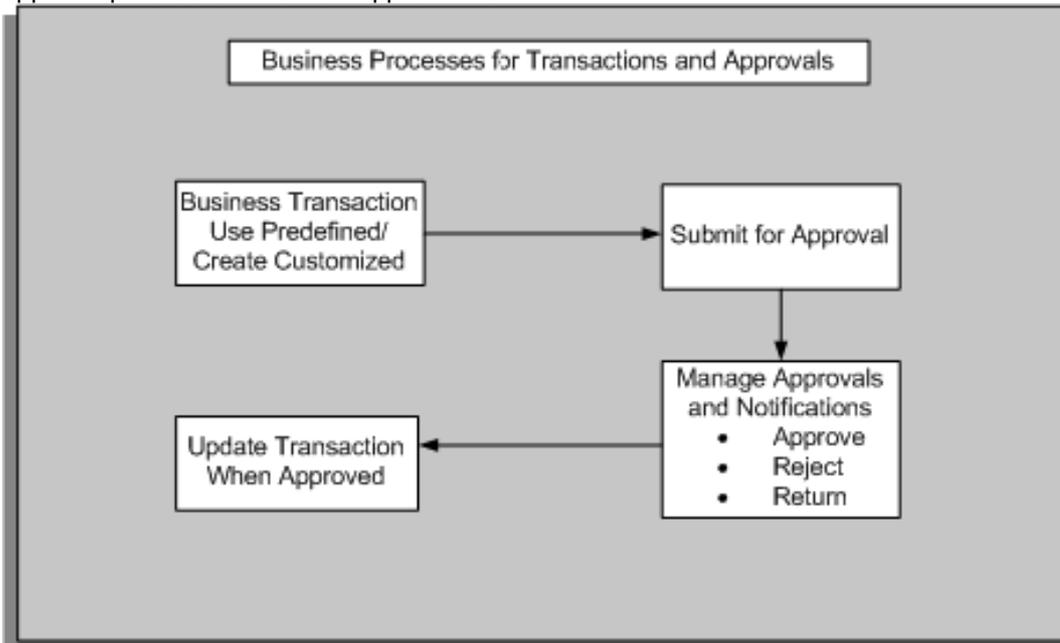




Overview of Oracle Approvals Management 1
 Implementing Oracle Approvals Management 5
 Implementing Tasks Approvals Management..... 6
 Implementing the Transaction Type 7
 Attributes 8
 Required Attributes 9
 How does AME use Attributes?..... 9
 Planning Your Test Cases (test on AME work Bench) 12
 Administration 12
 Example of Setting Approval rules 13
 Administration 14

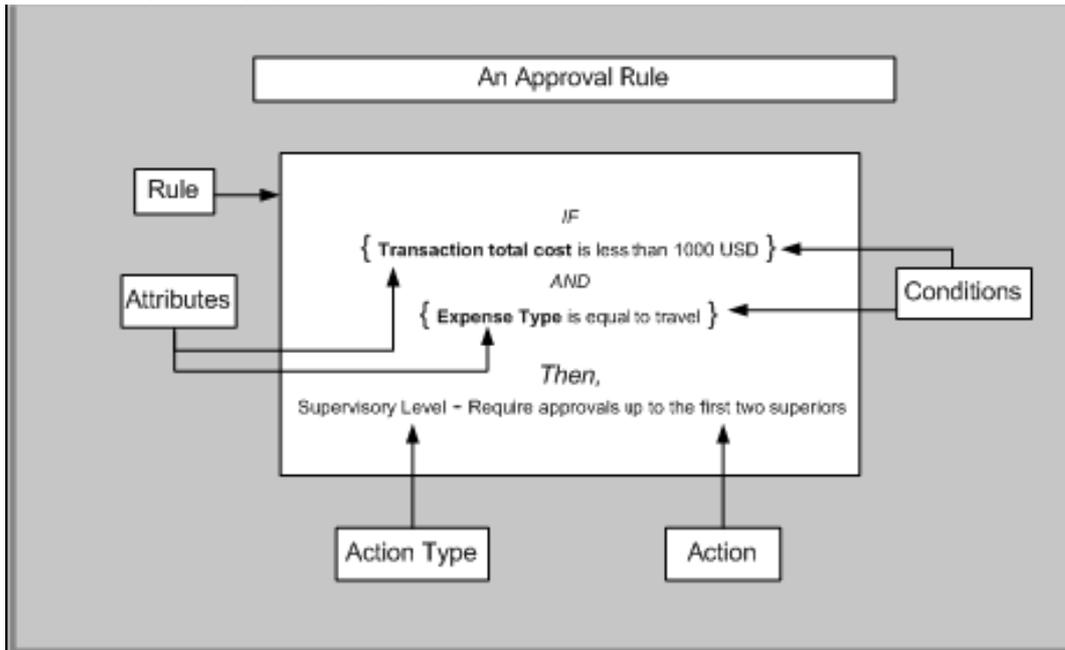
Overview of Oracle Approvals Management

The purpose of Oracle Approvals Management (AME) is to define approval rules that determine the approval processes for Oracle applications..



