Oracle Fusion Financial Approval Workflow - Release 12

Overview

The document provides complete and detailed information regarding various approval features offered by Fusion Financial. This includes nomenclature, navigation, different configurations and other aspects of configuring approval rules.

Worklist: Notifications and Approvals

All approval configurations are done through BPM Worklist application. There are two navigation path to access worklist application from the FA home page.

Bell Icon

1. Click on the Bell Icon
2. More Details
3. Financials

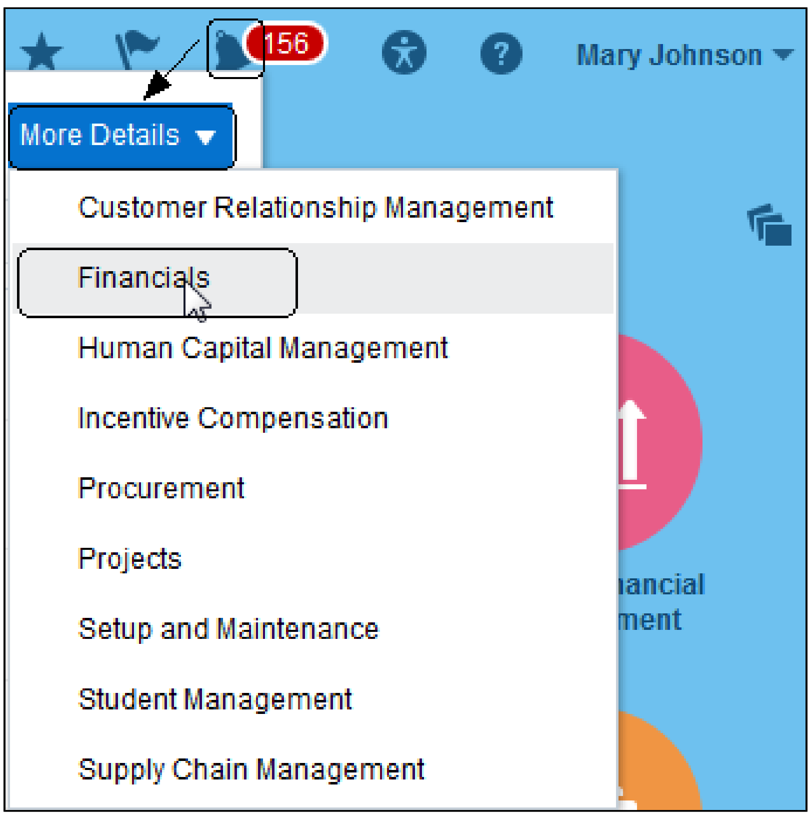

Fig 1

1. From the BPM Worklist application expand the user panel and select *Administration*

    - User will need SOA_BPM_WORKFLOW_ADMINISTRATOR_DUTY role to access this. By default this is given to FUN_FINANCIAL_APPLICATION_ADMINISTRATOR_JOB.

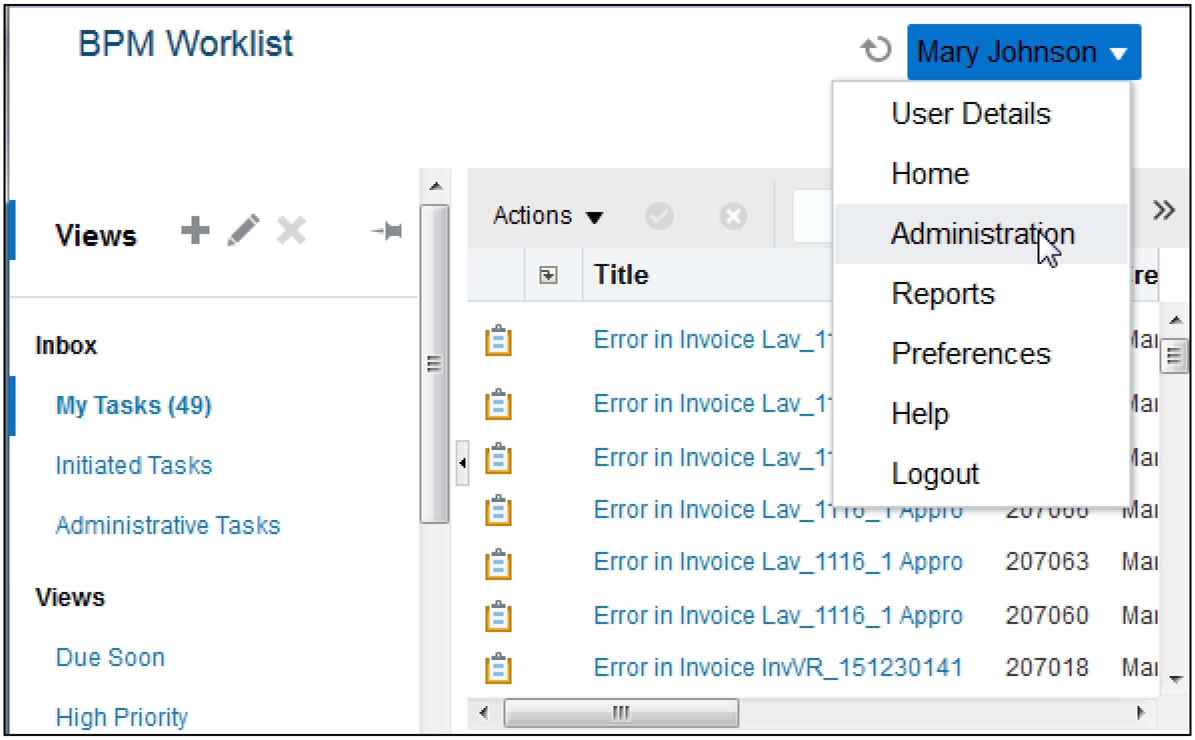

Fig 2

1. From Administration page, click *Task Configuration* tab
   - A separate *Approval Group* tab provides the ability to configure approval groups

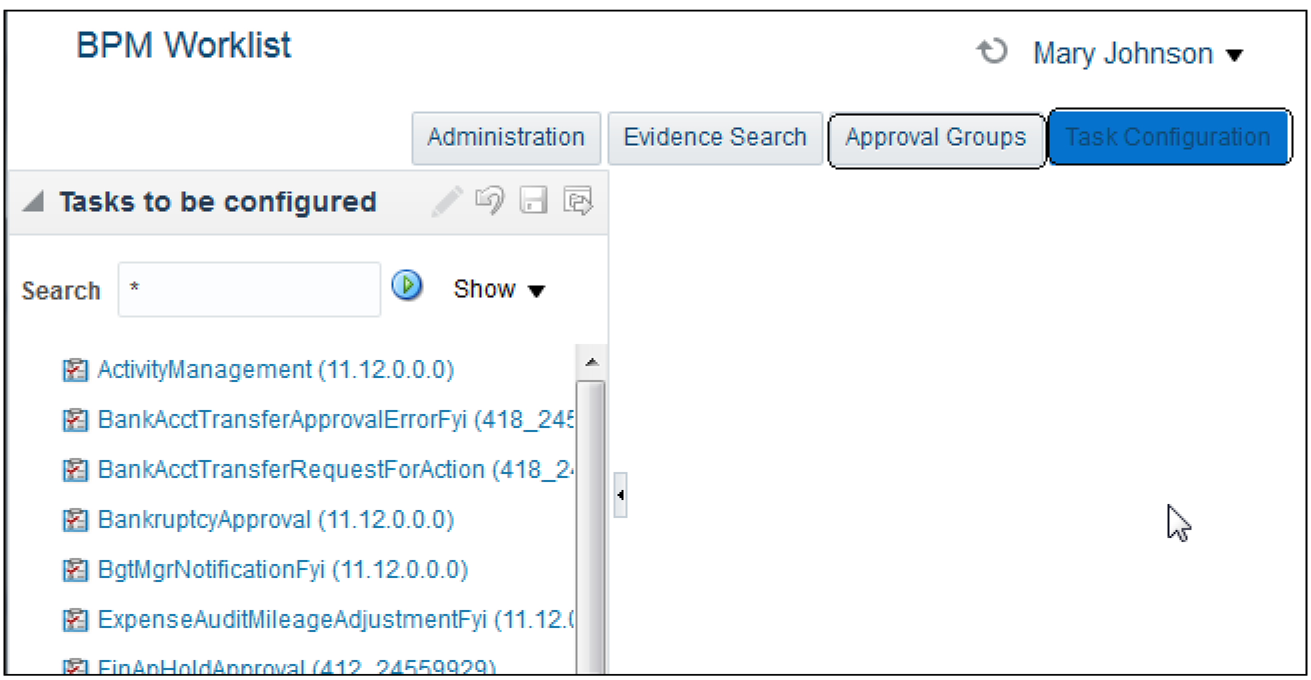

Fig 3

Setup and Maintenance

1. Search for *Define Approval Management for Financials*

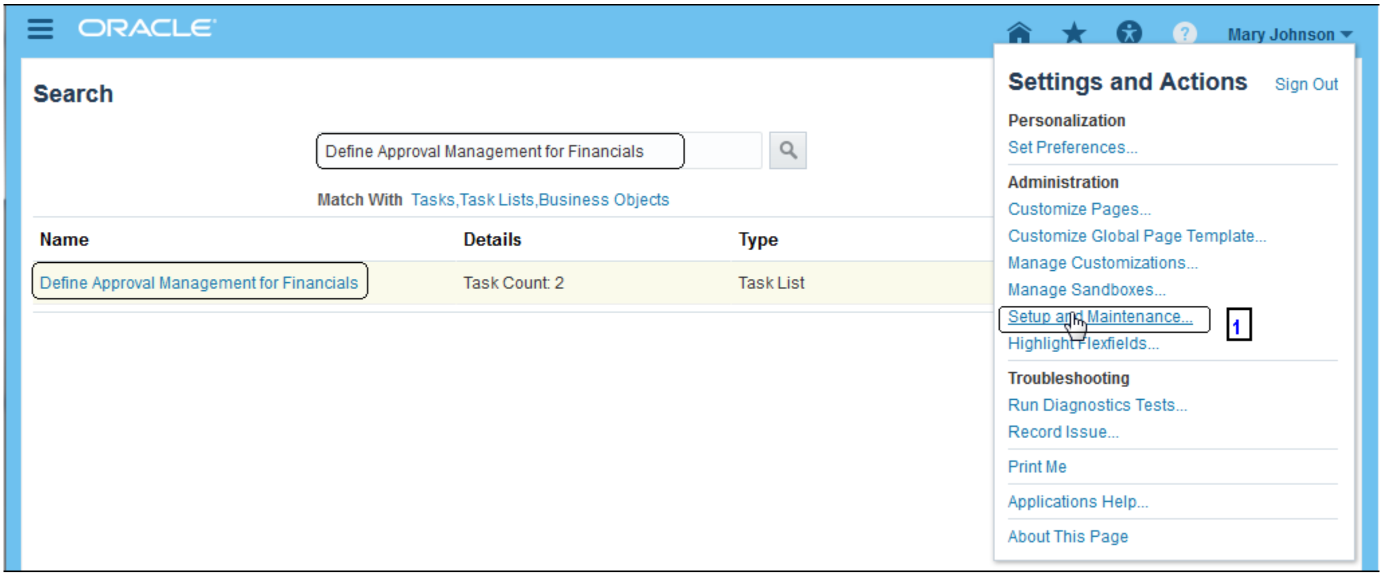

Fig 4

1. Click *Manage Task Configuration for Financials* to access Task Configuration tab of BPM worklist application

Fig 5

- Click *Manage Approval Groups for Financials* to access Approval Groups tab of BPM worklist application

Task Configuration

Task Configuration is a part of Worklist Application that enables business users and administrators to review and modify approval rules that were predefined by the workflow designer. These predefined rules can be changed in accordance with corporate approval policies. At any given instance there could be only one serving version or default version of Human task flow with other revisions being active but in retired state. Default version will serve the new request while retired version will keep on serving the requests originated on that version till they are complete. By default user will be presented with default version of human task flow but they can view ALL active versions by selecting *Active versions* under *Show* dropdown.

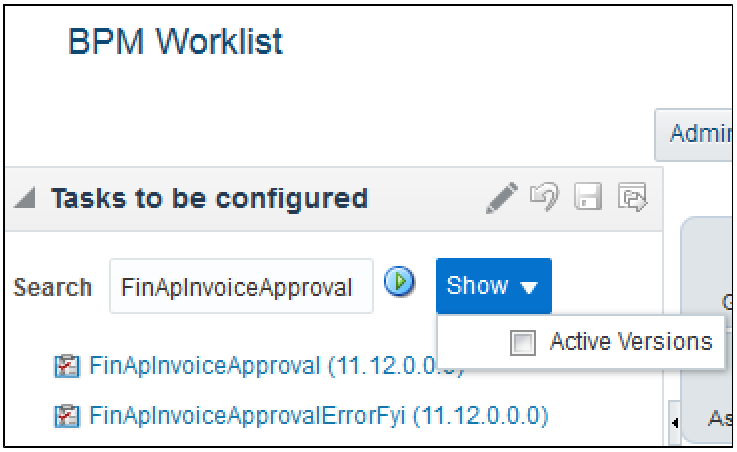

Fig 6

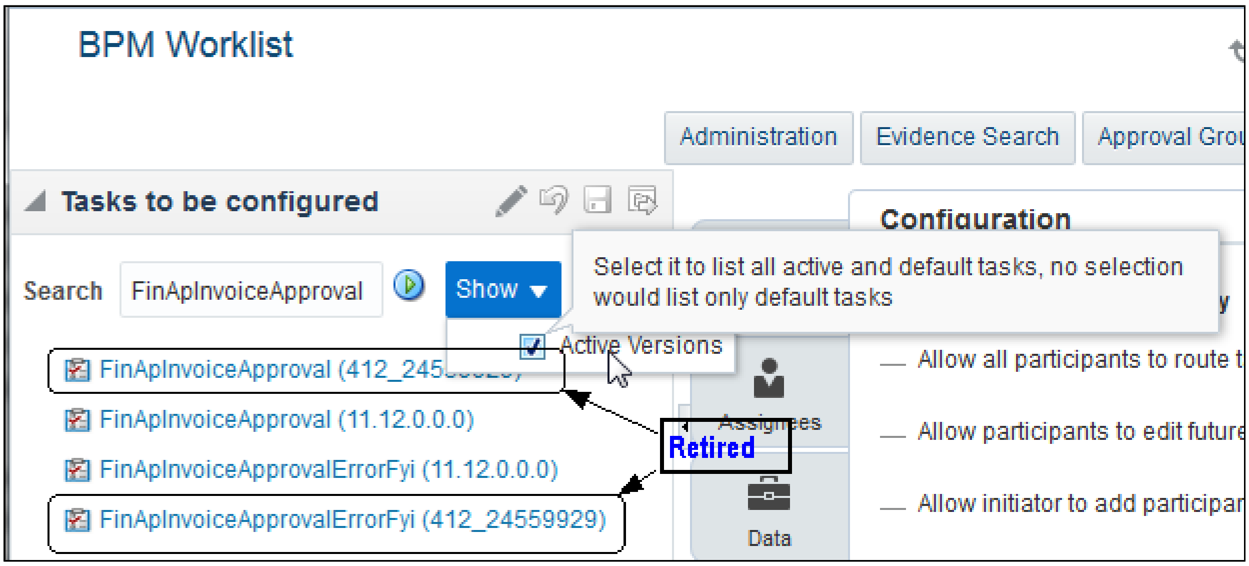

Fig 7

The Human Task editor enables user to configure the following sets of properties:

General

Enables user to define basic information such as the title, description, priority and owner.

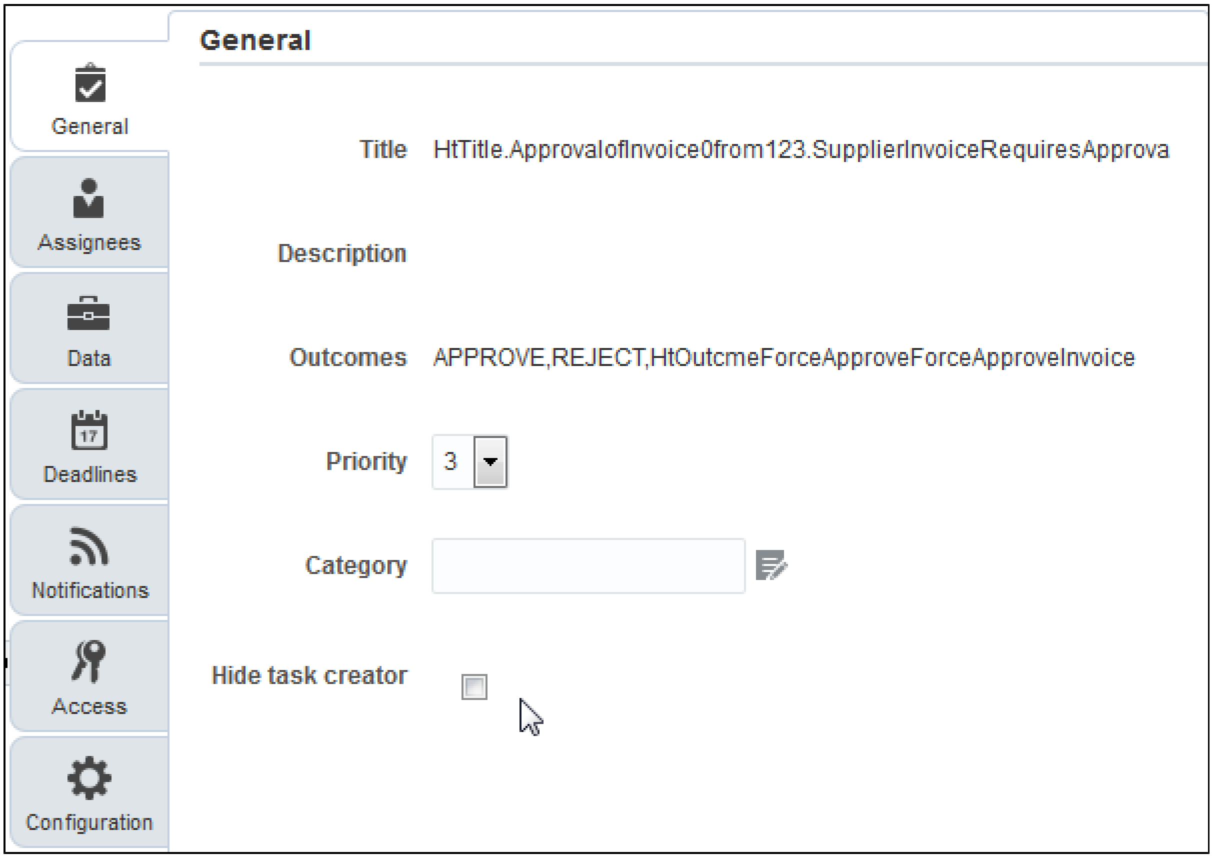

Fig 8

Checking the *Hide task creator* will prevent task creator name to be shown in approval notifications.

Data

Enables users to review the message elements that compose the structure of the task payload. Customer are not expected to edit this data.

Deadline

Enables users to specify the duration and expiration of a task.

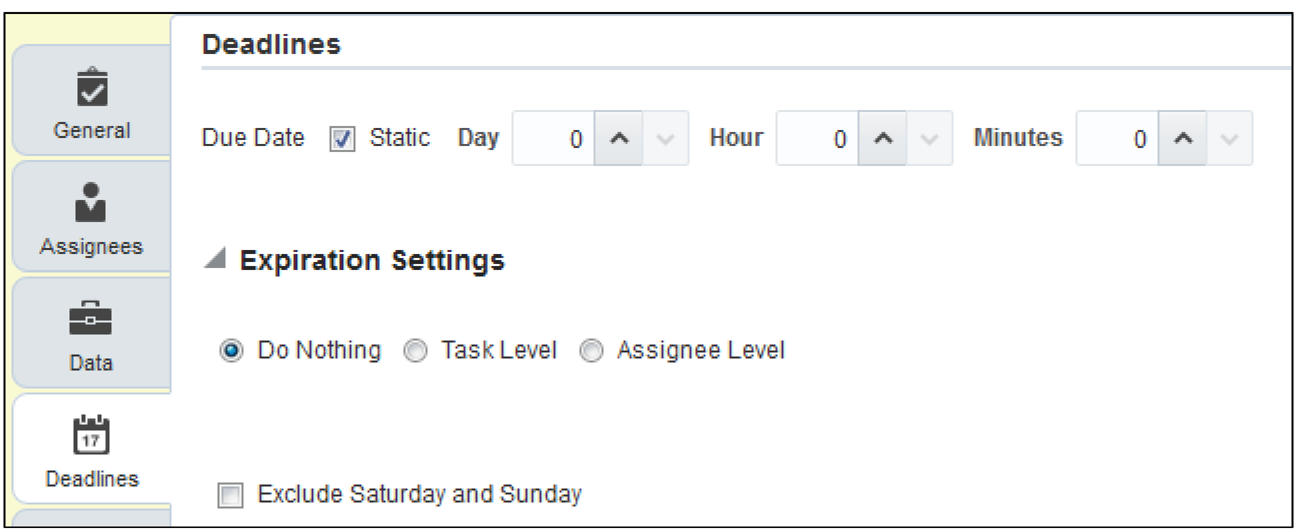

Fig 9

Due Date

It indicated the date when the approval task is due for action. Please note, this due date doesn't mean that the approval task is expired. It is just an indicator to remind approver that they have to act by certain time. System doesn't prevent them from acting even after the due date.

Exclude Saturday and Sunday

Marking this checkbox will excludes Saturday and Sunday in due date and expiration / escalation date computation and only weekday will be included.

Expiration Settings

User can specify the expiration policy at either *task level*, at individual *assignee level* or skip it entirely. Within the expiration setting, there are following options:

Escalate

Task can be escalated if a user does not respond within the allotted time. For example, if you are using the escalation hierarchy configured in your user directory, the task can be escalated to the user's manager the first time. If the user's manager still doesn't take the appropriate action in the specified duration then the approval task will be further escalated till it reach the maximum escalation level or the highest approver title. An escalated task will remains in the user inbox even after the task has been expired but user won't be take any action on the expired task.

