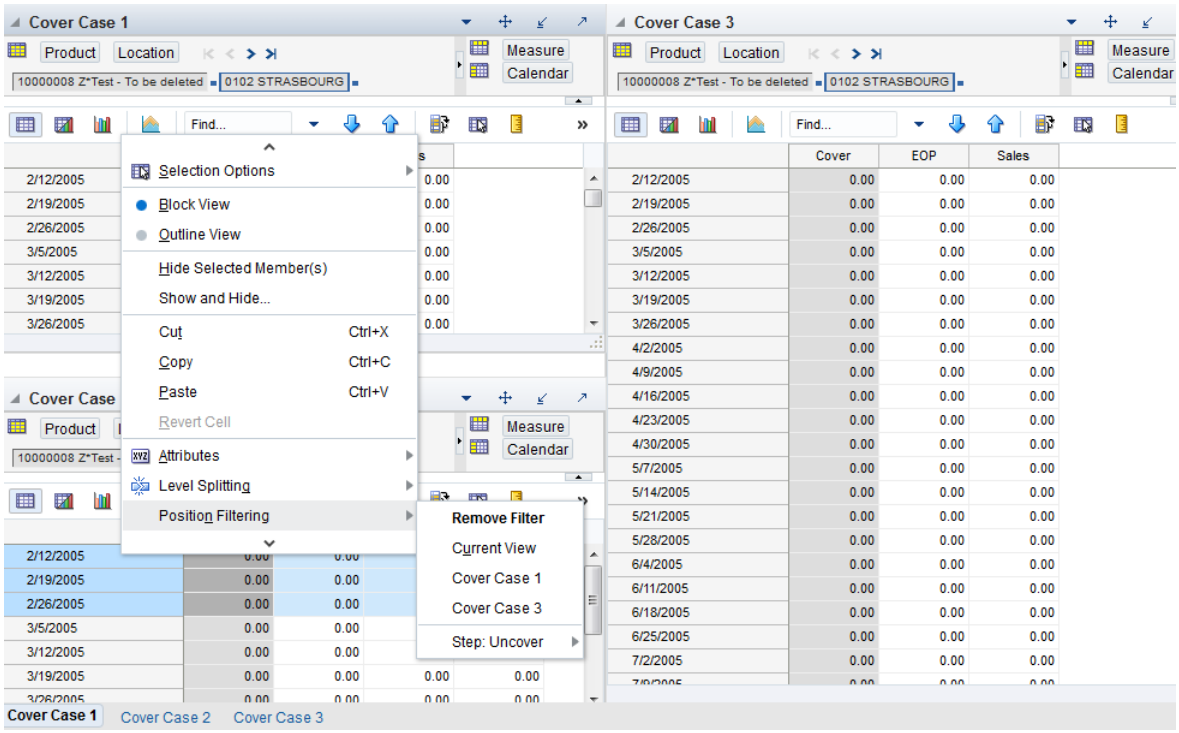

Figure 6.9: Position Filtering Stage 3

The views can further be increased to as many numbers as wanted and by configuring it into the configurable xml. From Cover Case 2, the positions are further filtered out

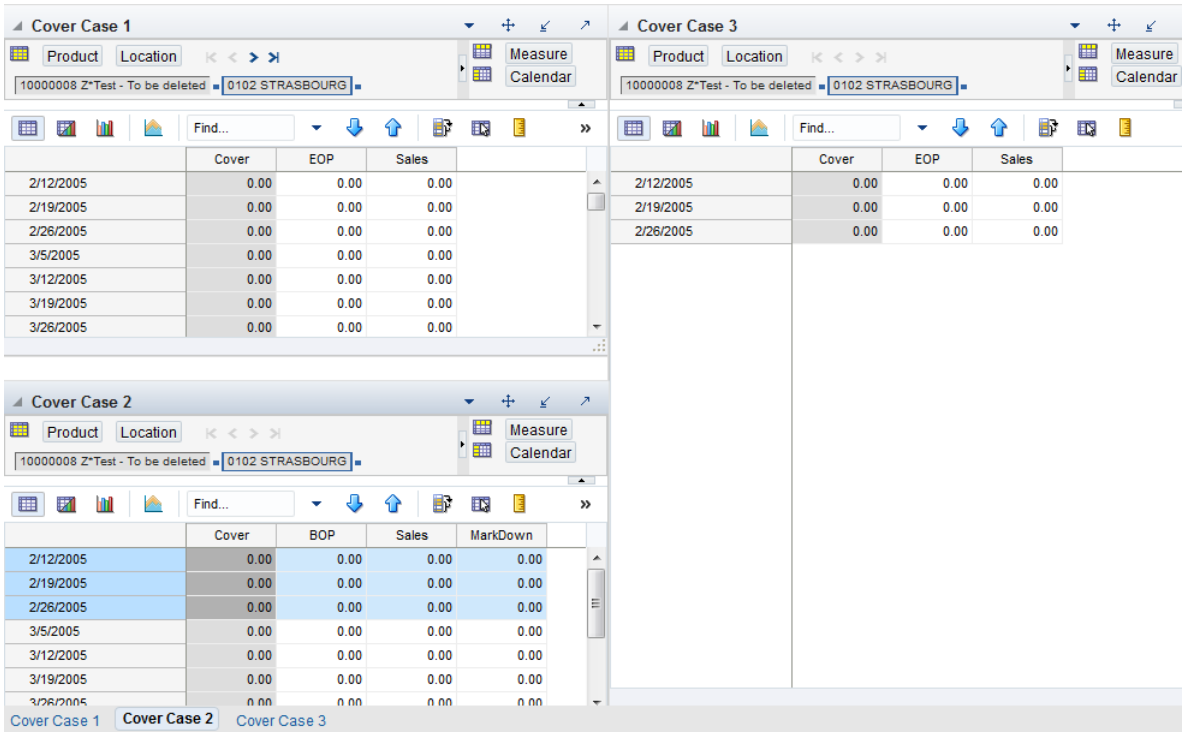

Figure 6.10: Position Filtering Stage 4

and displayed in view Cover Case 3 as shown in the fig.