Approval Rules

An approval rule is a business rule that helps determine a transaction's approval process. Rules are constructed from *conditions* and *actions*.



The approval rule's *if* part consists of zero or more conditions, and its *then* part consists of one or more actions. A condition consists of a business variable (in AME, an attribute) and a set of attribute values, any one of which makes the condition true. An action tells AME to modify a transaction's approval process in some fashion

AME enables you to define rules that express a wide variety of approval rules. For example, rules that:

- Require subject-matter-expert approval
- Require managerial approval
- Create exceptions for rules requiring managerial approval
- Substitute one approver for another in special cases
- Revoke a manager's signing authority in special cases
- Grant a manager extra signing authority in special cases
- Generate a production that assigns a value to a variable name such as the value *digital certificate* to the variable name *eSignature*.
- Send for-your-information notifications.

You can prioritize the approval rules. This enables you to apply rules of sufficient priority to any given transaction.

Transaction Types

An application that uses AME to govern its transactions' approval processes is termed an integrating application. An integrating application may divide its transactions into several categories where each category requires a distinct set of approval rules. Each set of rules is called a transaction type. Different transaction types can use the same attribute name to represent values that are calculated in different ways or fetched from different places. This allows several transaction types to share approval rules (thereby implementing a uniform approval policy across multiple transaction types). A rule use occurs when a transaction type uses a particular rule for a given time period, optionally at a given priority level.

A transaction's approval process can have two components:

- List of approvers

- Set of productions

Approver Lists

A transaction's approver list has a hierarchical structure. The transaction's approver list may contain several items' approver lists. Each item's approver list may have three sub-lists. Of these sub-lists, the authority sub-list can have a chain of authority generated by one or more action types. Each approver group or chain of authority can contain multiple approvers.

Items

AME can generate an approver list for a transaction's header, and a separate approver list for each item in the transaction. A transaction type can define multiple item classes. For example, a transaction type might generate separate approver lists for each transaction's header, line items, and cost centers. All transaction types include a header item class, which always has one item (the transaction's header). All other item classes are optional.

Sub-Lists

An item's approver list may contain three sub-lists:

- Pre-chain of authority
- Authority
- Post-chain of authority

The pre- and post-chain sub-lists contain zero or more approver groups; the authority sub-list contains zero or more chains of authority.

Action Types

An action type is a set of actions having a common purpose. Each sub-list can contain approver groups or chains of authority generated by several action types. For example, actions in the absolute-job-level action type all generate chains of authority by ascending the HR supervisory hierarchy until they reach a manager with a particular job level.

The actions differ according to the job levels they require.

Approver Groups

An approver group is a collection of approvers that you define. Typically, approver groups contain subject-matter experts.

Chains of Authority

A chain of authority ascends a hierarchy of approvers that are normally defined in applications other than AME, for example HRMS (supervisor position hierarchies). The start point of the chain, and how far it ascends the hierarchy, usually varies between transactions. You can also treat an approver group as a chain of authority. In this case AME ignores the approver group's group-specific properties. Generally, chains of authority contain managers. Approver groups and chains of authority behave differently in certain circumstances.

For example, when one approver forwards a notification requesting approval to another approver. Otherwise, approver groups and chains of authority behave similarly.

Approvers

An approver has the following two properties:

• Approver Types

An approver type is any Workflow Directory Services originating system that defines entities, which can receive Workflow notifications requesting an approval. For example, the HR application defines its set of employees as a Directory Services originating system, so an HR employee can be an approver.

• Approver Categories

AME can generate approvers belonging to either of two approver categories: action and informational (for-your-information or FYI) approvers. Action approvers must approve a transaction. FYI approvers merely receive a notification describing the transaction. The exact content of all notifications depends on the application that generates the notification.

Productions

In AME, a production assigns a value to a variable name. For example, AME might generate a production that assigns the value *digital certificate* to the variable name *eSignature*. AME does not interpret the productions it generates. In fact, AME does not even define any standard production variable names. Rather, it leaves these tasks

to integrating applications and their transaction types. AME generates two kinds of productions:

- Transaction-level productions that are variable name or value pairs associated with a whole transaction.
- Approver-level productions are associated with specific approvers within a transaction's approver list.

What Happens at Run Time

Once you have defined a set of rules for a transaction type, and the application associated with the transaction type is configured to use AME, the application communicates directly with AME to manage the transaction type's approval processes. Typically, the application communicates with AME when a transaction is initiated in the application, and then each time an approver responds to the application's request for approval of the transaction, until all approvers have approved the transaction. AME records each approval, and recalculates the approver list for a transaction each time an approver responds to a request for approval of the transaction.