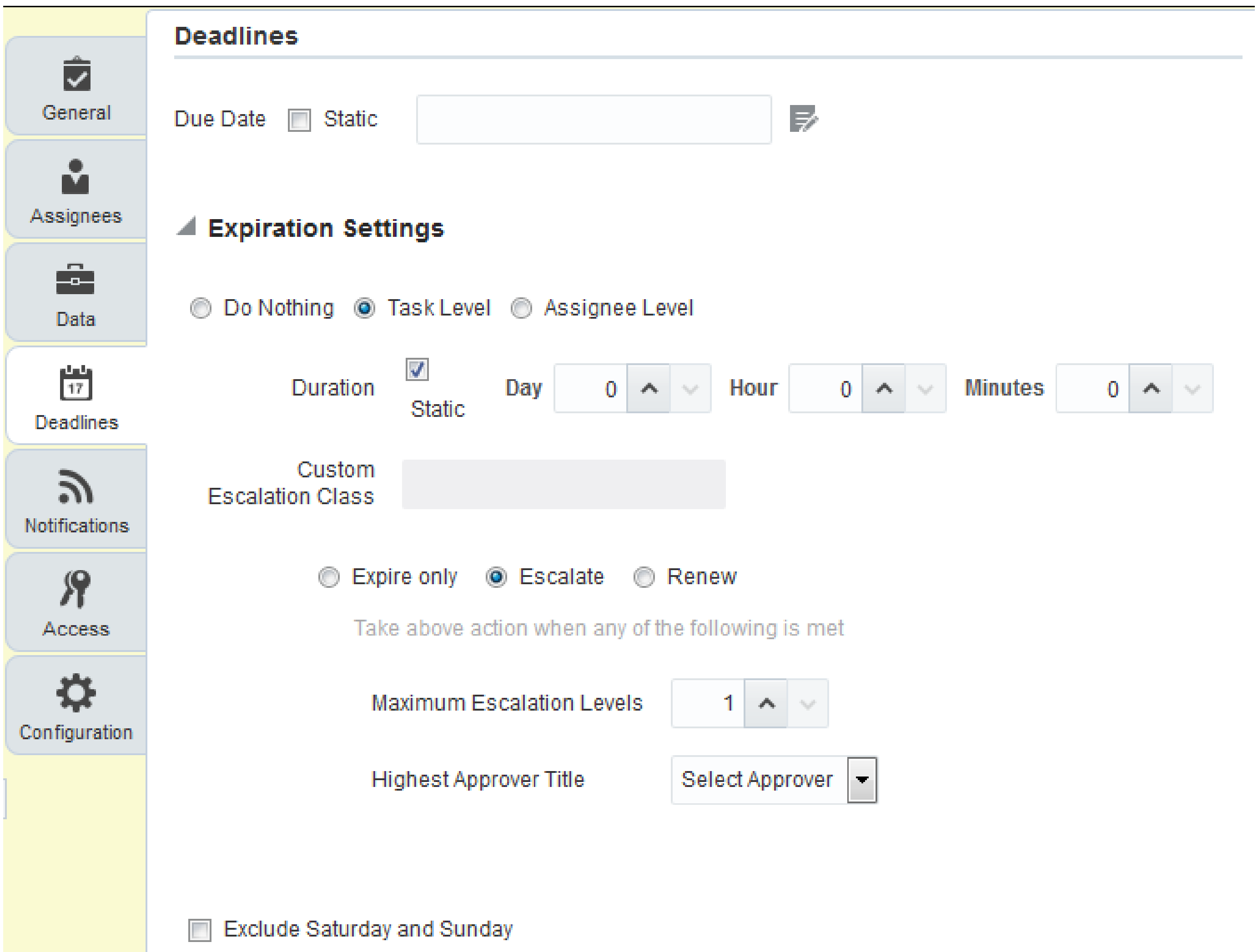

Fig 10

Escalation policy can be set to trigger after specified duration from the task creation time. This could be static or user can define it dynamically using various task attributes in expression builder.

- Maximum Escalation Levels

Number of management levels to which to escalate the task. This field is required. This traverse up the supervisory hierarchy.

- Highest Approver Title

The title of the highest approver (manager, director, or CEO). These titles are compared against the title of the task assignee in the corresponding user repository. This field is optional. This option is not available for the individual assignee escalation.

Expire

This gives user an option to expire the task after certain duration.

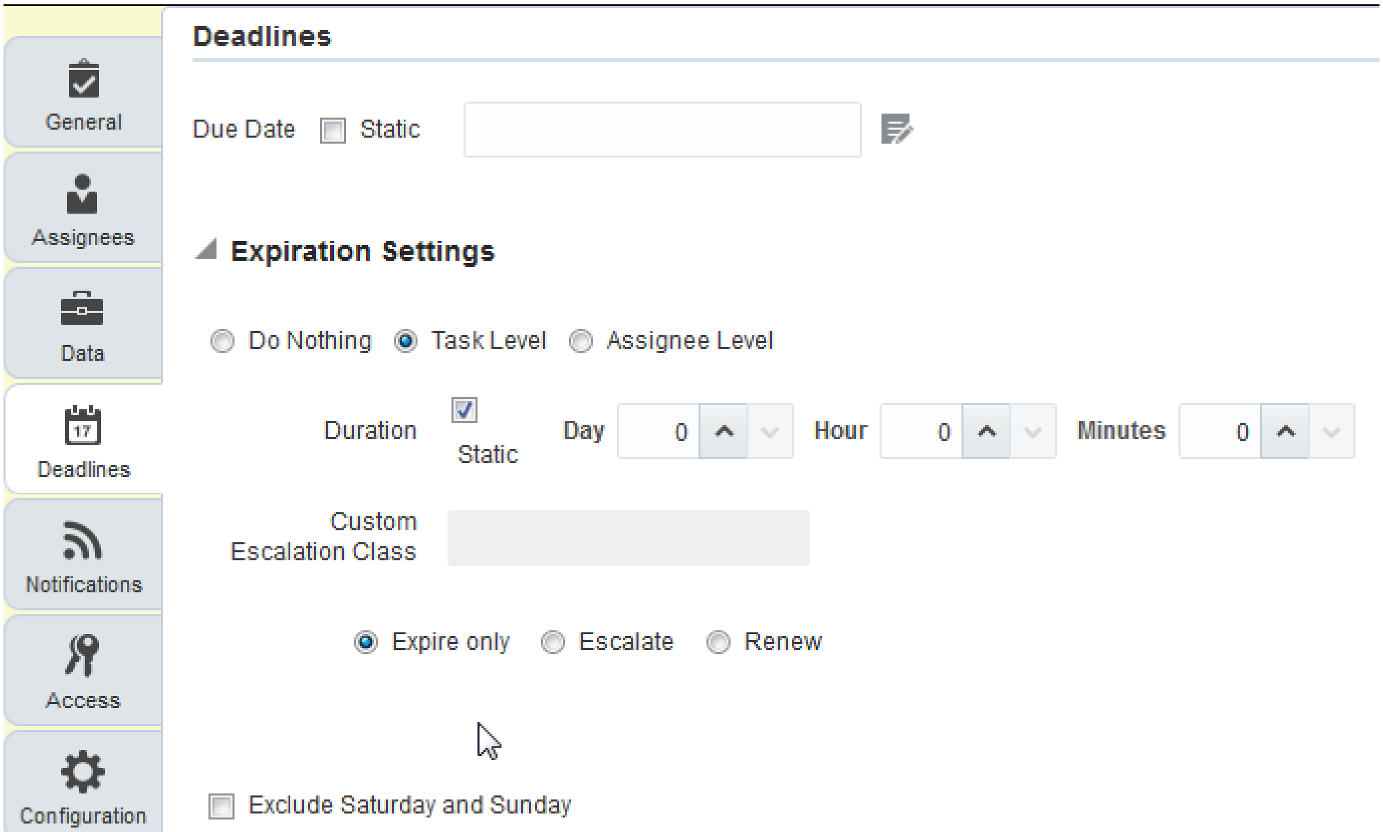

Fig 11

Renew

This gives users an option to renew the task after certain duration. User can also specify maximum number of time the task should be renewed.

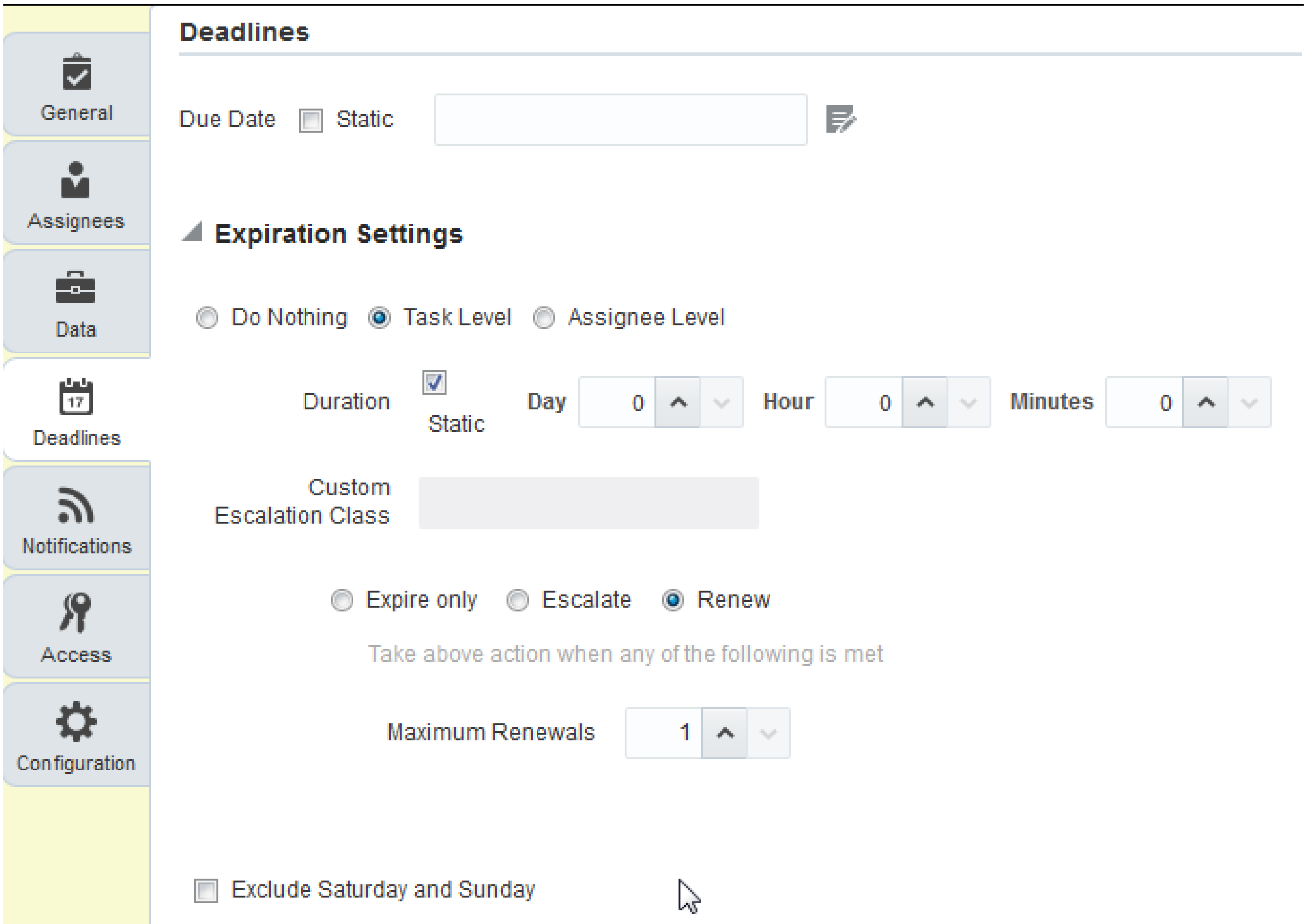

Fig 12

Notification

The approval notification helps informing the respective approver to review the business transaction and take action on them. Notifications can be sent through email, voice message, instant message SMS or inbuilt application bell notification. By default only email and inbuilt application bell notification are configured. Admin user can choose which channel they want to use to receive approval notification. This is setting is done at the BPM worklist Application Preference screen, as show below

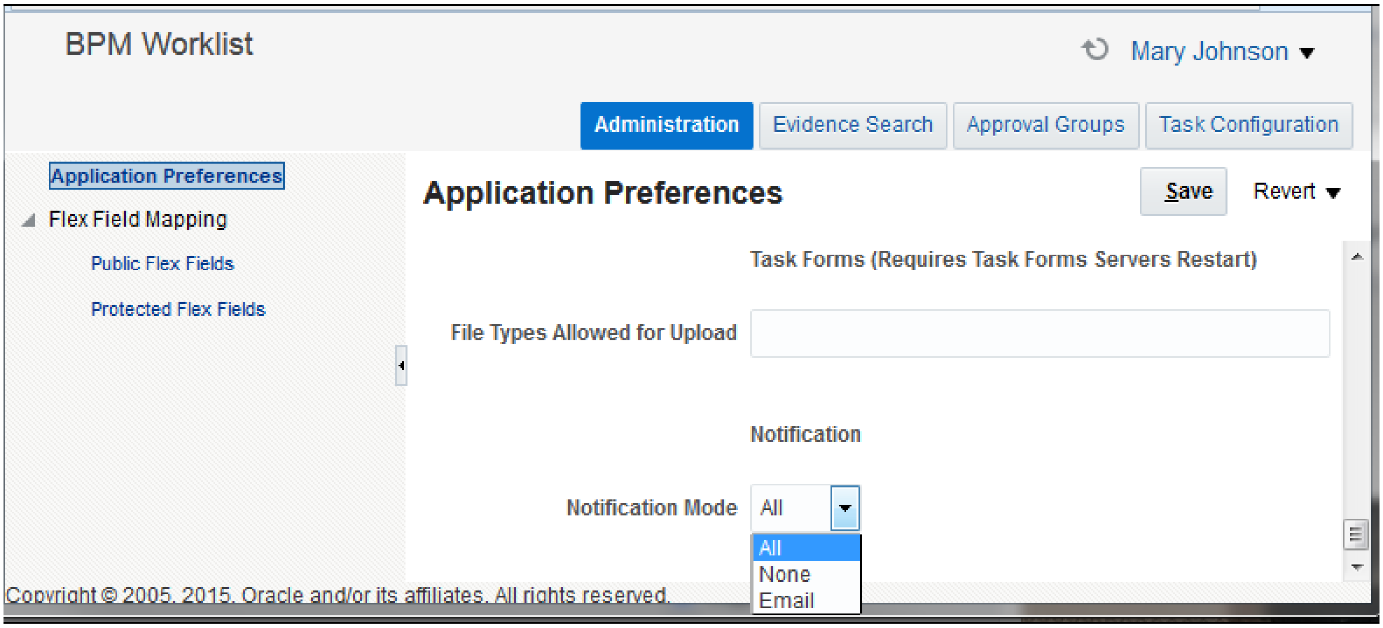

Fig 13

Notification Mode

- All
    - Both email and bell notifications
- None
    - Neither email nor bell notifications
- Email
    - Only email and no bell notifications

Notifications indicate when a user or group is assigned a task or informed that the status of the task has been changed. Notifications are sent to different types of participants for different actions. Notifications are configured by default with default messages. For example, a notification message is sent to indicate that a task has completed and closed.

Managing e-mail Notifications includes the following -

- Enable the notification - As mentioned below, click plus(+) to add new notification entry and choose the task status and recipient of the notification.
- Disable the notification - Select the appropriate action and click on the delete button to disable the notification for the specific party.

In the *more* section, users can allow actions on notifications and enable attachments to be sent with the notifications. Allowing actions will enable users who receive notifications to be able to approve/reject via email. If an attachment is present, the option allows to send the attachment via the email so that it can be reviewed.

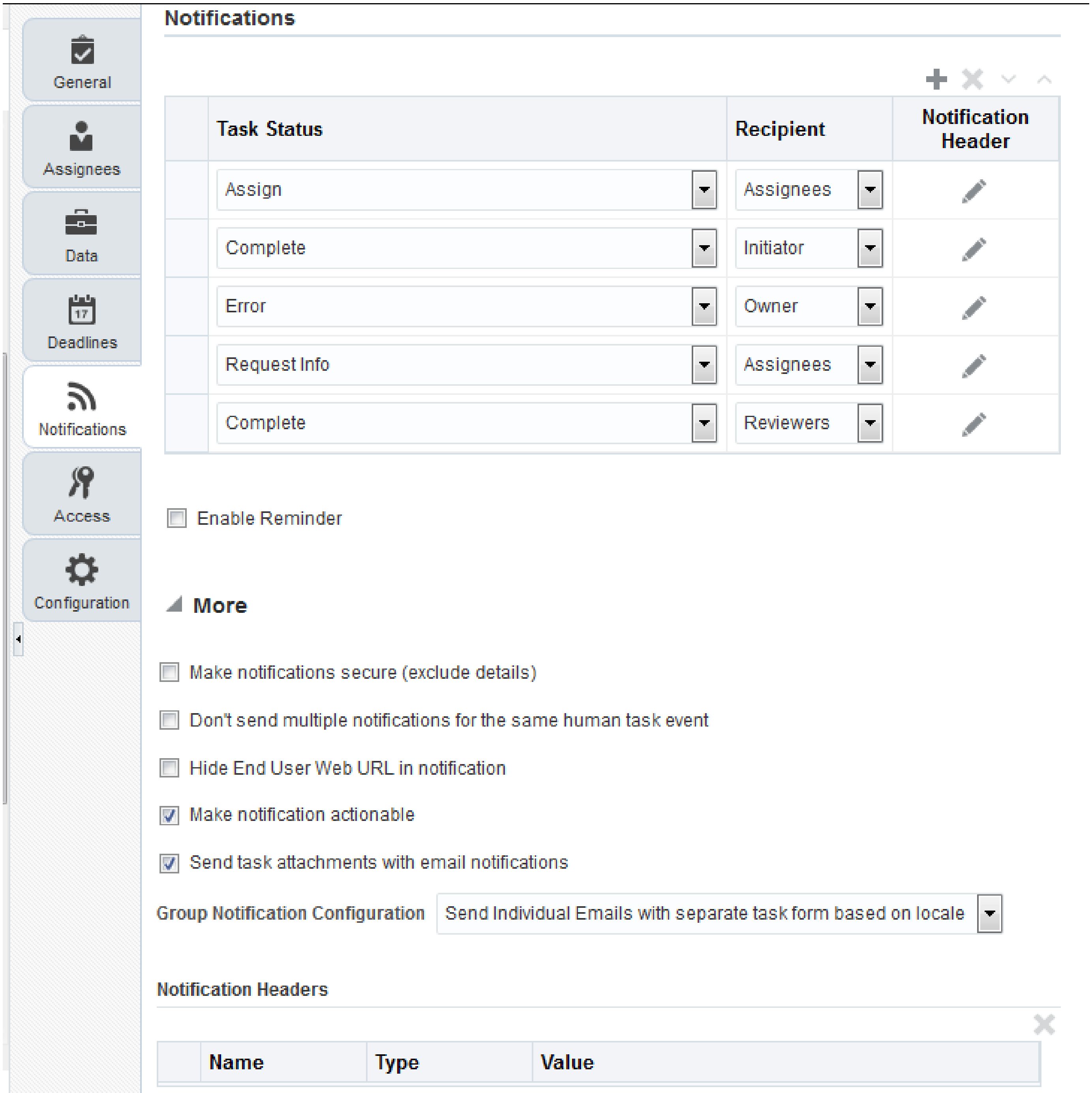

Fig 14

Notification Header

User has the limited ability to customize the notification header that appears in the header section of the email notification. For example, user can add their company logo or company name in this section.

- The default value of all the notification headers out of the box is *null.*
- Any new notification added by the customer will have the header value of "concat(string('Task '), /task:task/task:title, string(' requires your attention.'))". It is recommended to change it to *null*.
- Company Logo
    - In below string, change value of source to the URL of your company logo and give proper alternative text. Copy the modified string to "Edit Notification Message" '<img src="http://b-i.forbesimg.com/joshbersin/files/2013/07/oracle-logo3.jpg" width="230" height="69" alt="Oracle Logo">'

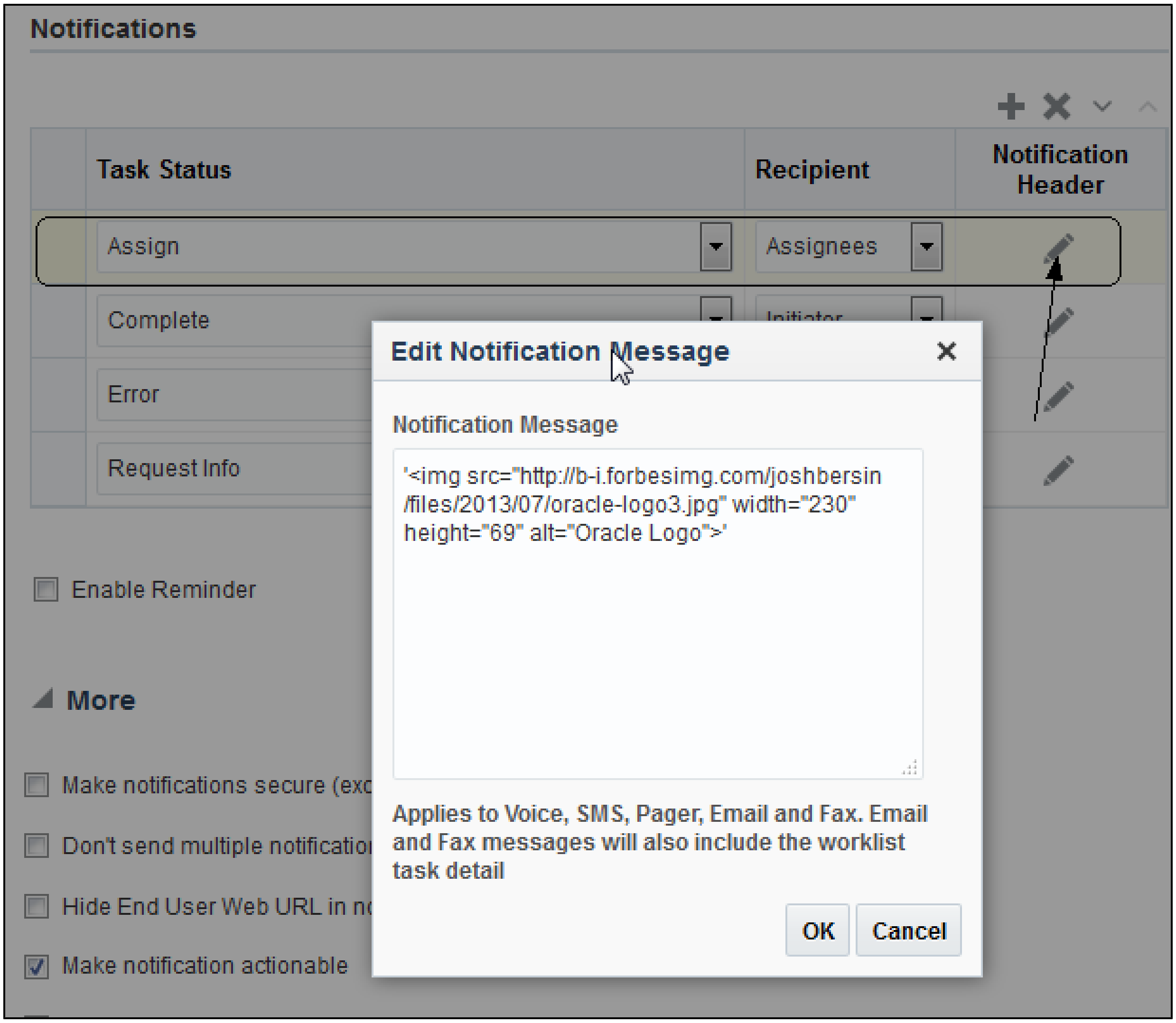

Fig 15

- Company Name
    - In the popup window, enter the company name or some other string in single quote. For example, 'Oracle'. For better formatting, you can put HTML code here, like '<h2>Oracle</h2>'

Bell Notification Sync-up

- There is ESS Job - **Synchronize Bell Notifications** that is introduced in R12.1 to sync-up the Bell Notification Popup with BPM online notifications. This means that all the completed notifications will be removed from Bell notifications.
- Privilege needed to Schedule/Execute this Job Definition is : FND_MANAGE_SCHEDULED_JOB_DEFINITION_PRIV (This privilege rolls upto all the Family Admin Job Roles).
- The customer has to Schedule the "**Synchronize Bell Notifications**" Job Definition at desired frequency to update all the tasks in the Bell Notifications to Synchronize with the BPM Worklist.