6.6 Charts with Position Filtering

Position Filtering can also be applied to view charts as pictorial representation and easy inference of data. According to customer requirements, charts have been integrated as a feature to display filtered data.

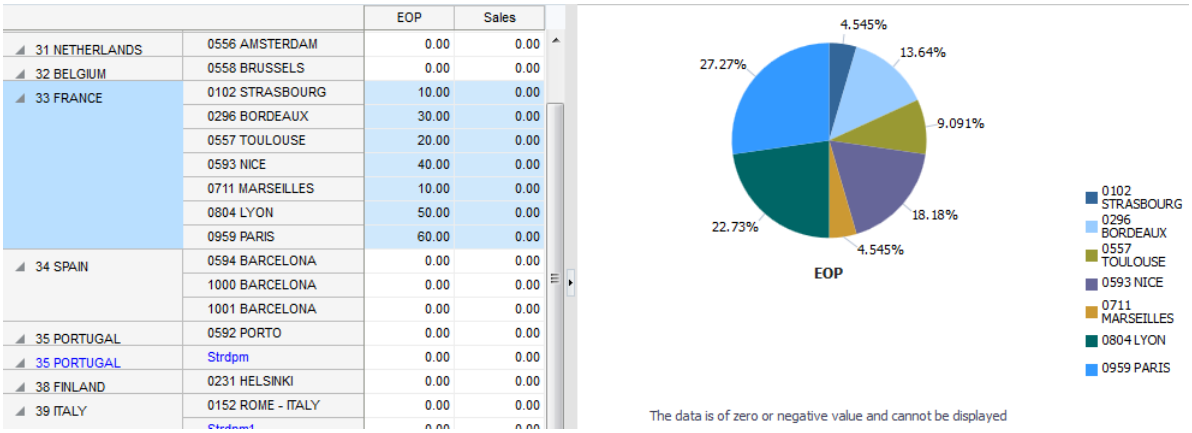

Figure 6.11: Chart before Position Filtering is applied

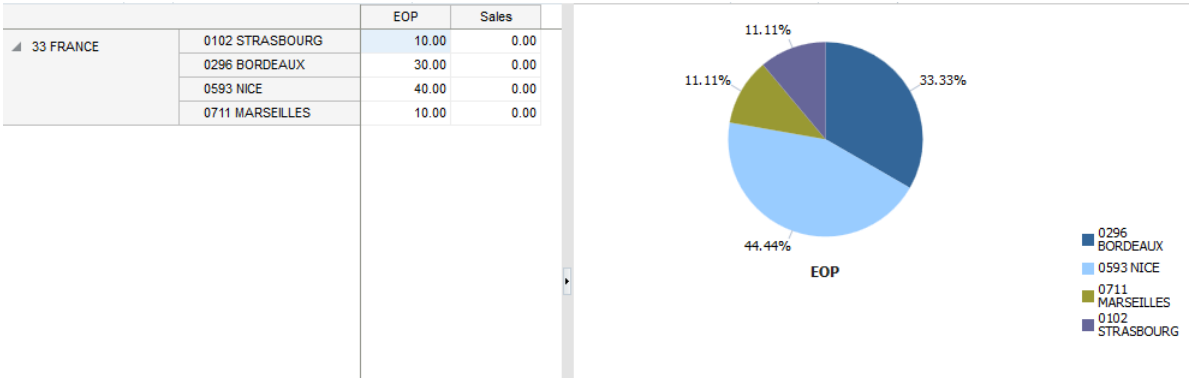

Figure 6.12: Chart after Position Filtering is applied

The filtered data can be viewed in different chart formats. For each view, the chart options are as shown in the fig. Area Chart, Bar Chart, Bubble Chart, Line Chart etc.

JET is the Oracle Javascript Extension Toolkit which provides a collection of open source JS libraries along with a set of Oracle contributed JavaScript libraries that make it as simple and efficient as possible to build applications that consume and interact with Oracle products and services, especially Oracle Cloud services.[1]

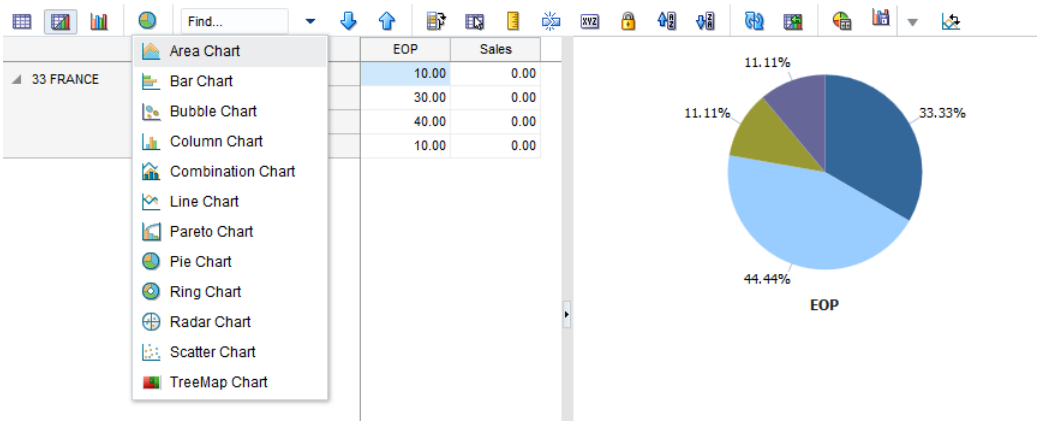

Figure 6.13: Chart Types

Chapter 7

Export Images To Excel

7.1 Introduction

Export functionality provides the user the ability to export the data in the current slice of a worksheet to Microsoft Excel or Text. With this enhancement, users now can export the images shown in pivot table (Attributes/Measures) as well. The workbook style and content are rendered using the Apache POI standard library. The exported format has now been changed to .xlsx format. SXSSFWorkbook from the Apache POI library, is created and is used to store all the workbooks pivot table information, including images. This allows for more number of rows and columns to be present in a workbook.[2]

7.2 Implementation

First of all, worksheet is created. Currently only one worksheet of data can be exported by the user, even if the workbook has multiple worksheets. For each tab/worksheet, user has to click on export option separately. For each worksheet in the workbook, the data for the all the rows are fetched and stored. Then for each row, the data for all its cells are fetched. Only first 100 rows are actually stored in a buffer at a time. When the 101st row comes, the 1st row is flushed out of the buffer. This has been to done to reduce the memory footprint of the buffer used, as there can be many rows in a worksheet.