AME recalculates the approver list each time an approver responds. This enables AME to account for several possible circumstances that can affect a transaction's approver list:

- An attribute value changes, thereby affecting which conditions are true and so which rules apply to the transaction.
- A condition or rule is added, changed, or deleted, again affecting which rules apply to the transaction.
- A change occurs in the organizational hierarchy used by the transaction type's set of rules, thereby changing the membership of the applicable chain of authority.
- Currency exchange rates change, thereby affecting which conditions on currency attributes are true and so which rules apply to the transaction.

By accounting for such changes, AME guarantees that transactions are always approved according to the most current business data possible.

Approval Process Execution

Approval-Process Execution

Integrating applications can communicate with AME in many different ways. For example, an integrating application can ask AME for a transaction's entire approver list, for the rules satisfied, or for the set of approvers the application should notify next.

The Standard Algorithm

Typically, an integrating application follows a simple procedure for managing a transaction's approval process:

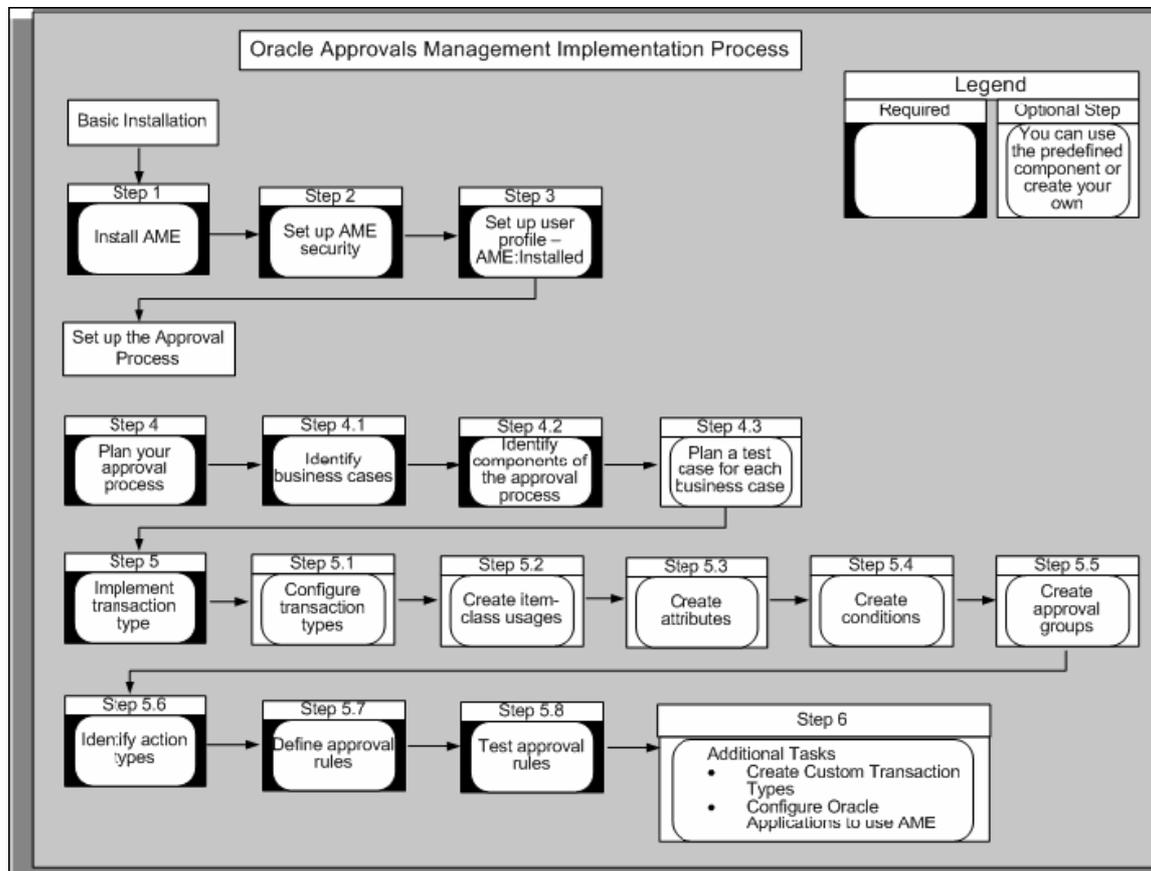
1. Ask AME for a transaction's entire approver list.
2. Display the approver list to the requestor, optionally prompting them to suppress or add approvers.
3. Communicate any approver suppressions or additions to AME.
4. Ask AME whether the transaction's approval process is complete, and if not, what approvers (if any) to notify.
5. If AME indicates that no further approvals are required, stop.
6. Notify any approvers identified by AME in step 5.
7. Wait until an approver responds to a notification.
8. Communicate the response to AME.
9. Go to step 4.