Managing Bell Notification

Enable/Disable Bell Notification also uses the same notification settings done for e-mail Notification. If the e-mail notification is disabled for specific action the user will not get any Bell and E-Mail notification.

Enable Reminder

You can send task reminders, which can be based on the time the task was assigned to a user or the expiration time of a task. The number of reminders and the interval between the reminders can also be configured.

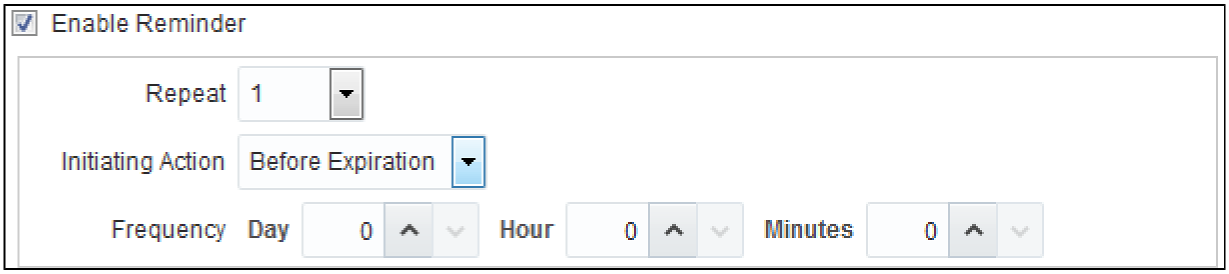

Fig 16

- From the list, select the number of reminders to send. System will only send upto 3 reminders.
- If you selected to remind the assignee one, two, or three times, select the interval between reminders, and whether to send the reminder before expiration (expiration policy need to be set under deadline section) or after the assignment.

More

There are few more selections that you can make to customize the email notification behavior, however it is advisable to leave the default settings on.

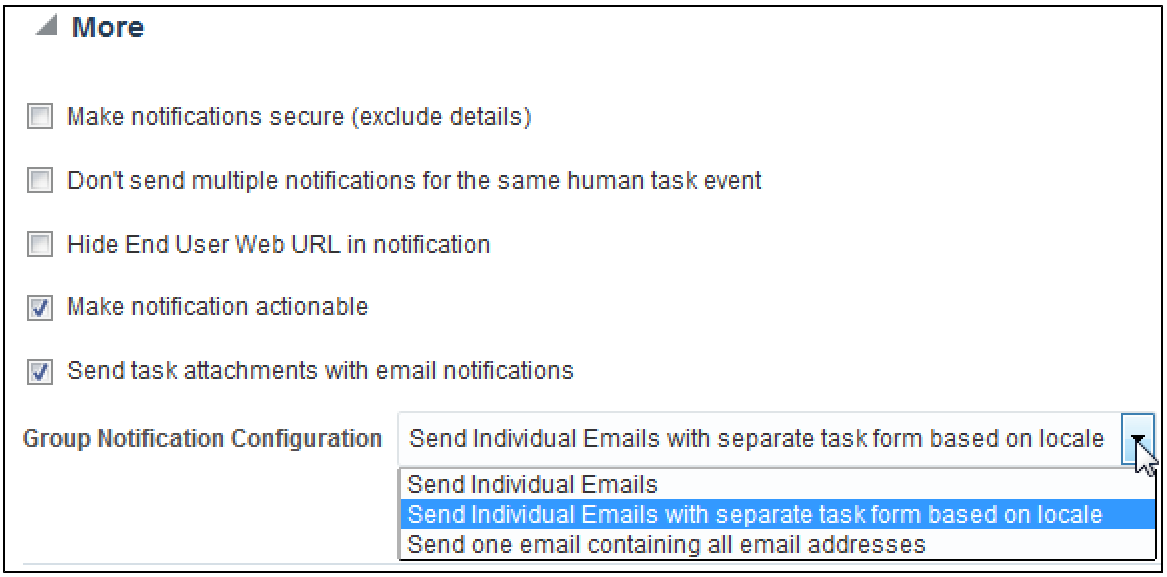

Fig 17

- Make notification secure (exclude details)
  - This will make the approval notification secure by preventing any business transaction details to be present in the email notifications
- Don't send multiple notifications for the same human task event
- Hide End User Web URL in notifications
  - This will remove the "*Access this task in the Workspace Application or take direct action using the links in this email:*" as the first line from the email body. It is highly recommended to check this checkbox to hide the workspace application link because clicking that workspace link will route the user to workspace application and not directly to the underlining transaction and this could lead to confusion with users.
- Make notification actionable
  - This controls the actionable links, *Approve* | *Reject* | *Request More Info* in the email body. This is checked by default.
- **Send task attachments with email notifications**
  - This options allows transaction any supporting documents attached to the transaction to be attached to the email notification as file attachments.

A sample email screenshot with the action links is provided below

From [redacted]
Subject **Action Required: Expense Report Approval EXM0111058958 for Mary Johnson (1,500.00 USD)**
Reply to [redacted]
To [redacted]

Access this task in the Workspace Application or take direct action using the links in this email:

Actions: Approve | Reject | Request More Info

Expense Report Approval EXM0111058958 for Mary Johnson (1,500.00 USD)

Details

From	Mary Johnson	**Report Period**	3/20/15 - 3/20/15
Assigned Date	5/29/17 7:12 AM	**Report Total**	1,500.00 USD
Task Number	200202	**Submission Date**	5/29/17
		Purpose	

Expense Items

3/20/15-**Airfare** ABBOTT, HILL, TX, United States testing	Requires Your Approval	**Project and Task** __ **Cost Center** 520	**1,500.00** USD Cash 3/20/15

Fig 18

Access

This tab controls the notification stakeholder’s access to the various different components of notification. Admin can customize these but it is not recommended.

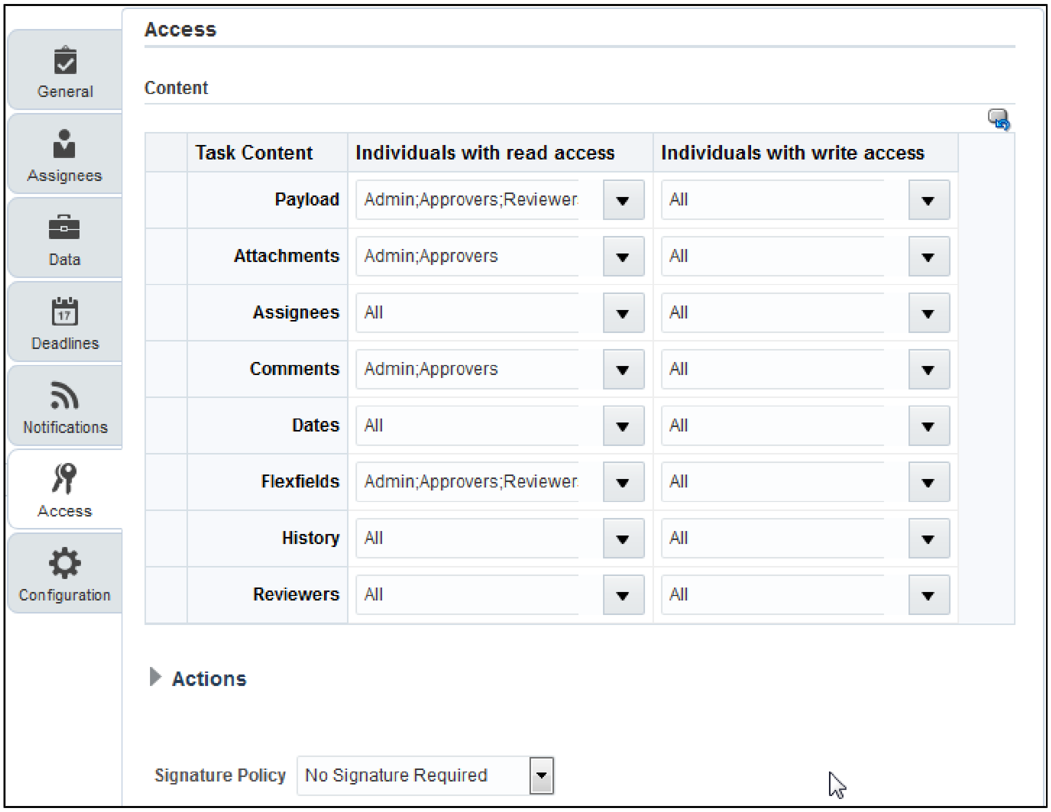

Fig 19

Expanding the action section allows to change the individuals having access to actions which can be performed on the task. For e.g., if approvers are required to approve themselves and not reassign the task to other folks in an organization, removing assignees from the reassign task action will disable the reassign functionality for approvers.

Actions

	Task Actions	Individuals with access
	Acquire	Assignees; Owner
	Adhoc Route	Assignees; Owner
	Approve	Assignees; Owner
	Reject	Assignees; Owner
	Delegate	Assignees
	Delete	Admin; Assignees; Creator; Owner
	Escalate	Admin; Owner
	Information Request	Assignees
	Override routing slip	Admin; Owner
	Purge	Admin; Creator; Owner
	Push Back	Assignees
	Reassign	Admin; Assignees; Owner
	Release	Admin; Assignees; Owner
	Renew	Assignees; Owner
	Resume	Admin; Owner
	Resume Task Scheduling	Admin; Owner
	Skip current assignment	Admin; Owner
	Suspend	Admin; Owner
	Suspend Task Scheduling	Admin; Owner
	Update	Admin; Assignees; Creator; Owne...
	View Task	Admin; Approvers; Assignees; Cre...
	Withdraw	Admin; Creator; Owner
	DECOMPOSE_TASK	Admin; Assignees; Owner

Fig 20

Configuration

Configuration tab provide various options for user to configure the working on workflow. All of the selections are self explanatory. Few of the key options are:

- Skip creator for Approval List
  - This is used to prevent self-approval. This is generally followed by checking "Assign to Creator's Manager" to route the creator task to his/her manager.
- Mandate Comments before updating these outcomes
  - This enforces task reviewer to provide comments before taking certain action. This could be very useful for auditing needs.

- Notify these participants when error occurs
    - This allows certain admin users to receive notification in case of any errors.

- Task Aggregation
    - This is one of the most important configuration options that govern how the task should be routed to different reviews in the approval cycle. Task aggregation allows notifications to be aggregated and sent only once if the same approver is assigned the task repeatedly. If the notification is already approved by the assignee, aggregation considers the approval if it is reassigned to the user and the user does not need to explicitly approve again. Following are the options for task aggregation:

        - None - Do not aggregate notifications, send one per task assignment
        - Once per task - For a given task aggregate all notifications for a user and send a single notification during the task for an assignee
        - Once per stage - For a given stage aggregate all notifications for a user and send a single notification during the stage for an assignee
        - All stages - Aggregate notifications for a user across all stages.

    There are various scenarios where a task can get aggregated and some of those will be discussed in detailed in the later part of the document.

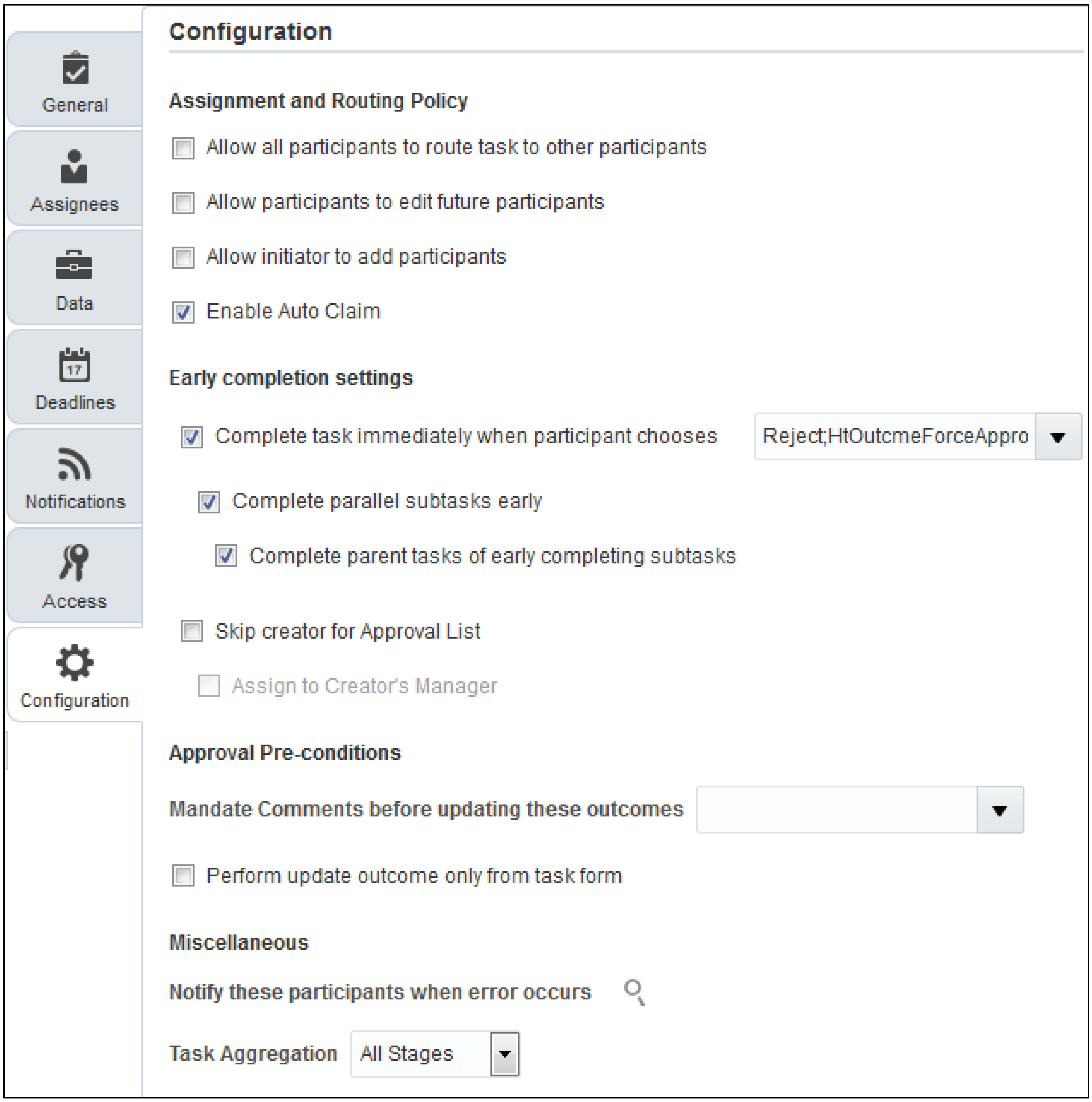

Fig 21

Task Assignment and Routing

Human workflow supports declarative assignment and routing of tasks. In the simplest case, a task is assigned to a single participant (user or group). However, there are many situations in which more detailed task assignment and routing is necessary. Human workflow provides declarative, pattern-based support for such scenarios.

Stage

A stage is a way of organizing the approval process for blocks of participant types. You can have one or more stages in sequence or in parallel. Within each stage, you can have one or more participant type blocks in sequence or in parallel.

A stage can contain multiple participants in parallel and sequential manner.

- **Parallel Mode**: When the participants are in parallel mode, the task gets assigned and notifications will go to all of the participants at once in parallel.
- **Sequential Mode**: The task gets assigned and notifications go in sequential manner, meaning one after another, to each participant in sequential mode. All have to approve sequentially to get the task approved.

Following figure shows 4 different stages for Expense approval flow.

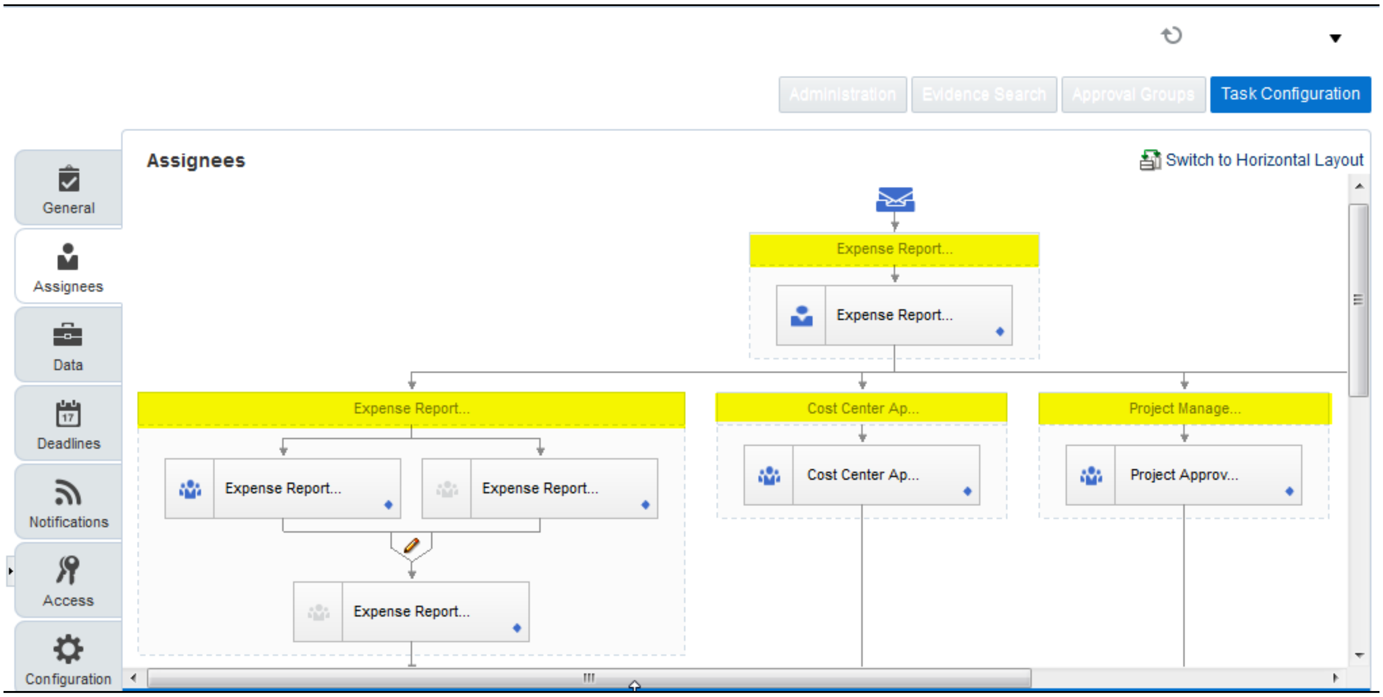

Fig 22

Stage can set to operate on a collection object in a repeated manner by selecting the checkbox shown in below figure. A collection is the grouping of transactional data. For example, a user can prepare and submit an expense report covering expense items from different cost center. There could be a requirement to get the expense item reviewed and approved by corresponding cost center manager before expense report approval. In this case the “Cost center” stage will repeat itself in parallel with each parallel branch evaluating one cost center.

Admin users are not allowed to change the collection and repetition of stage.

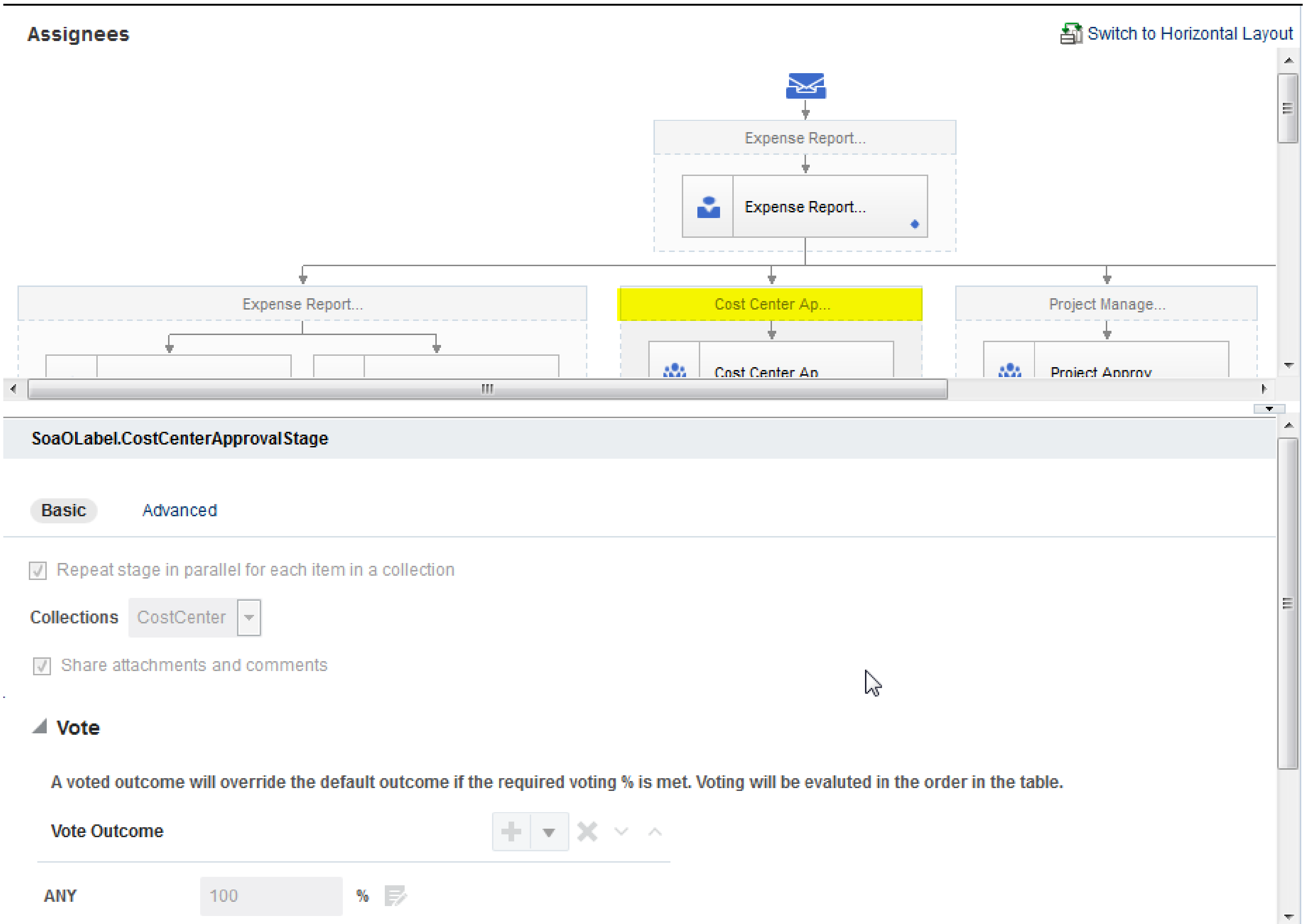

Fig 23

Based on the business requirement user can choose to disable the stage by checking the “Ignore Stage” checkbox in advanced section.