For the current workbook, even though theres no difference between headers and content cells, header styles are rendered first containing measures and dimensions for all the rows and columns. Only after that, the individual cell styles are rendered.Header styles for dimension are rendered differently, as compared to measures, as measures may

have informal positions added to it. All the style information for header and cells are present in the Data Access layer. Style information such as font size and color, background color, font alignment, cell bordering etc. are all either defined explicitly for each cell in the PivotTableExcelExporter class, or are available in the workbook object.

The image to be exported is first converted to byte array by using IOUtils class of Apache POI library and the dimensions of the image are fetched. The actual image width and height are then used to calculate aspect ratio. Thus, The aspect ratio of the image is calculated depending upon the size of the parent cell in the worksheet. The image dimensions are then maipulated to accommodate the image in its particular column, and then row, in the worksheet. A ClientAnchor object is then created to position the image on worksheet. The X and Y co-ordinates for the first and second cell are set by using their respective row and column positions.The index of the image is stored in the SXSSFWorkbook object created earlier for that particular byte array of the image.By using the ClientAnchors object and the calculated index, the image is drawn on the worksheet using Apaches POI Drawing interface.

7.3 Apache POI Library

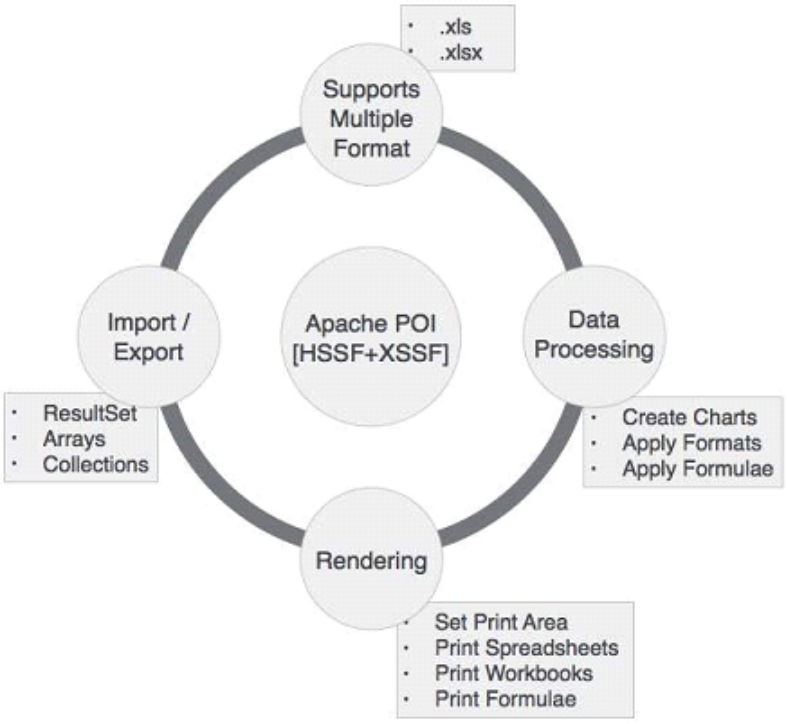

Figure 7.1: Apache POI Library[3]

Apache POI is an API that allows Java developers to create, modify, and display MS Office files. Developed and distributed by Apache Software Foundation, Apache POI is a library that can be used in Java programs to create any excel file. It can decode the data input by user, or an entire file, using its proprietary classes and methods, and then convert them into MS Office documents.

XSSF (XML Spreadsheet Format) is used to create files with .xlsx file format of MS-Excel. Its a class that implements the Workbook interface and implements both the high and low level file formats of excel. XSSFSheet is a class that provides a high level representation of excel spreadsheet. It can be found under org.apache.poi.xssf.usermodel package. Sheets are created by using the createSheet() method on the existing workbook object. XSFFRow implements the Row interface. Rows are created using createRow() method on the existing instance of sheet. Cells are then created by calling createCell() on the existing row instance. SXSSF API is built on top of existing XSSF class and provides a lower memory footprint. SXSSF is a streaming version of XSSF, having APIs similar and compatible to that of XSSF. Its used to create large excel sheets when low heap space needs to be used. While XSSF provides access to all rows in the document, SXSSF limits access to the number of rows in sliding window, thus utilizing lower memory. The access windows size can be defined at workbook construction time, to hold the specified number of rows in heap space. The default window size is 100. When the maximum value is reached, heap memory becomes full and the subsequent creation of an additional row, flushes out the row with the lowest index from the access window, onto the disk. Older rows that are no longer in the window become inaccessible. Since SXSSF is a streaming API, there are few obvious limitations when compared to XSSF. The number of rows accessible at a point of time are limited. Due to this, formula evaluation is not supported for large data sets.[1]

7.4 Export Functionality Workflow

The Export options dialog takes the user input before exporting the data to excel. The entered settings are used for exporting data to excel. A new option Export Images check box is now available to the user. By, default Export Images check box is checked, i.e. up on exporting, the images are exported and the default size will be Default. Theres also a Large size option available to the user in the drop-down menu.

Default size results in the Row height and Column width defaulted to the printexport.default.cell.height and printexport.default.cell.width properties available in properties file. Default size picks the image corresponding to printexport.default.image.size.id property. Large size results in the Row height and Column width defaulted to printexport.large.cell.height and printexport.large.cell.width properties available in properties file. Large size picks the image corresponding to printexport.large.image.size.id property.

In case of Default or Large, the cell width and cell height fields remain disabled. Checking the Custom checkbox enables these fields. Users can override the default cell width and cell height. The height and width are defaulted to the corresponding properties from properties file. Custom picks the image corresponding to the chosen size.