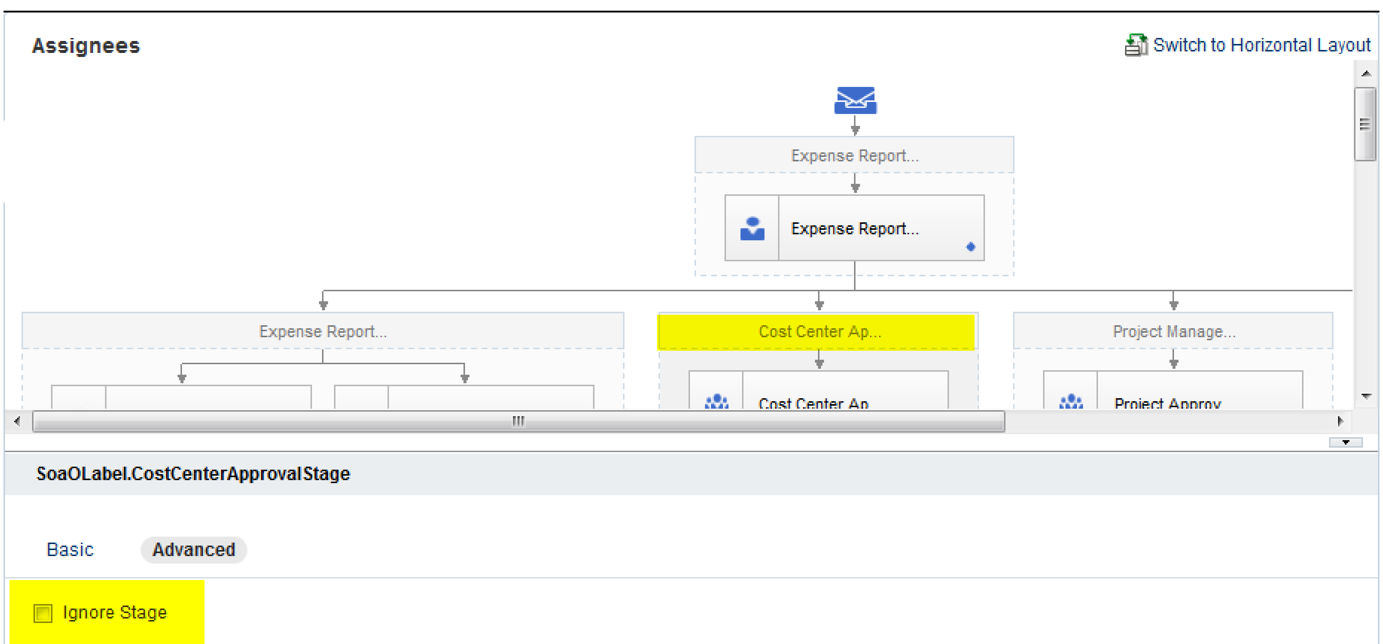

Fig 24

Participant

A participant is a user or set of users in the assignment and routing policy definition.

Participant Type

In simple cases, a participant maps to a user, enterprise group, or application role. However workflow supports declarative patterns for common routing scenarios such as management chain and group vote. The following participant types are available:

Single approver 

This is the simple case where a participant maps to a user, enterprise group, or application role.

For example, a vacation request is assigned to a manager. The manager must act on the request task three days before the vacation starts. If the manager formally approves or rejects the request, the employee is notified with the decision. If the manager does not act on the task, the request is treated as rejected. Notification actions similar to the formal rejection are taken.

Parallel 

This participant indicates that a set of people must work in parallel. This pattern is commonly used for voting.

For example, multiple users in a hiring situation must vote to hire or reject an applicant. You specify the voting percentage that is needed for the outcome to take effect, such as a majority vote or a unanimous vote.

Serial 

This participant indicates that a set of users must work in sequence. While working in sequence can be specified in the routing policy by using multiple participants in sequence, this pattern is useful when the set of people is dynamic. The most common scenario for this is management chain escalation, which is done by specifying that the list is based on a management chain within the specification of this pattern.

FYI (For Your Information) 

This participant also maps to a single user, group, or role, just as in single approver. However, this pattern indicates that the participant just receives a notification task and the business process does not wait for the participant's response. FYI participants cannot directly impact the outcome of a task, but in some cases can provide comments or add attachments.

For example, a regional sales office is notified that a candidate for employment has been approved for hire by the regional manager and their candidacy is being passed onto the state wide manager for approval or rejection. FYIs cannot directly impact the outcome of a task, but in some cases can provide comments or add attachments.

Each stage can have multiple different types of participants in sequential and/or parallel format. To skip or ignore any participant, please select the “Ignore Participant” on the advanced section.

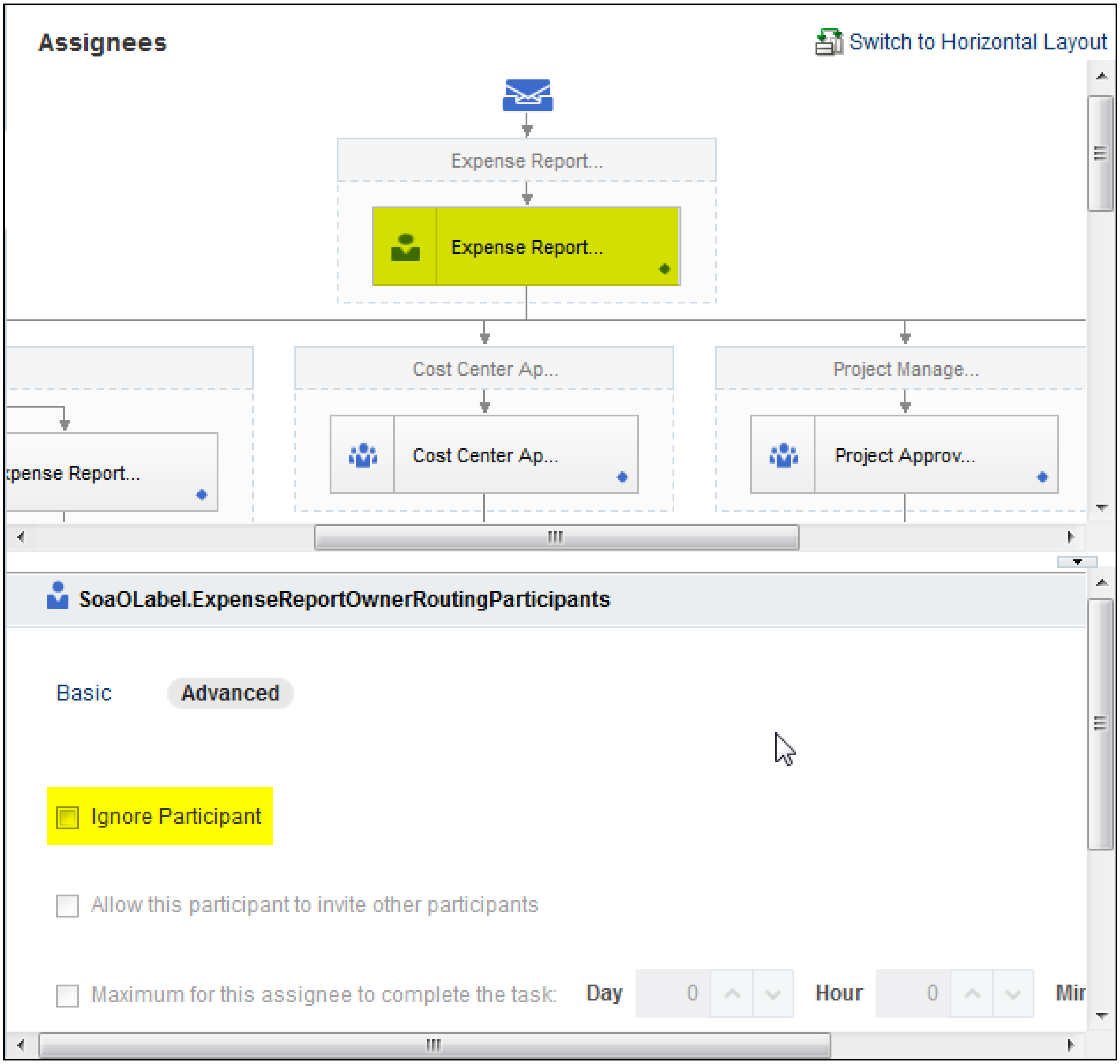

Fig 25

Ruleset

A ruleset provides a unit of execution for rules and for decision tables. In addition, rulesets provide a unit of sharing for rules; rules belong to a ruleset. Each participant is linked to a ruleset. For a ruleset to be evaluated, it must be active. Application also provides an option to set the effective date / time for execution of ruleset. It is recommended that effective date should be set to always.

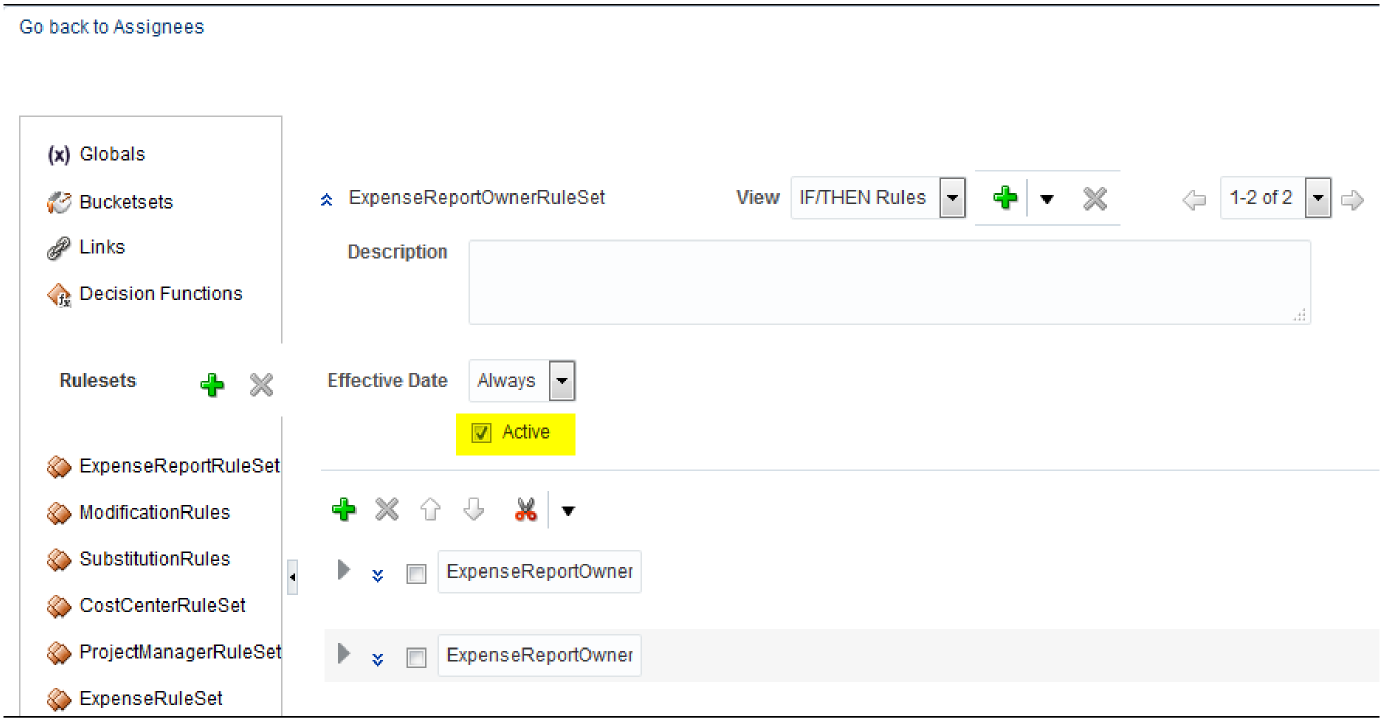

Fig 26

Rule

Business rules encapsulate the business approval requirement. Multiple rules can be created in a ruleset and all those will be evaluated at runtime in parallel. They consist of an IF and THEN component. In the IF section of the rule, you need to define the condition to determine when the rule should be applied. You can define multiple conditions if needed. In the THEN section of the rule, you need to define how approvers are to be generated if the conditions are met.

Similar to rulesets, rules also have option to select effective date and it is recommended to select it as "Always". There are 7 different options to set priority of the rule from *Highest* to *Lowest* and by default all rules are set at *Medium* priority. A rule has to be *Active* to be evaluated.

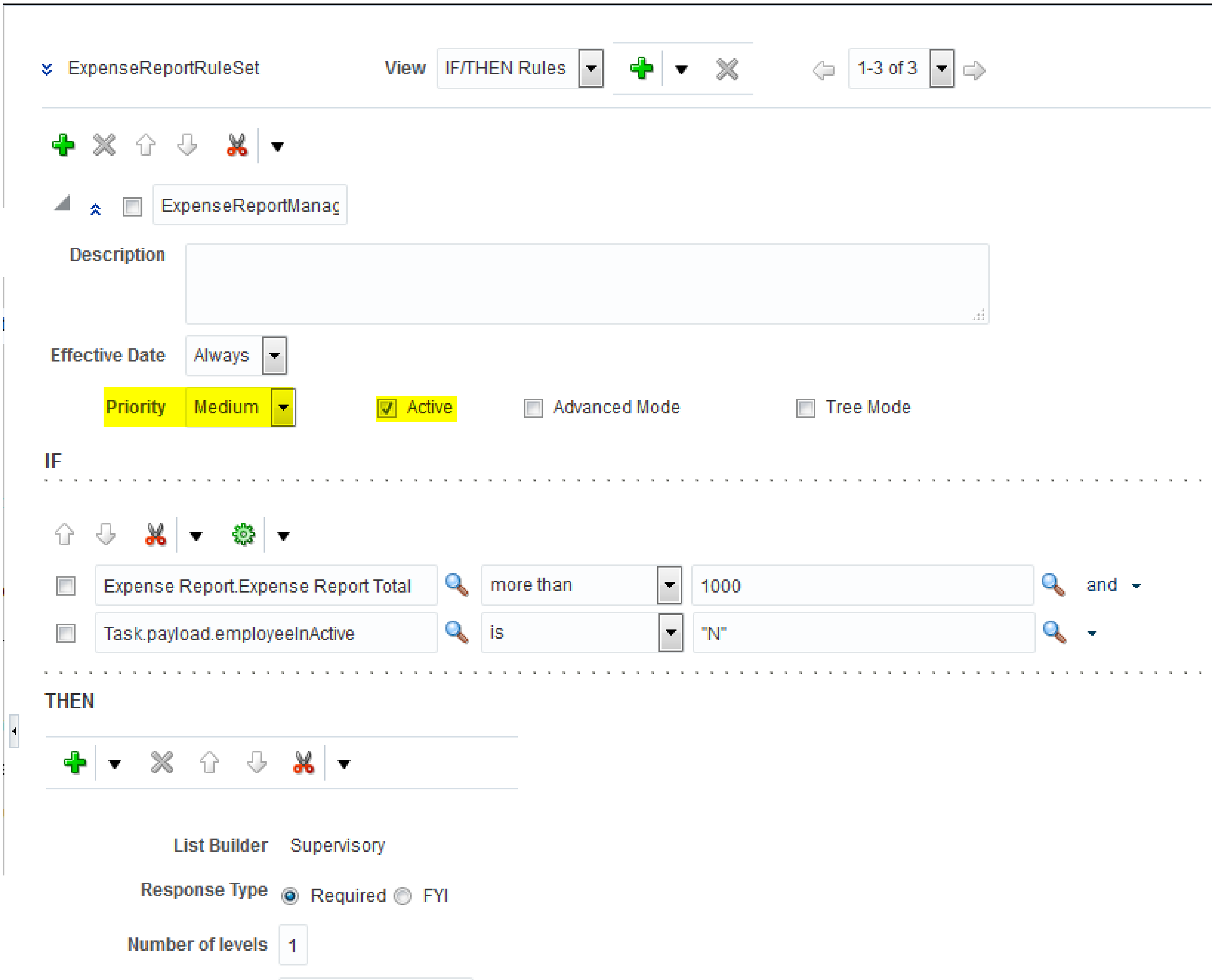

Fig 27

For using multi-valued lists in the conditions, user can use global lists. Below is an example of using a list of BUIds for a condition in a rule

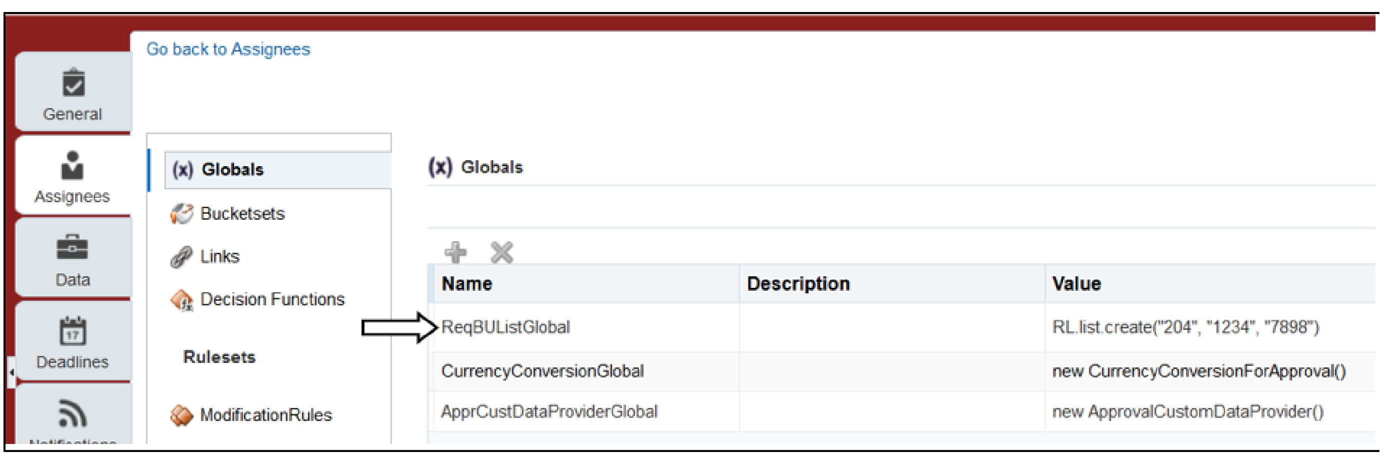

Fig 28

We can then create a rule to check if BUId matches any in our global list

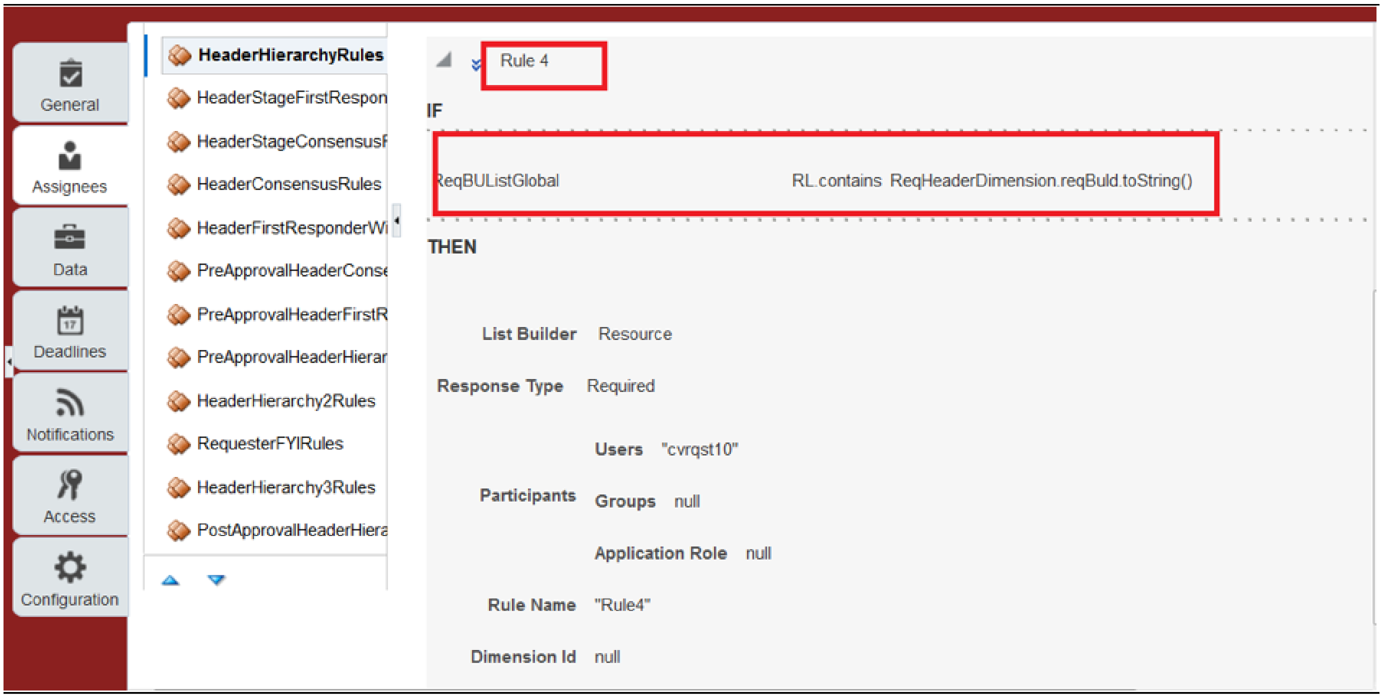

Fig 29

Advanced mode rule creation

Advanced settings for rules let user work with features that provide advanced options. These include additional pattern matching options allowing user more control over the child collection of a transaction like InvoiceLine, InvoiceDistribution, ExpenseItem, ExpenseDist, JournalLine etc.

Advanced Mode Pattern Matching Options

There are four additional pattern matching options that are only available in advance mode:

- *for each case where*
  - This is the default pattern matching option. A rule should fire each time there is a match (for all matching cases).
  - Rule will be evaluated for all the lines and notification will be sent for all the lines satisfying the condition.
- *there is a case where*
  - This option selects one firing of the rule if there is at least one match. In this feature rule will be evaluated until it finds a line satisfying the condition. As soon as it finds a line satisfying the condition notification will be sent for approval. If there is no line satisfying the condition, no notification will be send.
- *there is no case where*
  - The value specifies that the rule fires once if there are no such matches.
  - With this feature Rule will be evaluated until it finds a line satisfying the condition. Notification will be send only if there is no line satisfying the condition.
  - Should not be used with the NOT operator.
- *aggregation*
  - Aggregation functionality is used when customer need to perform operations (like Sum, Count, Maximum, Minimum, Average, Collection) on attributes of collection. When you write rule conditions that are based not only on one fact, but on many facts, you can use an aggregate. You use aggregate functions when the conditions have a view spanning multiple facts.

Steps to configure rules

Here we will see in detail on what are the constructs for defining rule in advanced mode and also in example which explains in detail on converting rule in basic mode to advanced mode.

Select "Advance Mode"

To show the advanced settings, next to the rule name click "Show Advanced Settings" icon and then select "Advance Mode" checkbox.

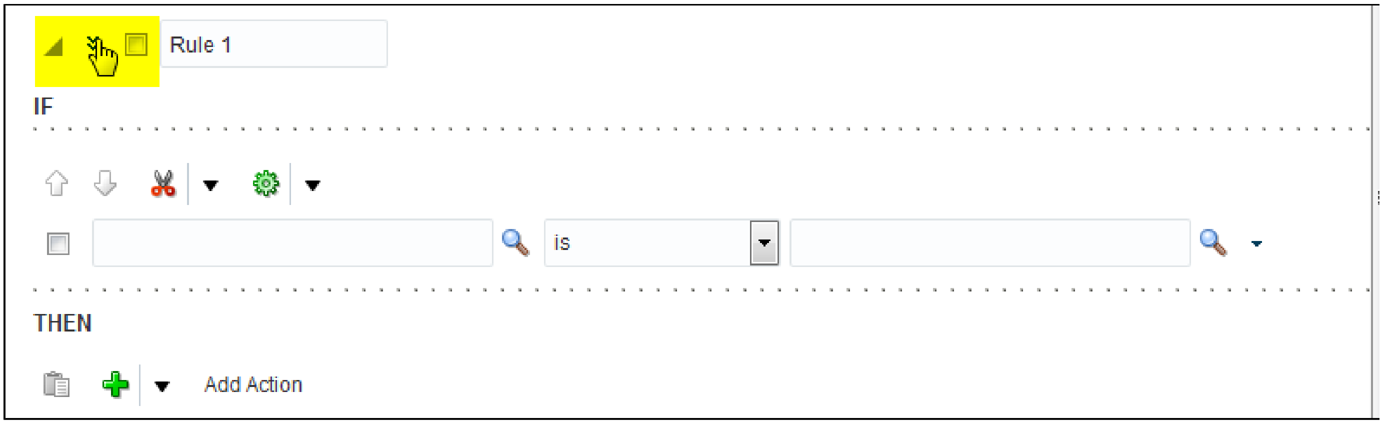

Fig 30

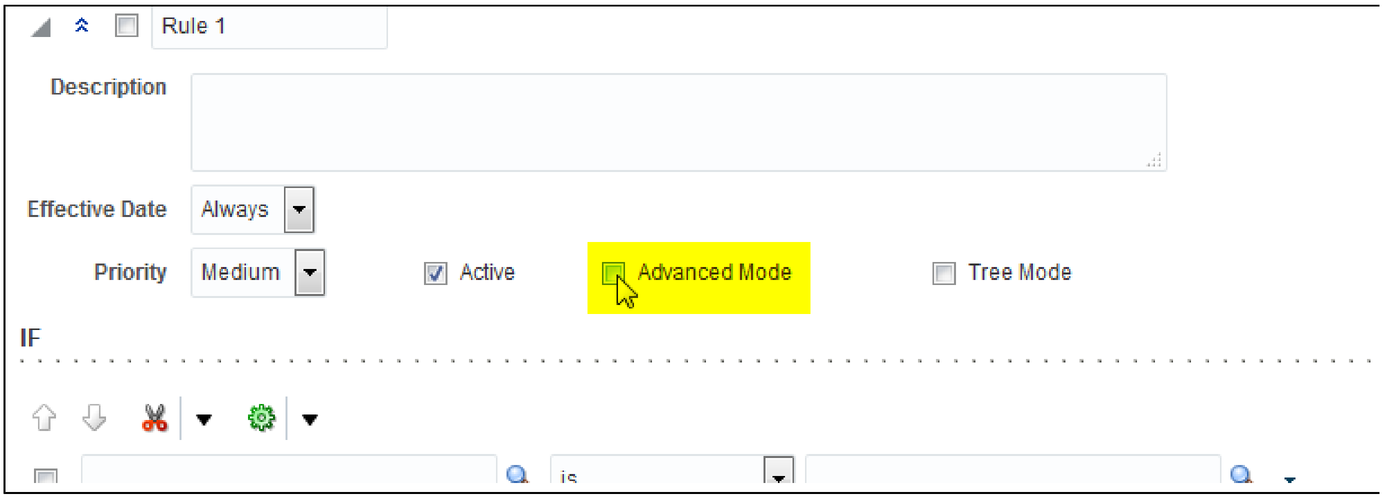

Fig 31

Add Pattern

By default, one "is a" pattern will be added to the rule as soon as "Advanced Mode" is selected. The left side of a pattern is a variable and the right side is a fact type or a fact path. By default, OBR Designer sets the variable name equal to the fact type or path. The following figure shows a variable "ER" of type "ExpenseReport".

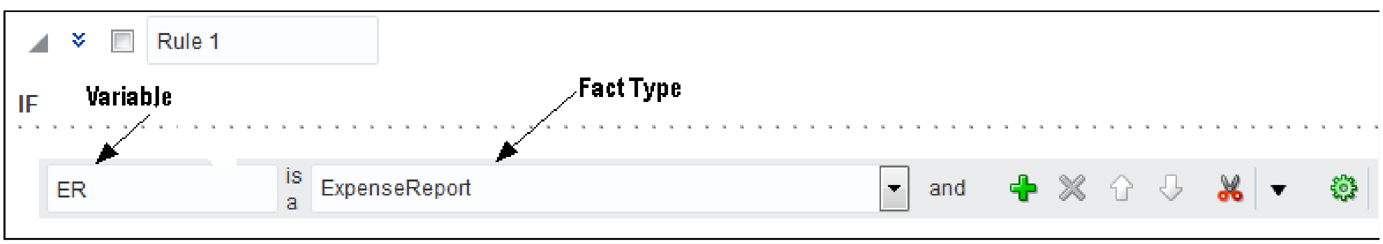

Fig 32

Define Condition

Use the variable defined in above patter to create a condition on which rules has to be based on. Following figure shows a condition where expense report total amount is greater than 1000 but less than 5000.

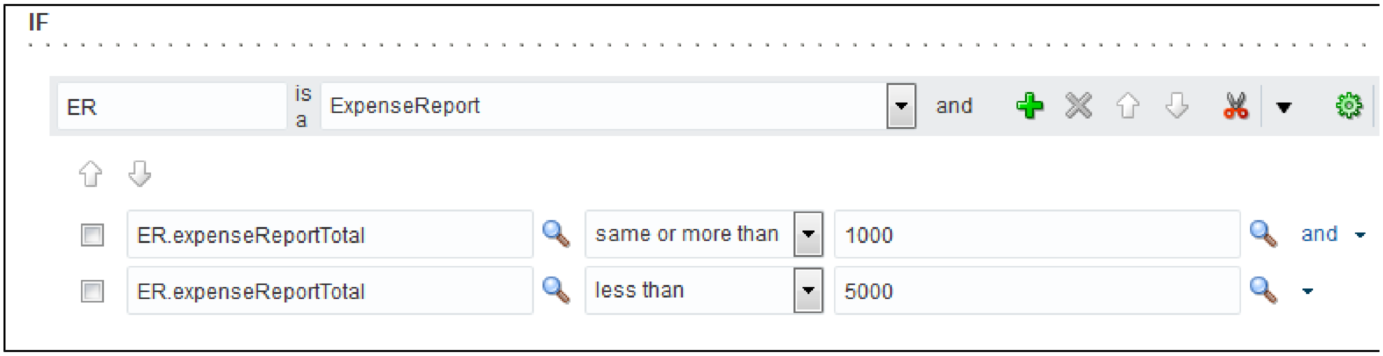

Fig 33

Adding Child collection

As Expense report could have multiple Expense items, lets see how to use various attributes of expense items in advance mode.

1. Define a collection "**Lists**". This is similar to the JAVA collection data type. This list will be populated with the matching child collection based on the condition defined. **Lists** is mandatory pattern. We need this to be defined because in THEN part we have a list builder [Supervisory, JobLevel, Resource, Approval Group, Position] which internally uses Lists collection.
    1. Now click on "Add Pattern" button to add other patterns.
    2. Add Lists pattern as below. No need define any conditions under this, as we just need this to iterate patterns defined using collections or also to prepare the list builder.
2. **Task** is used whenever user wants to define approval rules based on human task payload or any other children of Task. For Ex, Task.payload, Task.creator etc..

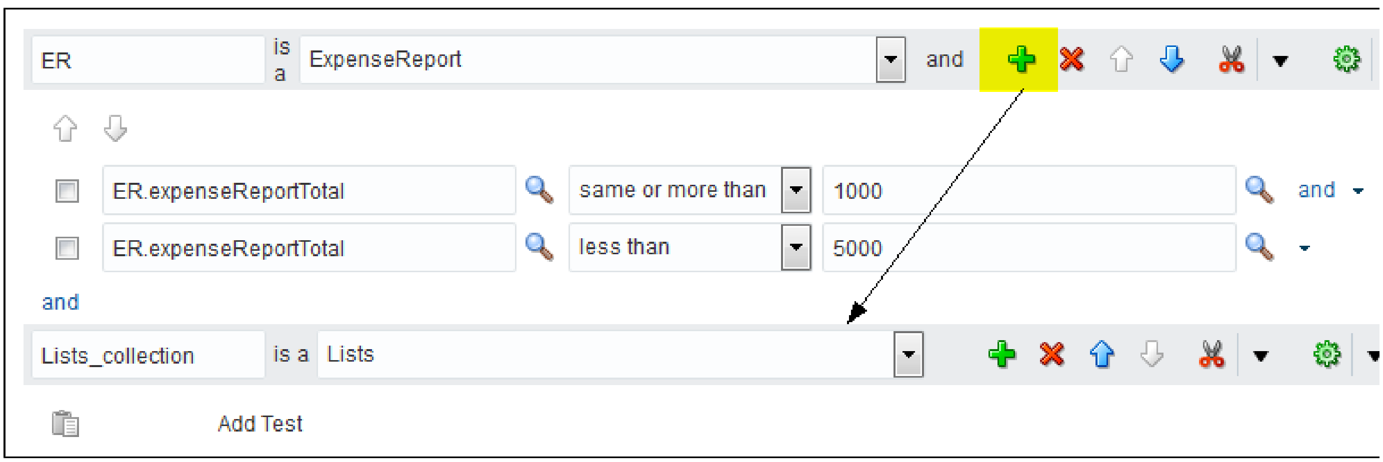

Fig 34

2. Now add pattern for ExpenseItem to add conditions based on item attributes. As rules need to be defined on ExpenseItem attributes, corresponding pattern need to be defined. The following figure shows a pattern on ExpenseItem with "item" variable of ExpenseItem type and then using ExpenseSource attribute of expense item.

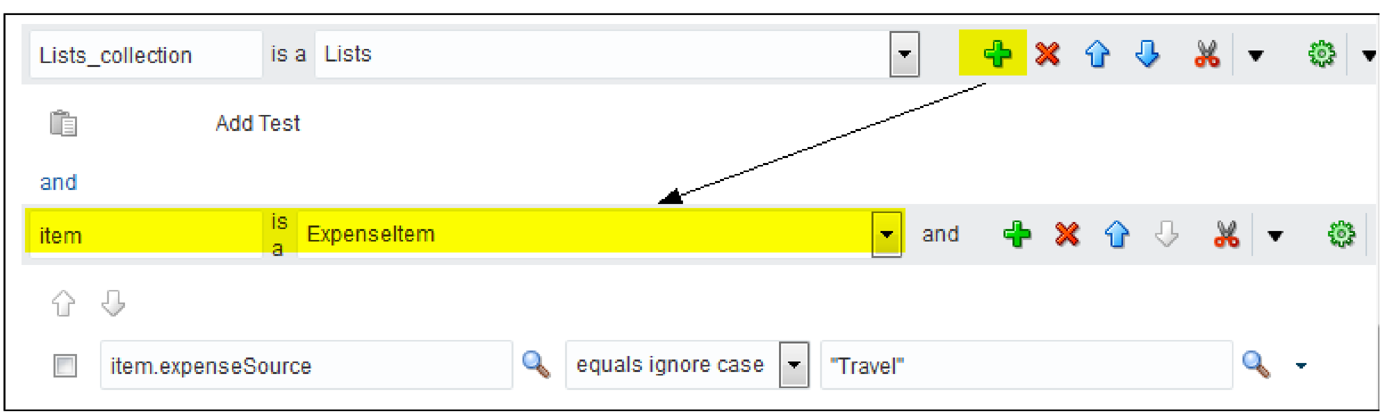

Fig 35

Using Advanced Pattern

As Expense report could contain multiple expense items, the next step is to define a pattern matching option for the expense items. To do this, surround the expense item patter with the advance pattern matching option.

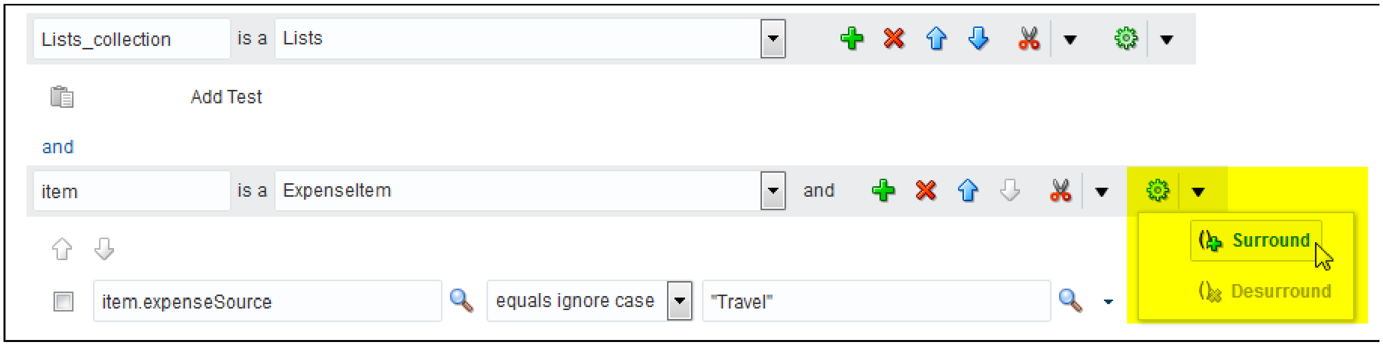

Fig 36

Based on the business requirement, different option can be selected.

- If ONLY one instance of ExpenseSource == Travel satisfy the requirement then select "there is a case where"
- If there should be NO instance of ExpenseSource == Travel then select "there is no case where"
- If ALL expense items should belong to Travel then select "for each case where"
- Select "Aggregate" to use various aggregation operations. Please refer to How to Use Advanced Mode Aggregate Conditions for more information.

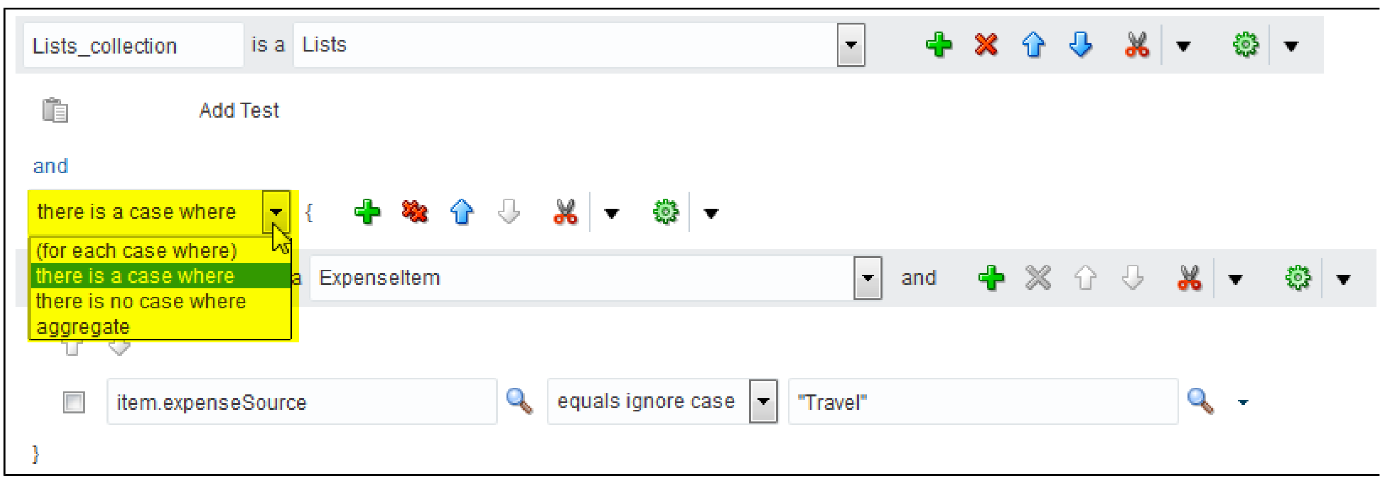

Fig 37

Validate, Save and commit

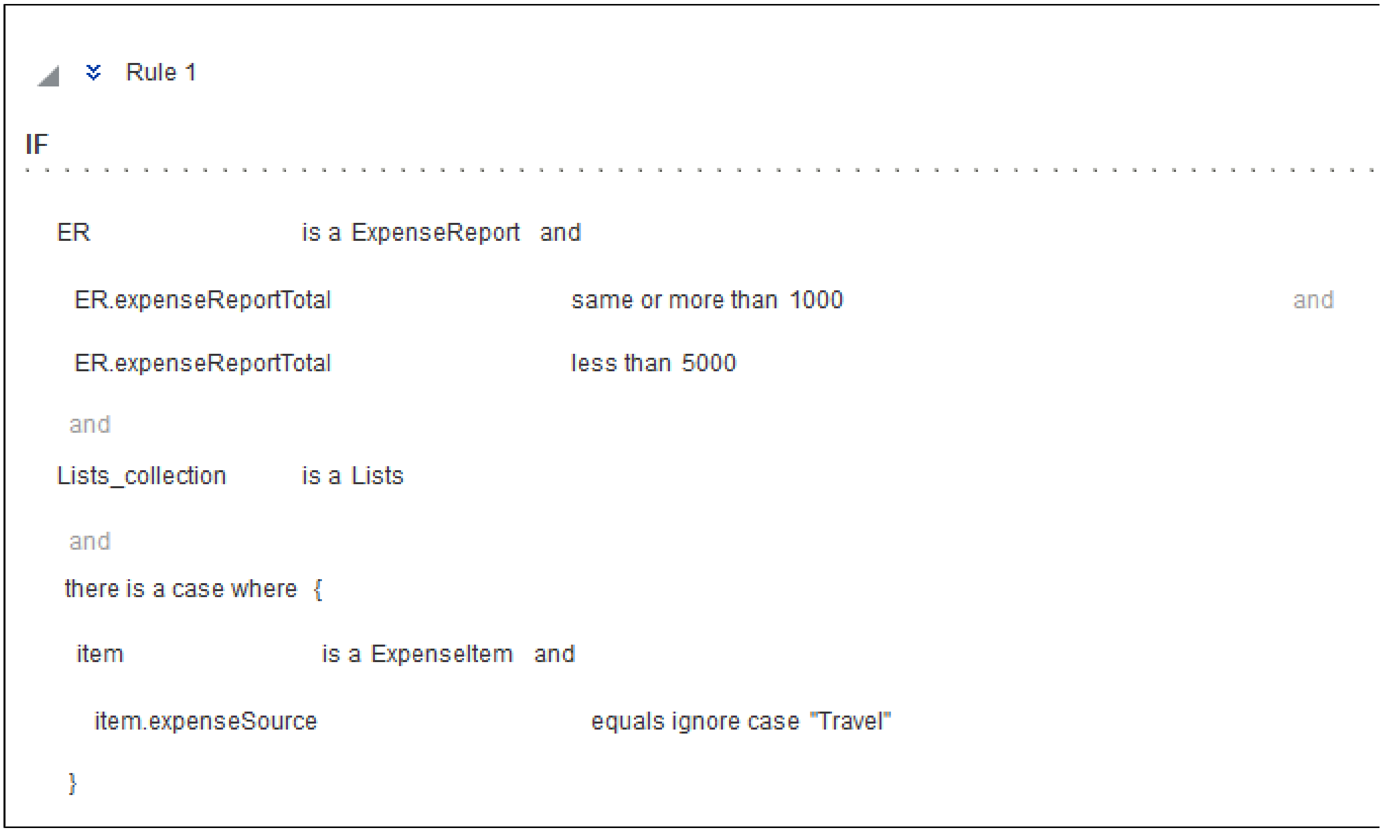

Fig 38

List Builders

List builder is the way approvals management builds the list of approvals required for a transaction based on the rule condition. Each approval rule is associated with a list builder for generating the list of approvers.

- Supervisory
- Job Level
- Position
- Approval Group
- Resource

Supervisory

Approvals can be set up based on the employee supervisory hierarchy, which is defined in HCM. Employees must be set up in HCM with appropriate jobs and supervisors. For example, the clerk reports to the manager, who reports to the director. Following figures shows how to add Supervisory list builder in THEN part of the rule.