The aspect ratio of the images exported is maintained when the Cell Width and Cell Height are given by the user. For example, say the actual image size is 347*400, and the user gives the size as 100*100, then the exported image size will be 87*100, maintaining the aspect ratio of the original image. The image repository is configurable. If theres no image for the corresponding size, then a default image is displayed. If the user decides to export only text, then image label is exported to excel.[2]

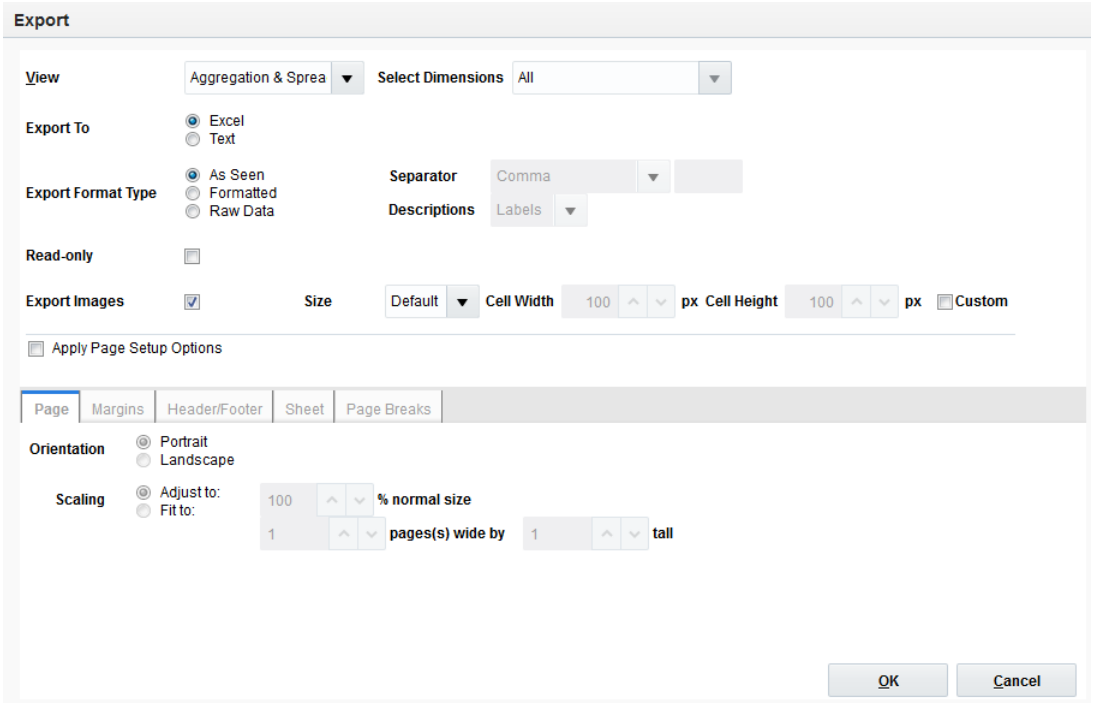

Figure 7.2: Export Image to Excel

Chapter 8

Edit View Overlay

8.1 Introduction

The positions in a workbook can be edited using Edit View Overlay(EVO). For a specific workbook, we can change the positions by clicking on the dimensions required and selecting any positions from available positions.

It also allows to edit dimensions after a workbook is created. The intersection level which we see the information can be edited. The measures visible in the pivot table can be made visible/invisible using show/hide. [2]

8.2 Edit View

All the dimensions i.e. Calender, Product, Location and Measure can be edited.

8.3 Edit Product

The positions which we want to see on the pivot table can be selected/deselected. In the figure below, Sku is the base intersection. Except Sku, we select Color. The positions of Color hierarchy can be seen. We don't select all positions, we deselect last four positions. The change in pivot table is as shown in the figure. Using the'+' icon, we can add dimension along a particular axis, in this case the Y-Axis.[2]

8.4 Edit Location

In order to change Location positions, we first add Location on Z-Axis. The dimension Location appears on Y-Axis on the pivot table. We can select/deselect the positions.

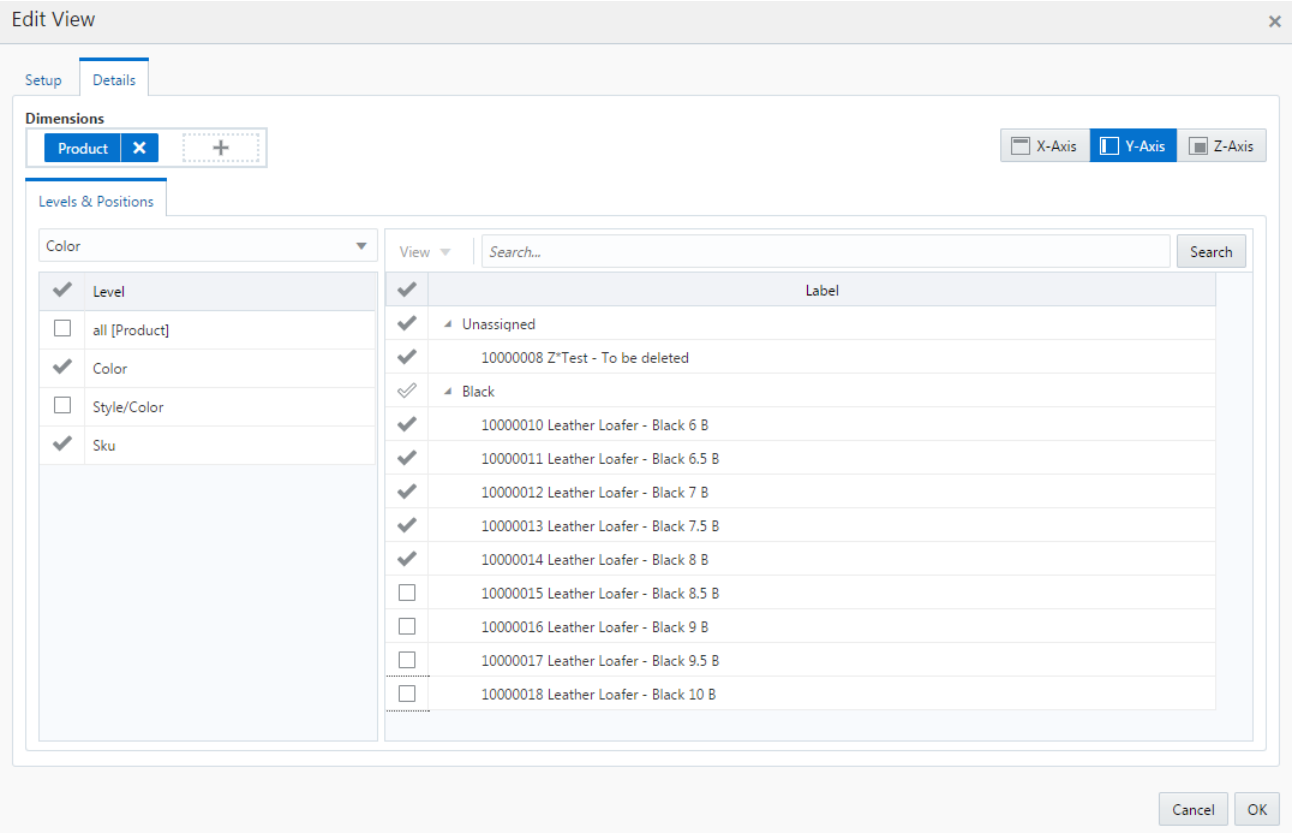

Figure 8.1: EVO Product 1

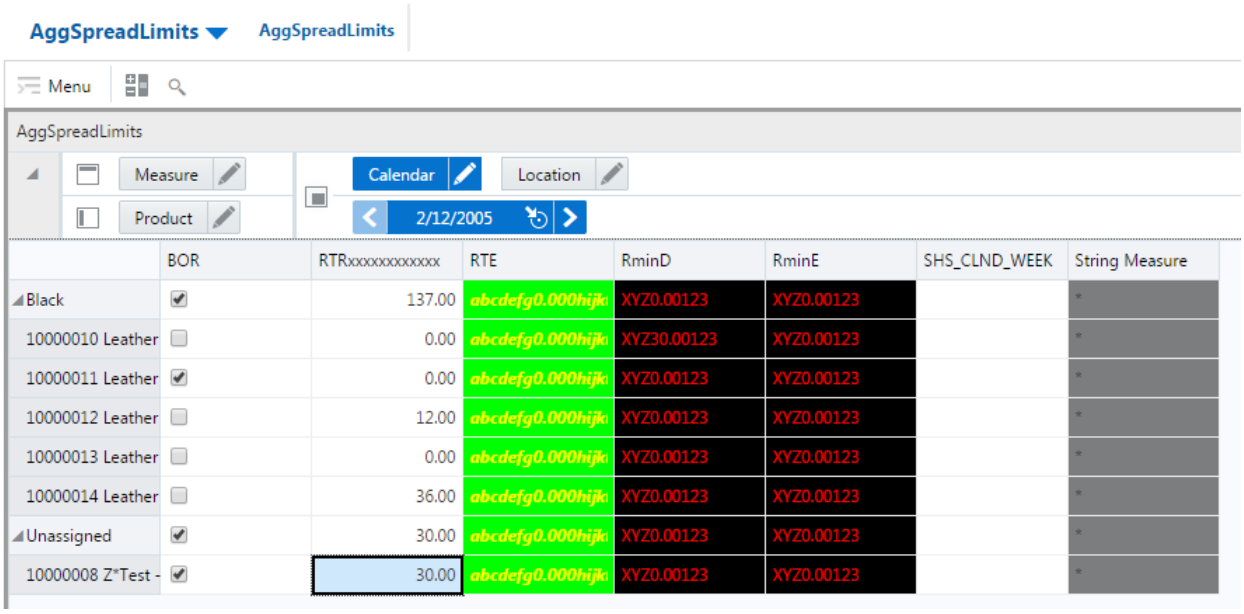

Figure 8.2: EVO Product 2

Using the'+' icon, we can add dimension along a particular axis, in this case the Z-Axis.[2]