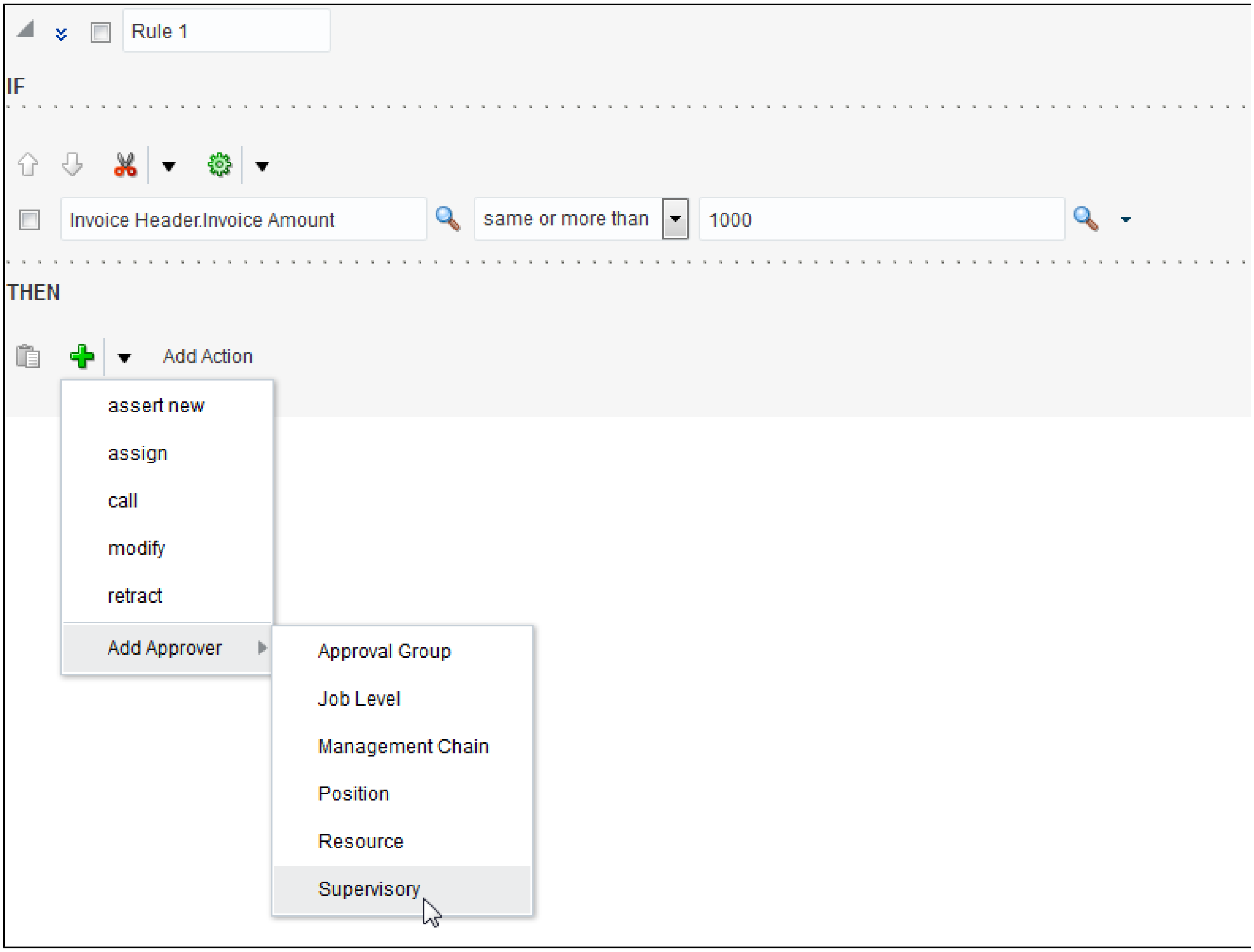
Rule 1
IF
Invoice Header.Invoice Amount
same or more than
1000
THEN
Add Action
assert new
assign
call
modify
retract
Add Approver
Approval Group
Job Level
Management Chain
Position
Resource
Supervisory

Fig 39

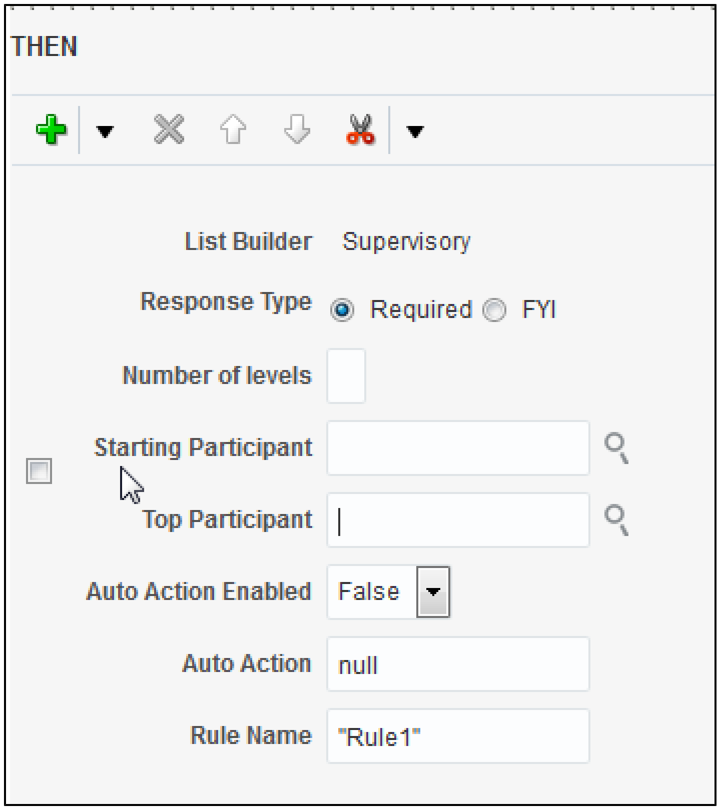

Fig 39

Number of levels represents the level of approvals required from starting participant e.g.HierarchyBuilder.getPrincipal("workflowsystem",-1,"","") . Top participant is the last approver in the approval chain. Approval does not go beyond this participant in a hierarchy. To select the starting or top participant, click on the search icon and appropriate selections.

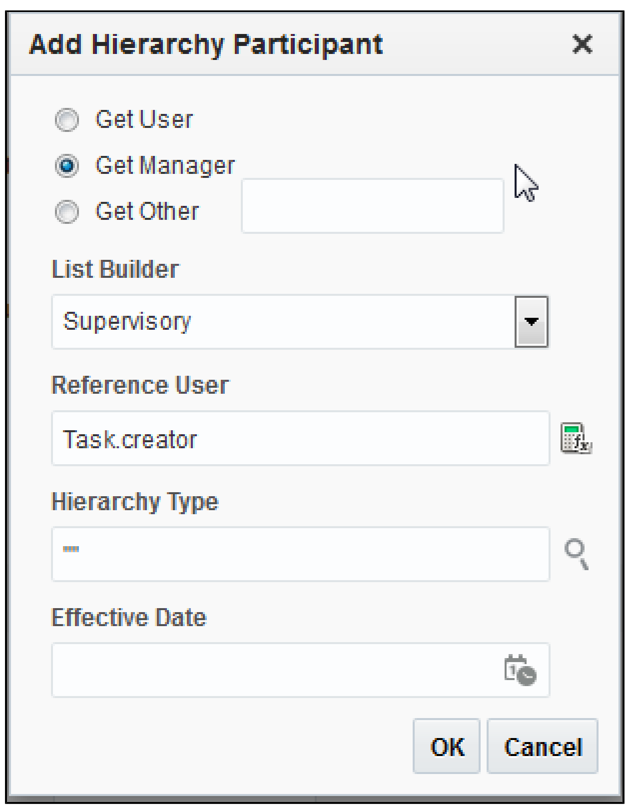

Fig 40

Hierarchy Type

There are four type of hierarchy that can be used:

- LINE_MANAGER
- PROJECT_MANAGER
- RESOURCE_MANAGER
- REGIONAL_MANAGER

Auto Approval / Rejection rule

By selecting "Auto Action Enable" to "True", the approval rule can be set to skip the human approval and set to take the action as specified in "Auto Action" field, APPROVE or REJECT. This option is available with Supervisory / Job / Position list builder but as supervisory is the light weight option, it is recommended to use supervisory list builder for setting up auto action rules. Starting participant will be evaluated all the time so it is recommended to use

- HierarchyBuilder.getPrincipal("workflowsystem",-1,"","")
  OR
- HierarchyBuilder.getPrincipal("Task.creator",-1,"","")

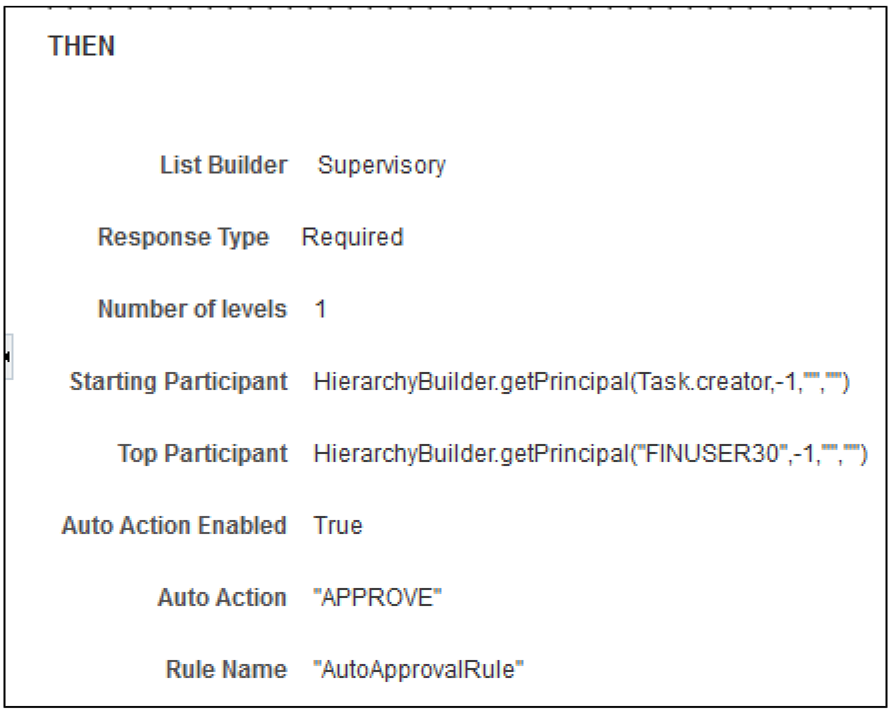

Fig 41

Job Level

Job level routings are based on the supervisory hierarchy defined in HCM. Employees must be set up in HCM with the appropriate job levels and supervisors. For example, Job Level1 employee, a clerk, reports to Job Level2 employee, a manager, who reports to Job Level4 employee, a director. The approval list is generated based on the starting position specified in the rule and continues until an approver with a sufficient job level is found. The supervisory hierarchy needs to be defined along with the corresponding job levels.

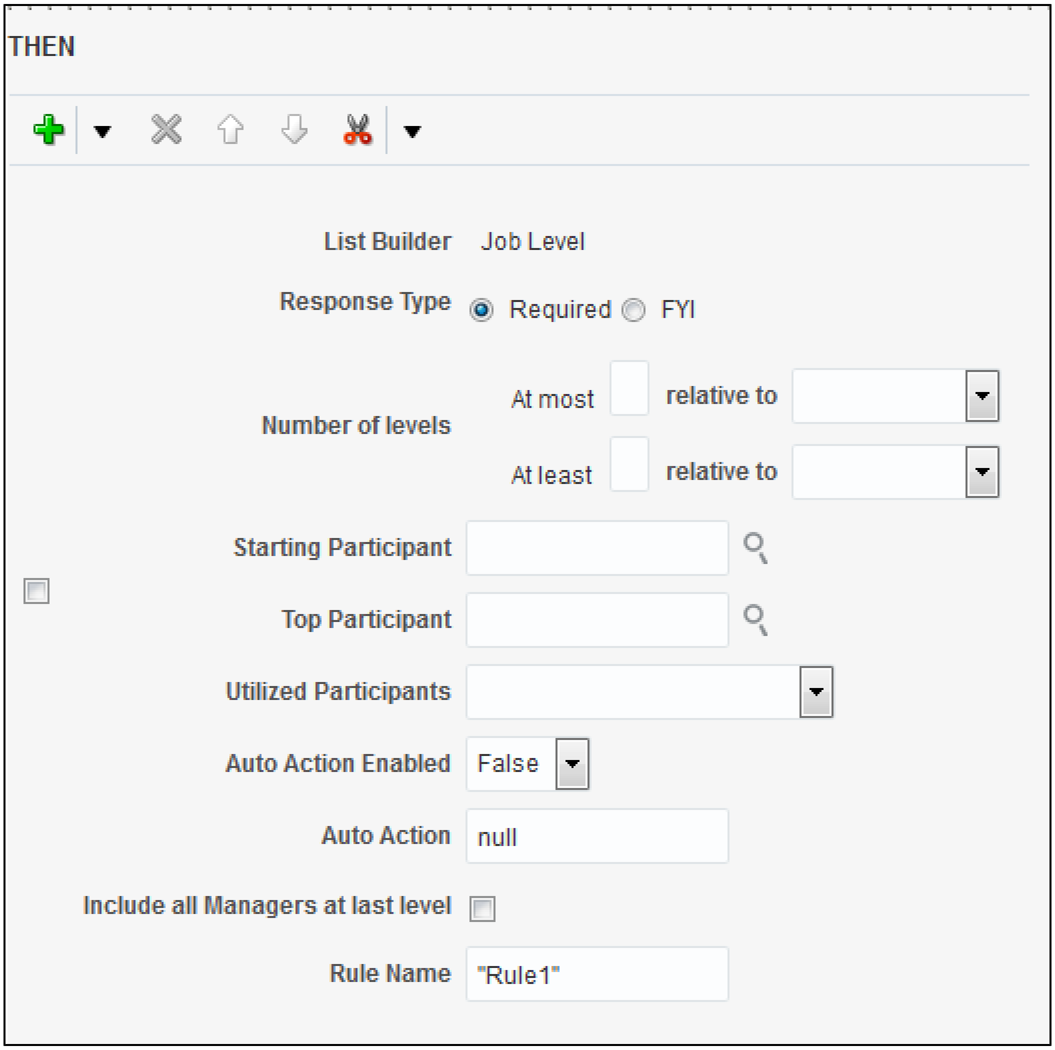

Fig 42

Position

Organizations can also choose to route invoice approvals based on the position hierarchy defined in HCM. The position hierarchy needs to be defined and employees must be assigned the corresponding positions.

Key concept in Job and Position list builder

Option Name	Description
Number of levels	This defines the number of levels the approval request should go. It has 2 values – x1(at least) and x2(at most). –relative to : values in (Starting Point, Creator, Absolute) • Absolute- Here the values x1 and x2 (at least and at most) will be absolute values starting from the 1st Job Level. • Creator– Here the values x1 and x2 will be relative to the CREATER. If creator is Job Level 3(JL3) then x1 and x2 will refer JL4 as the Job Level for x1=1 and x2=1. • Starting Point – Here the values x1 and x2 will be relative to the Starting Participant defined in the Rule. Again, x1 & x2 will be relative to Job Level of this user.
Starting Participant	The first participant in a list, usually a manager.This is not affected by any other condition like at-least, at-most, Top Participant (Except in case of Substitution. Substitution happens after all the rules are executed).
Top Participant	The last participant in the approval. Approval does not go beyond this participant in a hierarchy. If this participant comes in approval flow(and not skipped), then approval request will not go beyond this participant.

Utilized Participant	Utilizes only the participants specified in this option from the calculated list of participants. Available options are: Everyone, First and Last manager, Last manager.
Auto Action Enabled	Specifies if the list builder automatically acts on task based on the next option.
Auto Action	Specifies the outcome to be set. It can be null if auto action is not enabled.
Include all Managers at last level	If the job level equals that of the previously calculated last participant in the list then it includes the next manager in the list.

Example 1:

Lets assume user hierarchy where user with JL1 report to JL2 and so on till JL6. JL1 > JL2 > JL3 > JL4 > JL5 > JL6.

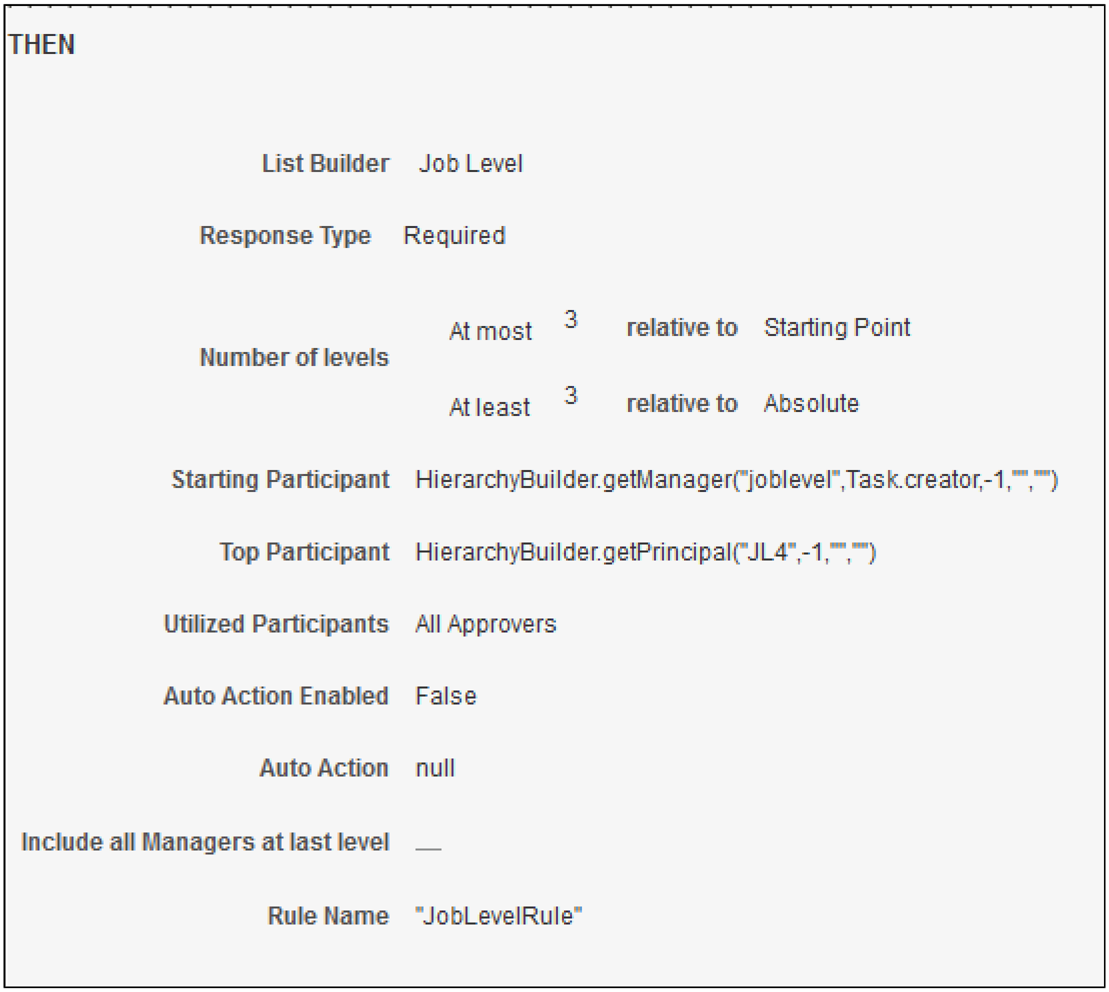

Fig 43

Task Creator = JL1

Starting participant = HierarchyBuilder.getManager("supervisory","Task.creater",-1,"",""). i.e. JL2

	At Least (relative to)	At most (relative to)	Utilize Participants	Approval Chain

1	3 (absolute)	3 (absolute)	All Approvers	• starting participant: JL2 • At least = JL3 • At most = JL3 Chain: JL2 > JL3
2	3 (absolute)	3 (Starting Participant)	All Approvers	• starting participant: JL2 • At least = JL3 • At most = JL5 Chain: JL2 > JL3 > JL4 > JL5
3	3 (absolute)	3 (Starting Participant)	Final Approver Only	• starting participant: JL2 • At least = JL3 • At most = JL5 Chain: JL2 > JL5
4	3 (absolute)	3 (Starting Participant)	Manager and Final Approver	• starting participant: JL2 • At least = JL3 • At most = JL5 Chain: JL2 > JL3 > JL5
5	1 (absolute)	1 (absolute)	All Approvers	• starting participant: JL2 • At least = no match • At most = no match Chain: JL2

Approval Groups

An approval group consists of a static predefined set of users configured to act on a task. Depending on the participant type defined, approval tasks are routed to an approval group in serial or parallel mode. For example, you can create an approval group called Finance Group comprised of users from the finance department who need to participate in the approving of a task. New approval groups can be created, or existing approval groups can be edited from the Approvals Groups tab on the BPM Worklist application Administration page.

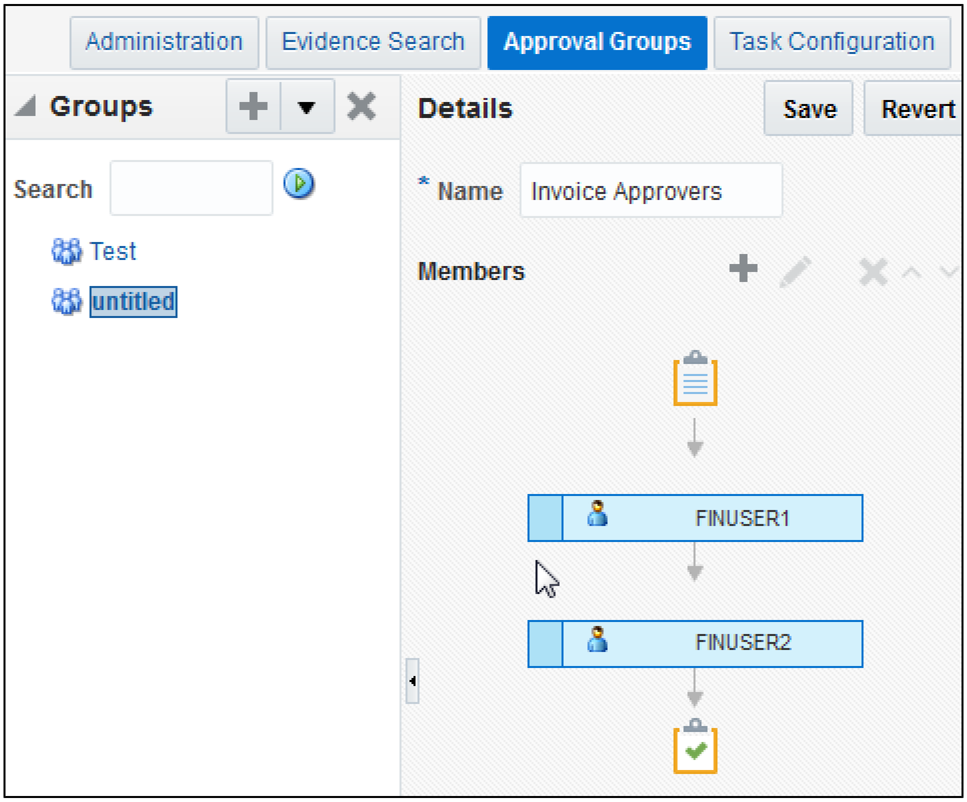

Fig 44

The approval group can then be selected in the list builder

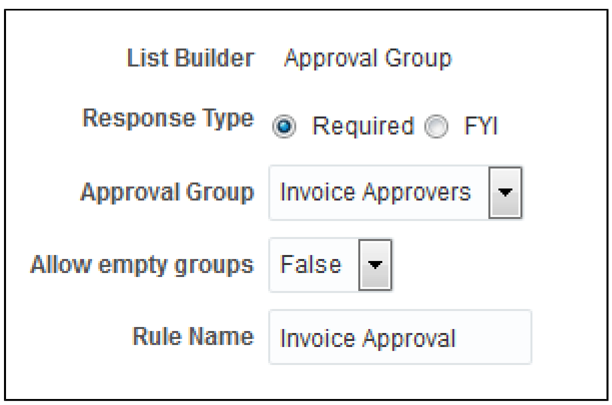

Fig 45

If empty approval groups are allowed and the selected approval group doesn't contain any member then the rule evaluation will proceed further without throwing any error. It is upto the BPEL process to handle the scenario where rules evaluation does not result in valid assignment. In most cases the transaction is either rejected or move back to original status.