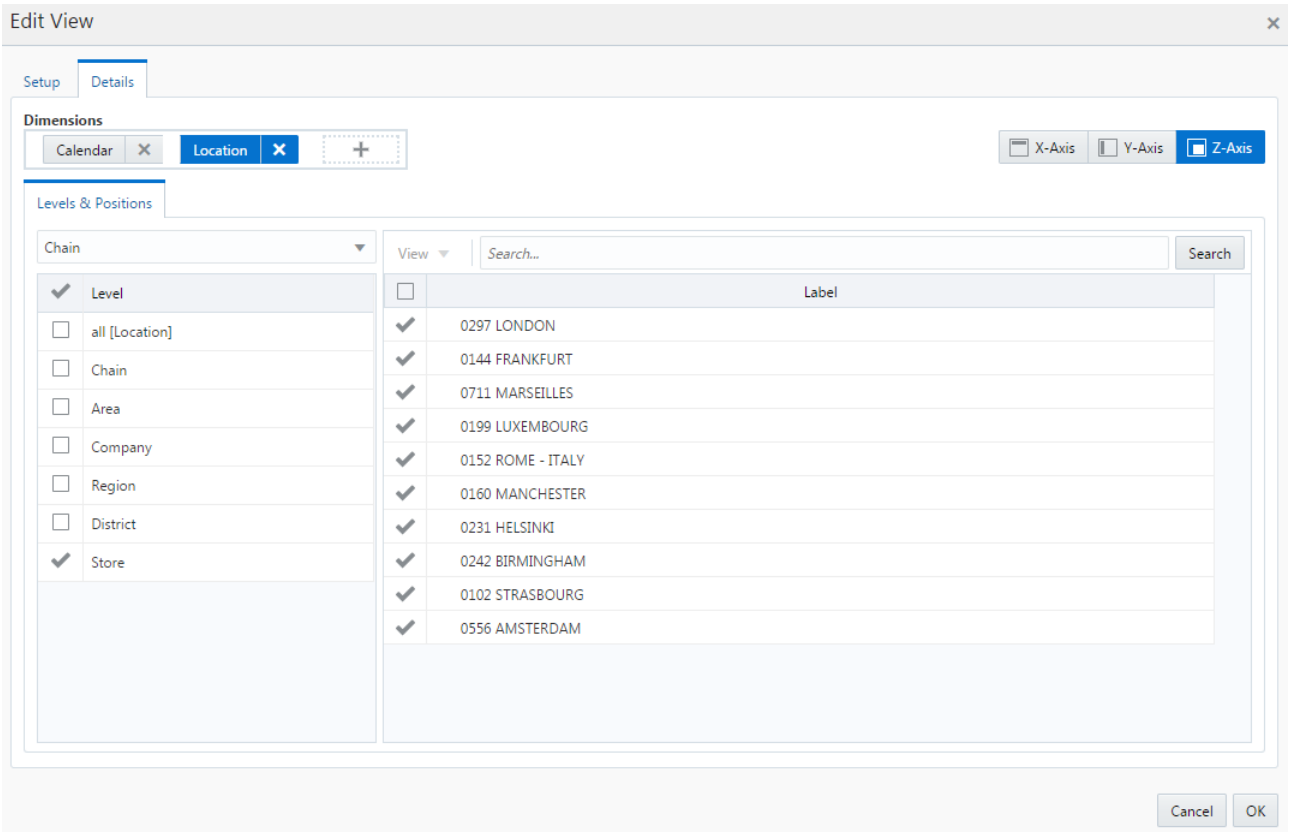

Figure 8.3: EVO Location 1

8.5 Edit Calender

The positions which we want to see on the pivot table can be selected/deselected. In the figure below, Week is the base intersection. Except Week, we select Month. The positions of Month hierarchy can be seen. We don't select all positions, we deselect last four positions. The change in pivot table is as shown in the figure. Using the'+' icon, we can add dimension along a particular axis, in this case the Z-Axis.[2]

8.6 Edit Measures

We can show/hide measures on pivot table. The measures can also be moved from their positions i.e. up, down, first or last. The corresponding changes will reflect in the pivot table. Later we can modify these changes too. A specific order in which measures are saved is called a Measure Profile. The profile which is inbuild is Default Profile where in all measures are selected by default. [2] As shown in the fig, measures RminD and String Measure are deselected. The profile name is saved as Test. When this dialog is closed,