Resource

Using the Resource list builder, you can build the approvers list by using a specific user, enterprise group or application role. Resource list builder is one of the easiest ways to build the approvers list.

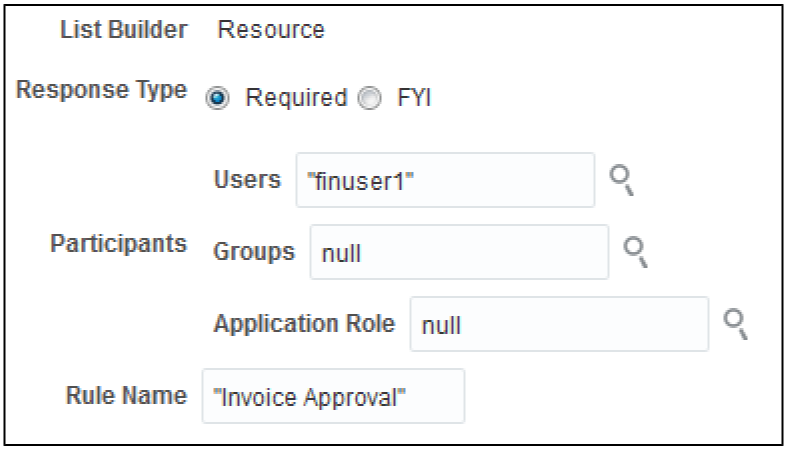

Fig 46

Approval management

This is the management of the transactions under approval process. This covers different actions available to the transaction reviewers to help make decision on the transaction under review.

Approval Channel

Following are the various avenues available to user to access the approval notifications.

Federated Worklist

We also have a Federated worklist or Integrated worklist. This worklist can be accessed from Navigator > Tools > Worklist or click the Tools icon and then click Worklist icon.

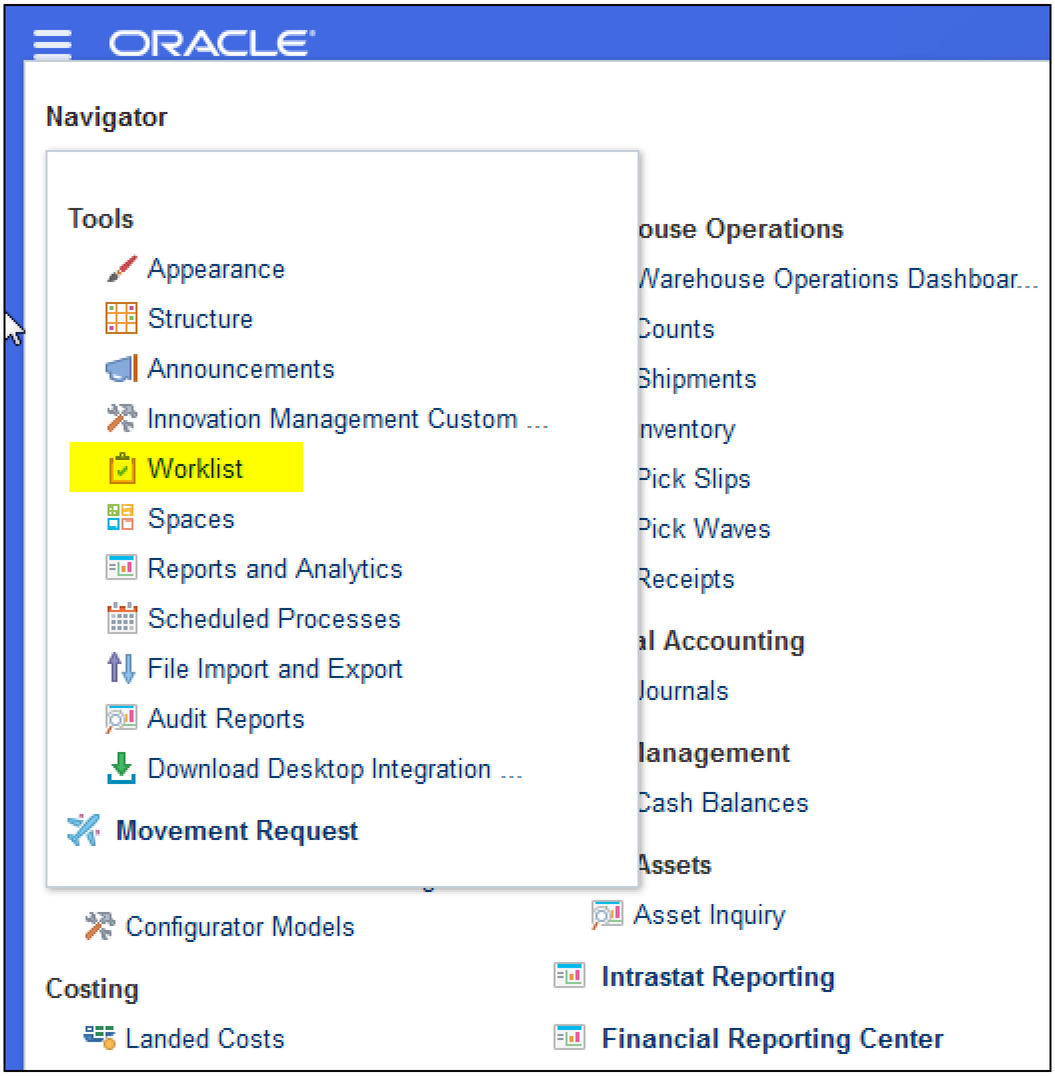
ORACLE
Navigator
Tools
Appearance
Structure
Announcements
Innovation Management Custom ...
Worklist
Spaces
Reports and Analytics
Scheduled Processes
File Import and Export
Audit Reports
Download Desktop Integration ...
Movement Request
ouse Operations
Varehouse Operations Dashboar...
Counts
Shipments
nventory
Pick Slips
Pick Waves
Receipts
al Accounting
Journals
lanagement
Cash Balances
Assets
Asset Inquiry
Configurator Models
Costing
Landed Costs
Intrastat Reporting
Financial Reporting Center

Fig 47

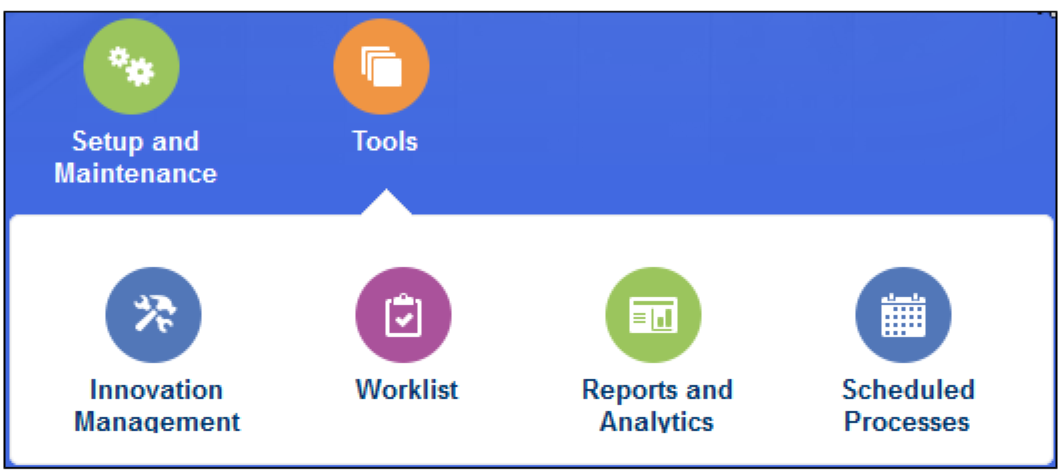
Setup and Maintenance
Tools
Innovation Management
Worklist
Reports and Analytics
Scheduled Processes

Fig 48

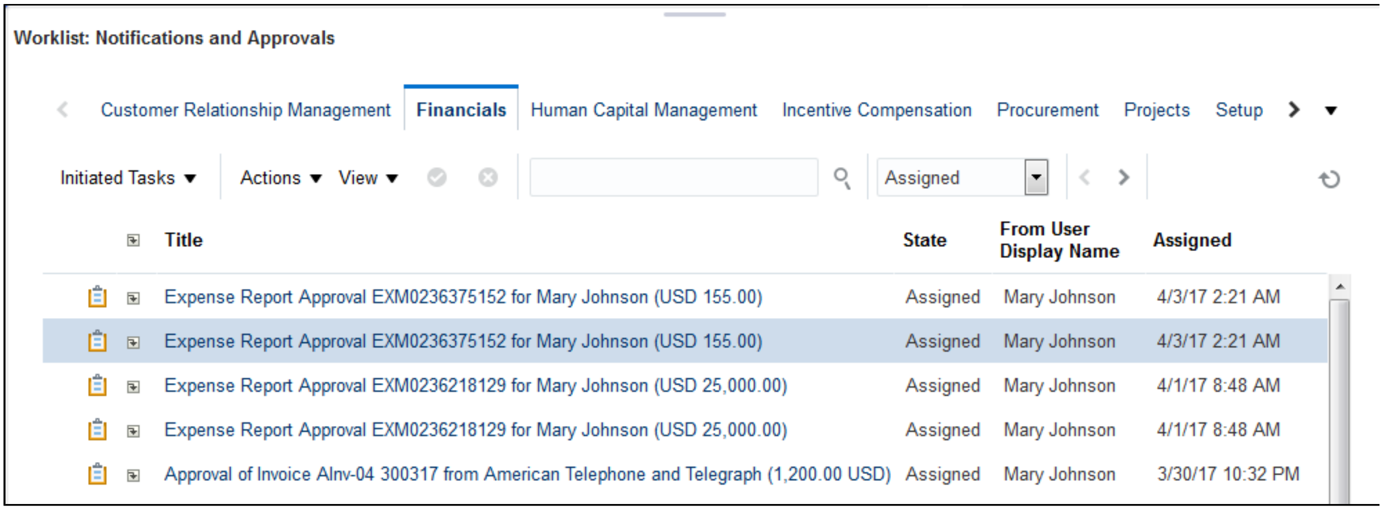

Fig 49

The data on the worklist application is arranged in multiple tabs where each tab showing the notifications for that functional area. Based on the user privilege they can view different notifications. All the tabs will be shown irrespective of user privilege.

On this page, user can view approval notification in following states, by default showing "Assigned" tasks:

- Assigned
- Completed
- Suspended
- Withdrawn
- Expired
- Errored
- Alerted
- Information Requested

Based on the selected row, various actions like Approve / Reject / Withdraw / Reassign / etc options are available in the *Actions* drowdown. Customer can also control the number of tasks they want to view on a single page, by selecting *Servers* under the *View* dropdown. Notification title is a hyperlink, opening up the detailed notification.

Bell Icon

Bell shaped icon is available in the global navigation displaying recent notifications from all functional area. It gives the count of the pending notifications and visually differentiate between new, unread and read notification. Once the notification is acted upon then it will disappear from the user's list.

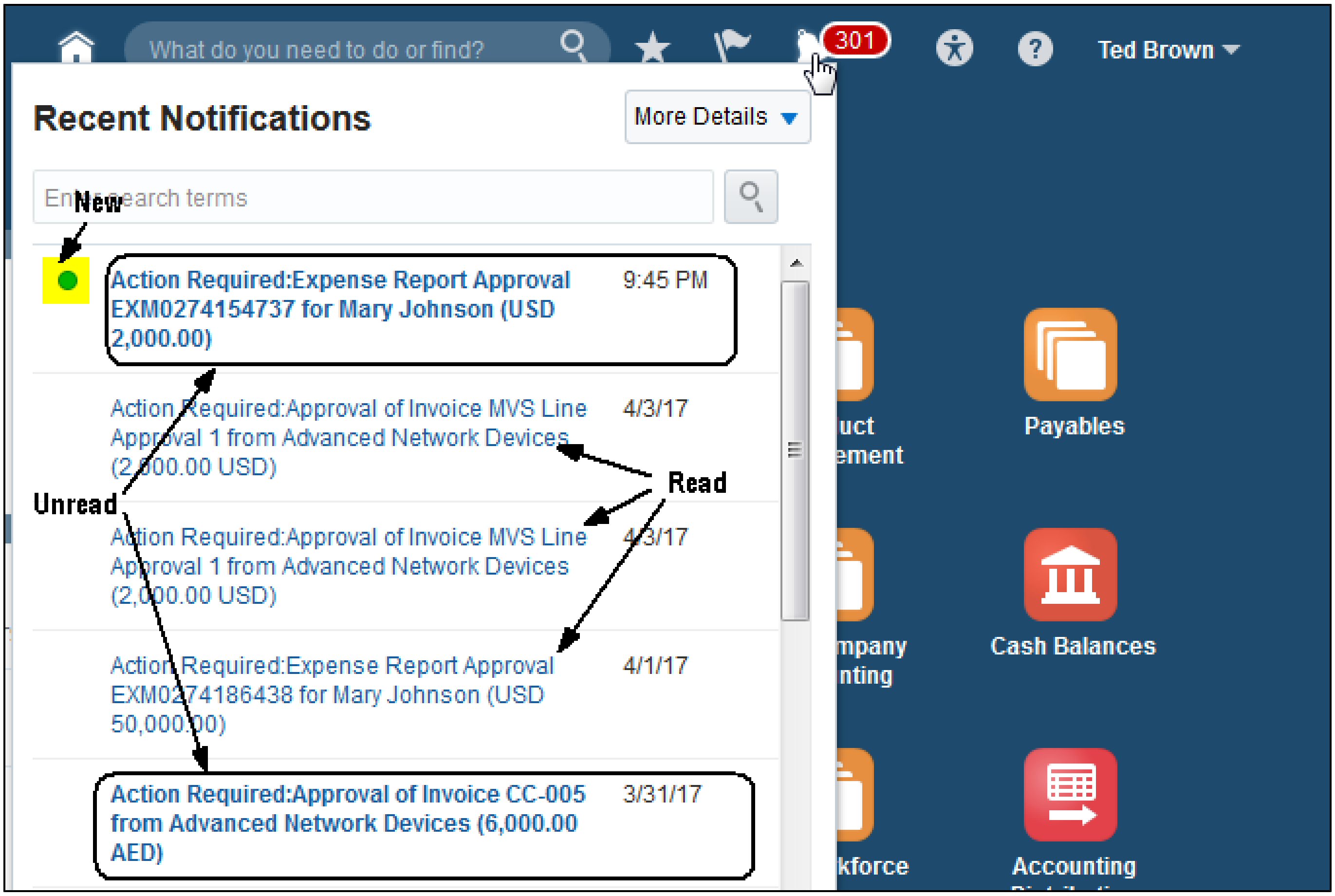

Fig 50

- Blue dot: New notifications
- Bold font: Unread
- Normal font: Read

BPM Worklist

BPM worklist refers to the domain specific worklist. The navigation is covered in the earlier sections of this document.

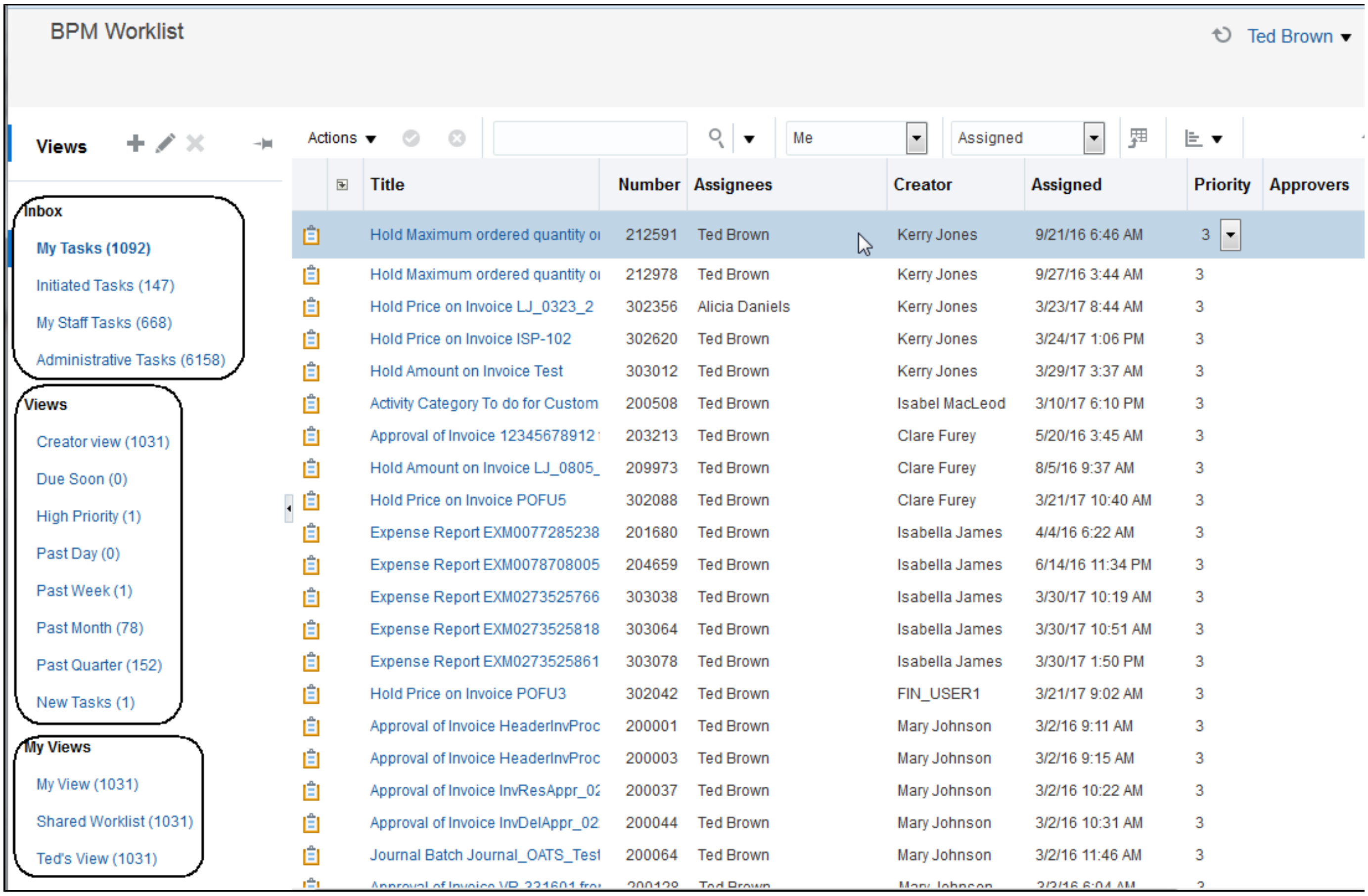

Fig 51

BPM Worklist lists all the notifications and present multiple views to filter them.

1. Inbox
    1. My Tasks - List of actionable notifications upon which logged in user need to act
    2. Initiated Tasks - List of approval tasks that user has initiated
    3. My Staff Tasks - ??
    4. Administrative Tasks - If the user has the admin privilege then this will list all the notification tasks to which admin user have access.
2. Views.
   These filter the tasks based on when the task was created, due date etc.
3. My Views
   User can create custom views based on their requirement. Those custom views are listed in this section.

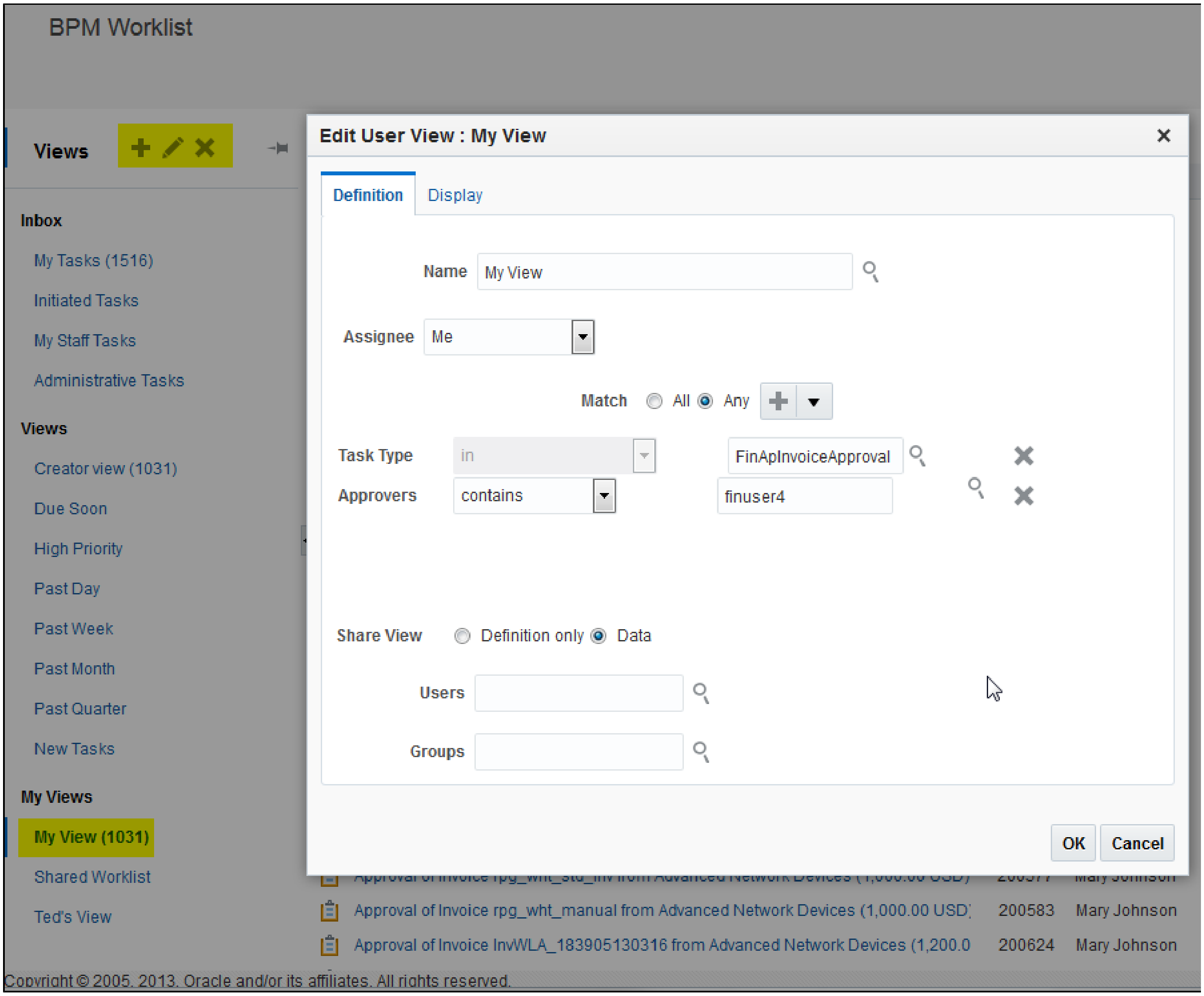

Fig 52

Email

Once the email is configured for a user, the user receives email notifications too.The user can approve, reject or request more info by replying to the email using the link in actions. While constructing the reply, make sure not to change the text of the email, except the text for comments.

From [redacted]
Subject **Action Required: Expense Report Approval EXM0111058958 for Mary Johnson (1,500.00 USD)**
Reply to [redacted]
To [redacted]

Access this task in the Workspace Application or take direct action using the links in this email:

Actions: Approve | Reject | Request More Info

Expense Report Approval EXM0111058958 for Mary Johnson (1,500.00 USD)

Details

From	Mary Johnson	**Report Period**	3/20/15 - 3/20/15
Assigned Date	5/29/17 7:12 AM	**Report Total**	1,500.00 USD
Task Number	200202	**Submission Date**	5/29/17
		Purpose	

Expense Items

3/20/15-**Airfare** ABBOTT, HILL, TX, United States testing	Requires Your Approval	**Project and Task** __ **Cost Center** 520	**1,500.00** USD Cash 3/20/15

Comments

No data to display

URL Attachment

No data to display.

Fig 53

Product UI

Every product UI has an action for approval within the UI. For each product, you can access the invoice/expense report/journal from the corresponding manage UI. In the manage page, the actions from the table menu have a approval option

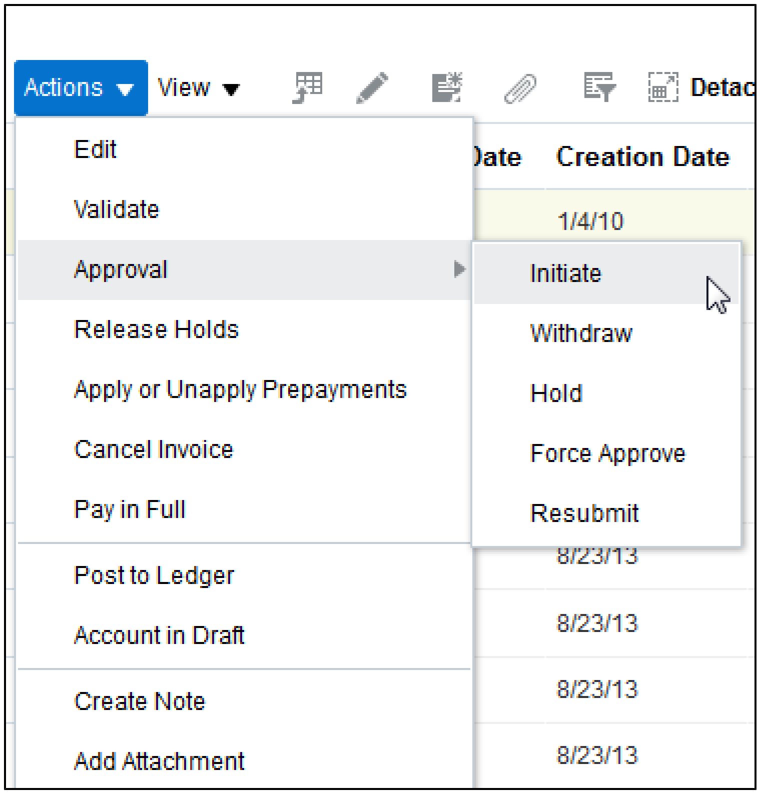

Fig 54

The create and edit UIs also have an option to initiate approval.

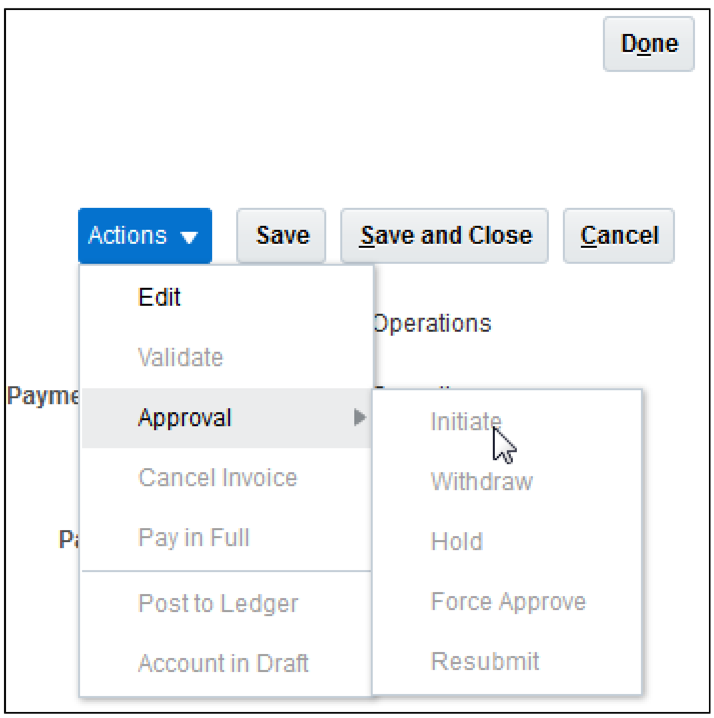

Fig 55

Approval Actions

Approve

Once a task is raised for approval, the approver can use the approve action to continue the task. The task completes if the user is the final approver, if there are more approvers in the chain, the task is forwarded to the next approver.

Reject

Similar to approval, a task assignee can reject the task to prevent further approval.

Request more information

If the task assignee wants to have more clarification from the requestor he can request more information and send the task back to the requestor. The task is not rejected in such cases and once the required information is provided, the approval flow resumes.

Reassign

The task assignee can use the reassign option to send the task to another user for approval. Once reassigned, the task now uses the reassigned user's hierarchy for approval. For e.g, if a user receives and approval requests and finds to be relevant to another department, he can reassign the task to the user from another department.

Delegate

If a user wants another user to act on his/her behalf for a task, the user can delegate the task. After the delegate approval, the orignal user's hierarchy is used for approval. The delegated use can still act on the task post task expiry. For e.g. A user can delegate another employee to approve a current task on his behalf.

Adhoc routing

A user can use the adhoc routing to insert new approvers in the approval queue outside the user hierarchy. The users can be inserted dynamically during approval. For e.g. if approval is required from a user and his manager and the user feels that it should it be reviewed by another employee before a manager approval, the user can insert the employee as an approver.

Withdraw

The task initiator can withdraw the task after the approval has been initiated

Escalate

A user can escalate the task from the current assignee to his/her supervisor.

Best Practices

Using direct attribute reference

There are two way to access the business transactional attributes in the approval rule.

1. Direct attribute reference
2. Accessing the attribute from *Task.Payload* object

By directly accessing the attribute reference, system will optimize the back end service call and only fetch the distinct attributes used in approval configuration but if the same attribute is accessed through Task.Payload object, then system will build the FULL sized object which could put unnecessary load on the system resulting in slower processing and response time. Thus it is strongly recommended to only access the direct attribute reference.

Example:

1. Direct attribute reference >> Invoice Header.Invoice Amount
2. Accessing the attribute from *Task.Payload* object >> Task.payload.getInvoiceHeader1Response.result.Invoice Amount

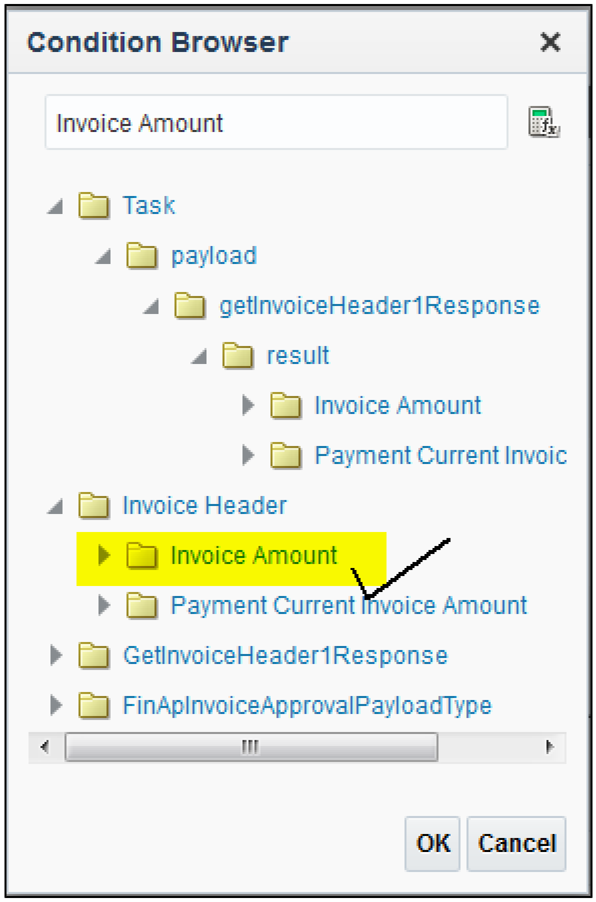

Fig 56

Advanced approval rules

In a typical financial transactional model, there is a header object with one or more line objects and each line then could have one of more distributions.

- Header
  - Lines[n]
    - Distributions[n]

Often to achieve business requirement, different attributes from various level, header, line or distribution, are required. If any non-header level, line or distribution attribute is used in the approval rules then by default approval rules engine will evaluate each line / distribution object. This works find if the business requirement is to review each and every line / distribution but if the business requirement to to check at least one occurrence then this configuration will put unnecessary load on the SOA servers. Thus it is highly recommended to configure approval in **Advanced mode** where ever any non-header attribute usage is required. As mentioned in earlier section, advance mode give users more options to best match their business requirement:

- *for each case where*
  - This is the default pattern matching option. A rule should fire each time there is a match (for all matching cases).
  - Rule will be evaluated for all the lines and notification will be sent for all the lines satisfying the condition.
- *there is a case where*
  - This option selects one firing of the rule if there is at least one match. In this feature rule will be evaluated until it finds a line satisfying the condition. As soon as it finds a line satisfying the condition notification will be sent for approval. If there is no line satisfying the condition, no notification will be send.
- *there is no case where*
  - The value specifies that the rule fires once if there are no such matches.
  - With this feature Rule will be evaluated until it finds a line satisfying the condition. Notification will be send only if there is no line satisfying the condition.
  - Should not be used with the NOT operator.
- *aggregation*
  - Aggregation functionality is used when customer need to perform operations (like Sum, Count, Maximum, Minimum, Average, Collection) on attributes of collection. When you write rule conditions that are based not only on one fact, but on many facts, you can use an aggregate. You use aggregate functions when the conditions have a view spanning multiple facts.

Using NULL checks for optional attributes

If any attribute used in the approval rules is non-mandatory on the UI then it is quite possible that for some transaction that particular is not entered by the user and in those cases SOA server will through null pointer exception and the transaction will be system rejected. For those attributes, it is highly recommended that attributes usage is accompanied by NULL check.

Miscellaneous

Task Aggregation

By definition, aggregation means combining multiple entities into fewer. In the context of approval process, approval message or notifications are getting aggregated to avoid sending multiple notifications for same transaction to same user. Following are some of the scenario where a task aggregation comes into picture:

Multiple Rules

As all approval rules in a ruleset will be executed and thus it is possible that two more rules will be evaluated to be TRUE and can result in same user for assignment. In this case instead of sending multiple notifications to the same user, system will aggregate as per the Task aggregation option and send one consolidated approval notification to approver. Following example explain this in detail:

Rule 1:

```
IF {InvoiceHeader.amount > 1000 }
THEN
{assign to supervisor}
```

Rule 2:

```
IF {InvoiceHeader.businessUnit  == USA }
THEN
{assign to Bob}
```

Now assume that user *Peter* report to *Bob* and if he submits an invoice with amount greater than 1000 and Business Unit *USA* then both of the approval rule would satisfy. If Task aggregation is enabled (not none) then instead of Bob receiving two notifications for the same invoice, he will receive only one notification.

Multiple Participants

In cases where multiple participants are enabled, it is possible that two or more participant in same stage can result in same user of assignment. If the case where participants are in parallel mode then those will be executed at same time but one consolidated notification will send out to the approver. Figure 32 shows participants 1 and 3 in parallel.

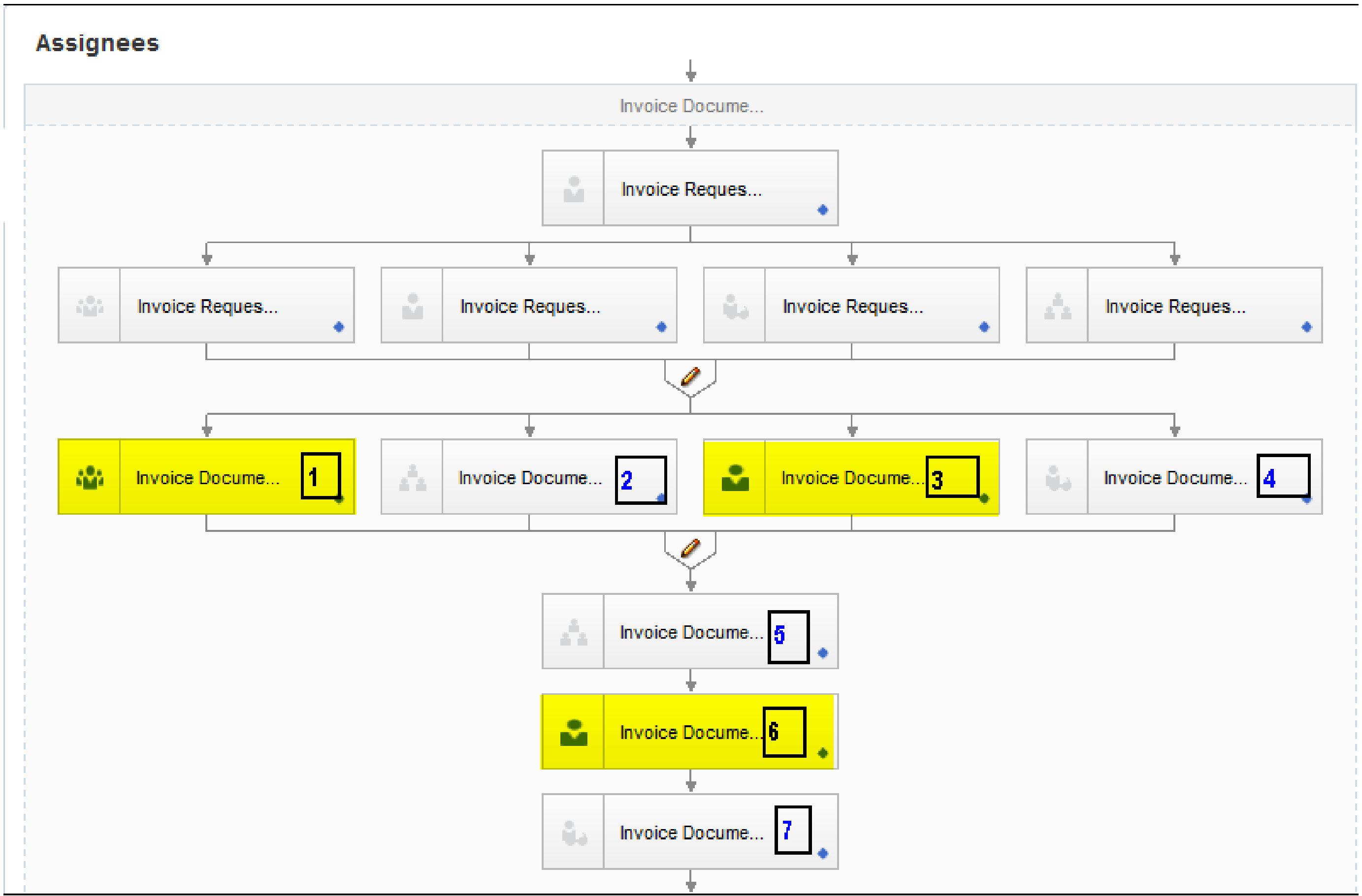

Fig 57

If the participant are in sequential mode then notification will be send in accordance to the first participant that comes in the sequence and then approval action taken in that participant will be auto-applied to the later participant. For example if user received notification as per participant 1 and user choose to approve that task. Later when the same user is evaluated in participant 6 then his/her previous action (taken in participant 1) will be auto-applied, i.e. system will stamp "Approve" action on the task and move forward. This is done to avoid asking same user multiple time for the same thing.

Multiple Stages

Similarly to rules and participants, multiple stages can also evaluate same user to receive multiple notification. If the task aggregation option is set to *once per task* then those notification will be aggregate and user will receive one notification. If it is set to *once per stage* then user will receive separate notification, one from each stage.

Repetitive Stage

A stage is said to be repetitive if *Repeat stage in parallel for each item in collection* check box is selected and appropriate collection is defined. With this, the stage will run multiple iteration in parallel based on the collection. To explain it in detail, lets assume user submits an expense report with 10 expense items from 3 different cost centers, namely Relocation, Travel and Business Expense. The business requirement is to get the expense items from each cost center to be reviewed and approved by the corresponding cost center manager. To add to it, lets assume that cost center manager for Relocation and Travel is same person.

If the task aggregation is enabled then the relocation and travel cost center manager will receive one approval request for the above expense report otherwise system will send two separate notifications, one for each cost center.

Line level attributes

A typical transaction is defined using header, line and distribution level attributes. Header is the top most object, header can contain multiple lines and lines can contain multiple distributions. In approval rule setup, customer can use attributes from any level. Following figure shows an example where if invoice amount is greater than 1000 and poDistributionid is not null and attribute1 is null then assign that invoice to ap_supv_all_bu user for review. As invoice header can contain multiple lines so lets assume that one particular invoice contain 7 lines. Out of these 7 lines, first, third and seventh line have not null poDistributionid. This mean that the rule will be executed 7 time, one for each line with evaluating to be TRUE for first, third and seventh line.

If the aggregation is enabled then these 3 lines assignments will be aggregated and ap_supv_all_bu will receive one aggregated notification instead of 3 separate notifications, one for each line.

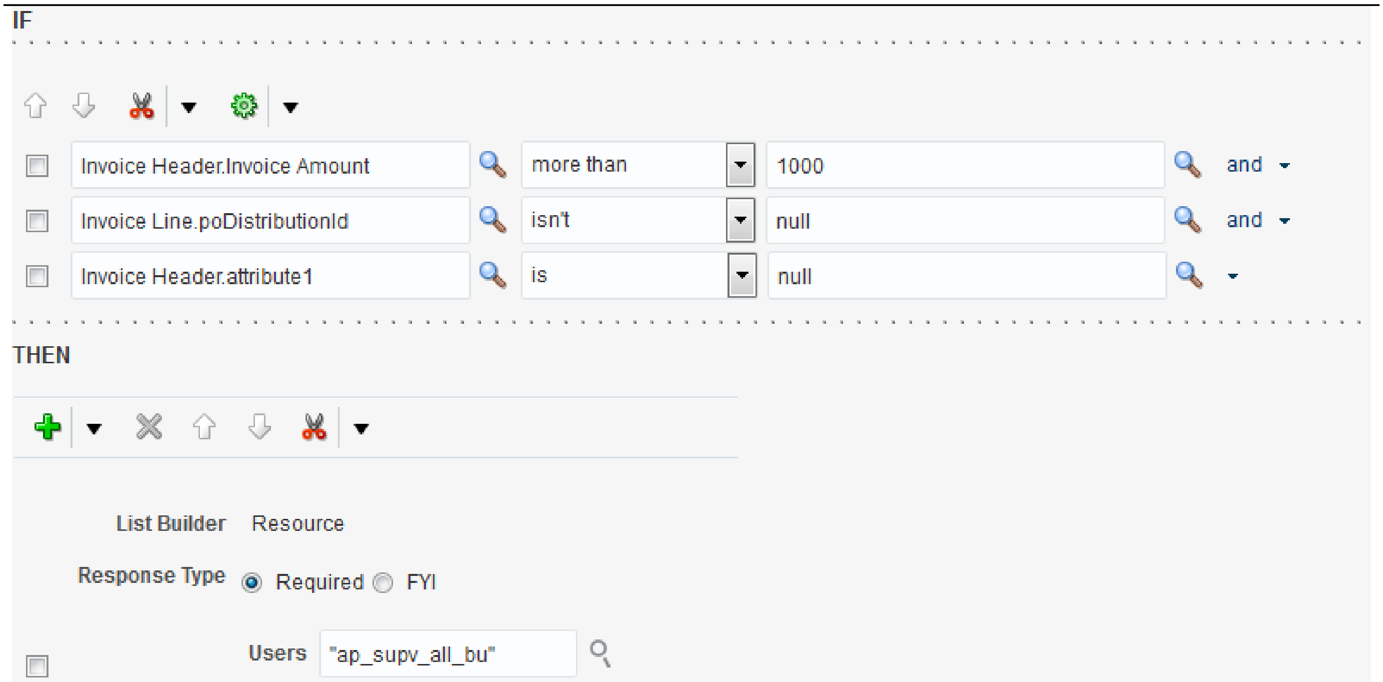

Fig 58

In-Flight Assignment

Customers have the ability to reassign or delegate the approval task to any other user. This raises following scenario:

- Reassigning task to other member of approval group / position / job hierarchy at same level.
    - In this case, the other user have already received approval notification because of original assignment and thus manual reassignment / delegation will not send new notification to the user.
- Reassigning task to user that has already taken approval action.
    - In this case, no new notification will be generated and the approval action taken previously will be stamped on the new assignment.