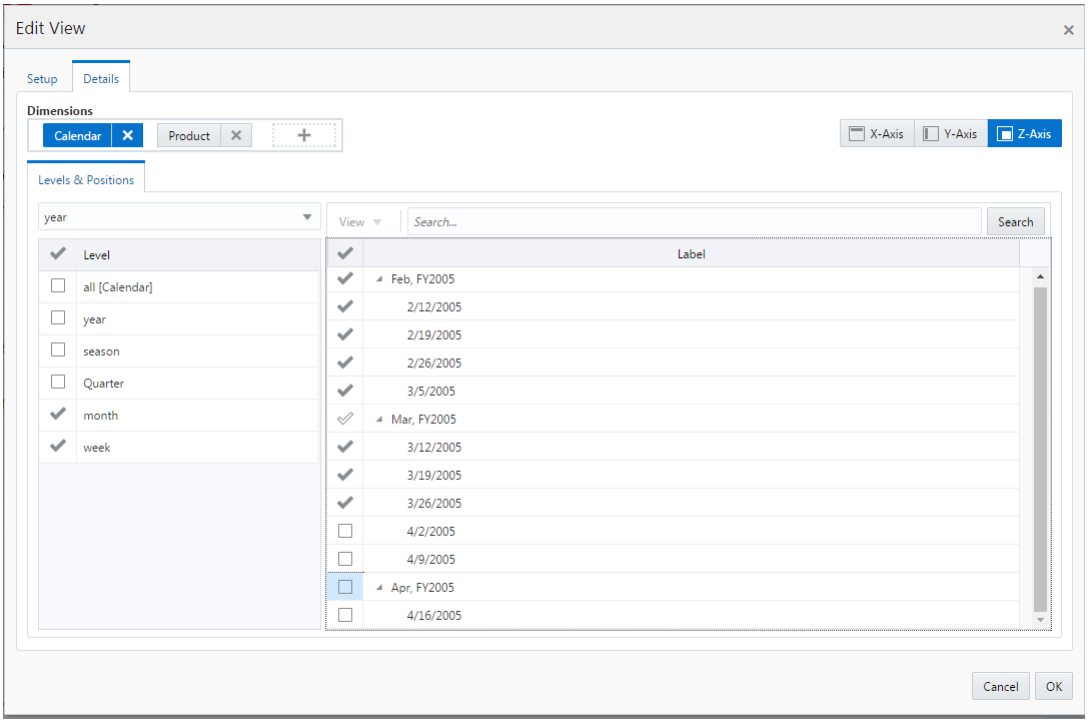

Figure 8.4: EVO Calender 1

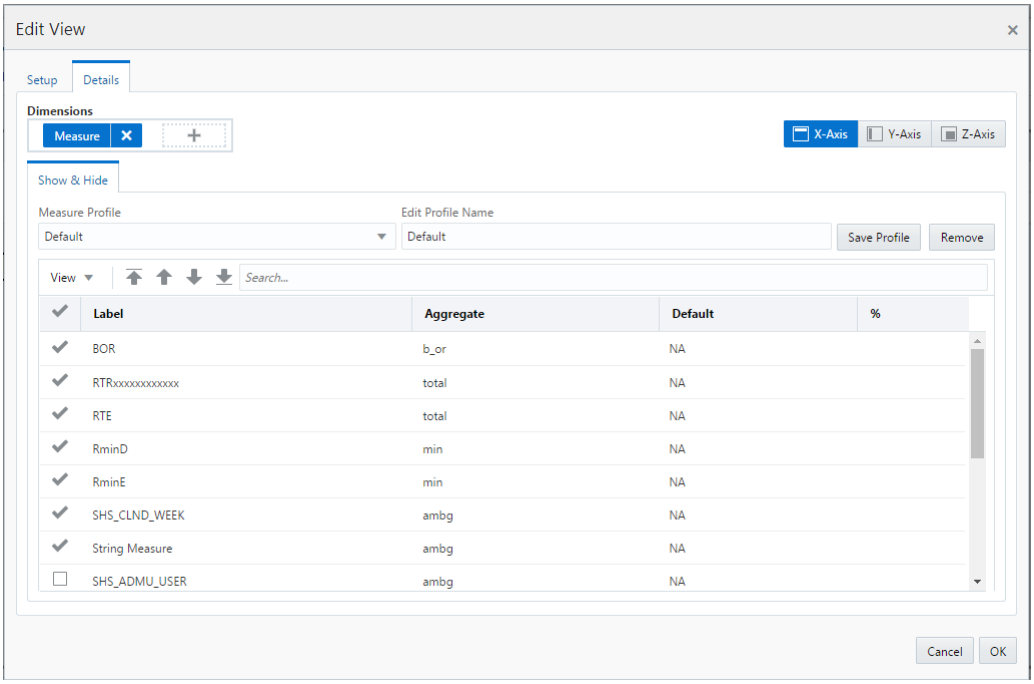

Figure 8.5: Default Measure Profile

changes will be reflected on pivot table. Only measures selected for Test profile will be visible.

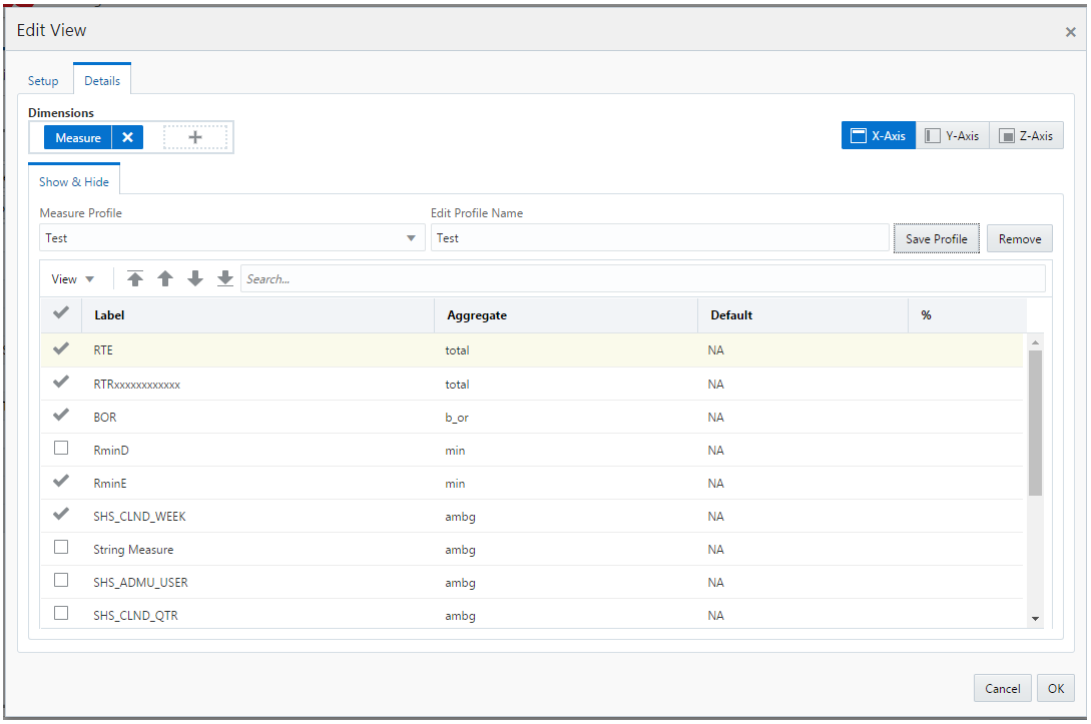

Figure 8.6: Measure Profile Test

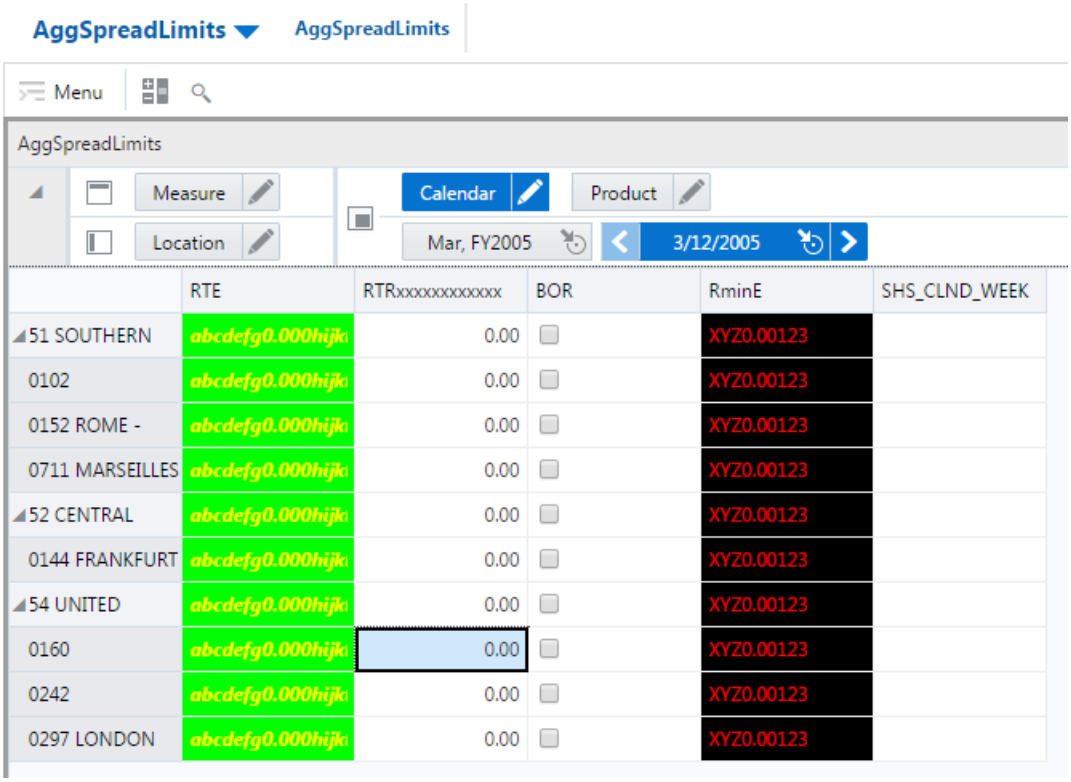

Figure 8.7: Pivot table with Profile Test

Chapter 9

Synergy Automation

9.1 Introduction

Oracle Synergy software provides regression and functional test automation support for acceptance testing and test case development. Synergy uses a keyword and scripting interface with a GUI. Synergy uses keyword based testing approach where test cases are written in a tabular format. We can access every element in the screen by using Xpath queries and perform actions like mouse click or keyboard events on the same.

9.2 Automation Scripts for OAT

Any time RPAS and FC needs to be released; FC undergoes a smoke testing cycle. We have introduced the smoke testing automation scripts using synergy tool for OAT, Dynamic Hierarchy, Position Filtering, Export To Excel and Edit View Overlay. Synergy uses robot framework to capture screens and objects in it.This allows us to eliminate repetitive manual testing every time a sprint will end. Also, we have tested them on iOS, Android Tab and Windows 10 OS with touch surface using Surface Pro 4.

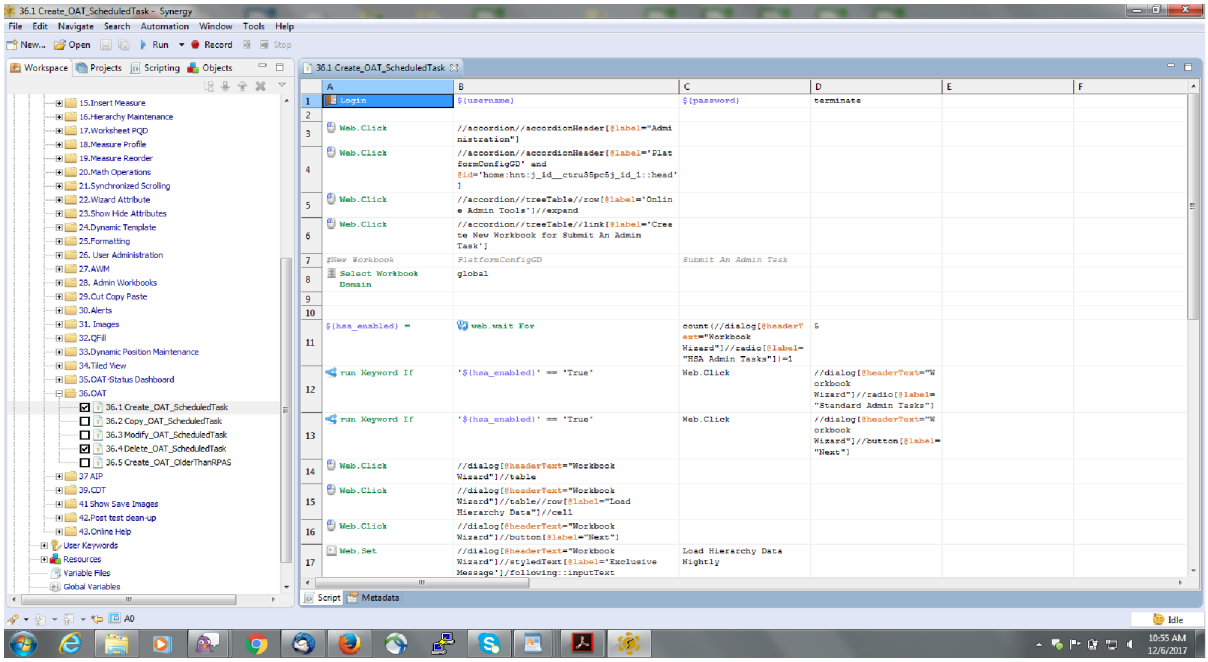

Figure 9.1: Synergy Test Case example for OAT tasks

Chapter 10

Conclusion and Future Work

10.1 Conclusion

- With development of OAT, the users can now submit tasks and run batch processes for multiple solutions and their respective domains by logging onto the client only once. Earlier, the user had to log on to each solution individually and submit tasks. To allow administrators to do this in cloud environment where they don't have direct access to back-end servers, Admin Tasks Manager provides an interface allowing authorized users to launch back-end processes from Client itself.

- For rolling up to a particular level within a dimension, previously, the whole workbook had to be re-created. To avoid this overhead of recreation of the workbook, Dynamic Hierarchy feature is introduced, where even if the levels attributes changes, the user can successfully roll-up to the required level in an existing workbook.

- When a workbook is created initially, a large amount of data, or positions are included. But when user needs to perform operations on only a set of data, it was cumbersome. The Position Filtering feature helps segregate only a set of positions on which the user can perform the required action. This avoids the overhead of data management.

- Export functionality provides the user the ability to export the data in the current slice of a worksheet to Microsoft Excel or Text. With Export Images To Excel enhancement, users now can export the images shown in pivot table (Attributes/Measures) as well.

- The positions in a workbook can be edited using Edit View Overlay(EVO). For a specific workbook, we can change the positions by clicking on the dimensions required and selecting any positions from available positions.

10.2 Future Work

- Integration between Admin Task Manager and Status Dashboard has to be done. In future, Task Name will be converted to a link and clicking on it will provide the user an option to open the task and to modify or copy it.

- Support for MultiSolution Architecture The current Cloud FC only connects one solution at a time from one machine. In future, Cloud FC will be enhanced to support the above feature.

- Export functionality doesnt work on Tiled View workbooks, where the data and images are represented in the form of tiles. Also, instead of exporting each sheet of a workbook in individual .xlsx, a single file shall be created.

- Future enhancements as per functional requirements and bug fixes in the existing product.

Bibliography

[1] "Oracle internal documentation."

[2] "Oracle rpas fusion client user guide (url: http://docs.oracle.com/)."

[3] "Oracle retail predictive application server guide (url: http://docs.oracle.com/)."

[4] "Oracle rpas fusion client administrator guide (url: http://docs.oracle.com/)."