Approver-Notification Order

The order in which AME presents approvers for notification at step 4 of the standard algorithm depends on a variety of ordering modes and order numbers that together

determine a unique ordering of the approvers in a transaction's approver list. An *ordering mode* tells AME how to order the collections of approvers at a given level of the hierarchy constituting a transaction's approver list. For example, the sub-list ordering mode basically tells AME whether to notify pre-approvers at the same time as authority approvers, all other things being equal. AME typically uses ordering modes, before a transaction is submitted to AME for approval, where the number of things to be ordered is unknown. For example the approvers generated by a particular action type maybe notified sequentially or in parallel. *Order numbers* establish a fixed ordering of a collection of approvers at a given level of the hierarchy constituting a transaction's approver list, for example, the approvers in an approver group are assigned order numbers. Order numbers are not necessarily unique. Thus several approvers in an approver group can have the same order number. AME typically uses order numbers where you know the number of things to be ordered before a transaction is submitted to AME for approval.

Implementing Oracle Approvals Management



Configuration Variables

The configuration variables you should consider when implementing approvals business cases are:

- adminApprover
- allowAllApproverTypes
- allowAllItemClassRules
- allowFyiNotifications
- productionFunctionality
- purgeFrequency

- repeatedApprovers
- distributedEnvironment
- currencyConversionWindow
- rulePriorityModes
- forwardingBehaviors

Mandatory Attributes

The mandatory attributes related to implementing approvals business cases are:

- ALLOW_DELETING_RULE_GENERATED_APPROVERS
- ALLOW_REQUESTOR_APPROVAL
- AT_LEAST_ONE_RULE_MUST_APPLY
- EFFECTIVE_RULE_DATE
- EVALUATE_PRIORITIES_PER_ITEM
- REJECTION_RESPONSE
- REPEAT_SUBSTITUTIONS
- USE_RESTRICTIVE_ITEM_EVALUATION
- USE_WORKFLOW
- WORKFLOW_ITEM_KEY
- WORKFLOW_ITEM_TYPE

Approver Groups

You must create or include approver groups in your transaction type.

Rules and Rule Use If your organization has more than a few approvals rules, it can be useful to represent the rules in an approvals matrix or a decision tree. An *approvals matrix* is just a table that had one row per rule.

Implementing Tasks Approvals Management

To implement AME, you need to carry out the following steps:

1. Install the application.

AME's installation routines and administration features determine which applications can use AME. Installation and administration are typically jobs for a technical specialist. Installation is generally done only once, and administrative tasks (using AME's Administrator Dashboard) are usually only necessary to enable a new application to use AME, or to access or clear a transaction's error log.

2. Set up AME security by completing the following:

- Attach the predefined roles to a user or group of users.

AME uses the Role Based Access Model (RBAC) to provide users access to AME functions. This access model has the following predefined roles:

- Approvals Management Administrator
- Approvals Management Analyst
- Approvals Management System Viewer
- Approvals Management System Administrator
- Approvals Management Process Owner

Each of the five predefined roles has a specific set of functional grants. The grants provide users access to AME. To enable functional grants, you must assign roles to users using the User Management page. For information about accessing User Management page, see: Oracle Applications System Administrator's Guide - Security.

Set up user access as follows:

1. Login as Administrator.
2. Select the User Management responsibility.
3. Select the Users page.
4. Search for the user to whom you wish to grant AME roles.
5. In the results table, click Update. In the Update User page, you can view user details along with a list of roles available to the user.
6. Click Assign Roles.
7. Select the roles from the resulting list and click Apply.

When you assign any of the five predefined roles to a user, you are indirectly assigning AME responsibilities to the user. Grant data access to users. As AME restricts access to transaction types using Data Security, you grant users access to the transaction types using the Grants page. Set up user access as follows:

1. Login as Administrator.
2. Select the Functional Administrator responsibility.
3. Select the Grants tab.
4. Click Create Grant.
5. Select Specific User as grantee type.
6. Select the user as grantee key.
7. Select AME Transaction Types as object:
 1. All rows: This grants access to all AME Transaction Types to the User
 2. Instance: This grants access to a specific AME Transaction Type specified by the following parameters: FND_APPLICATION_ID => Application ID of the application to which the transaction type belongs
TRANSACTION_TYPE_ID => Unique identifier of the AME Transaction type within application
 3. Instance set: This grants access to one or more AME Transaction Types specified by the following parameters:
 - Use the predefined instance set AME Transaction Type Instance Set.
 - In the next page, select FND_APPLICATION_ID as Parameter1 and a wild card search string for TRANSACTION_TYPE_ID as Parameter2.
 8. In the next page there will be three options for instance type: select AME Calling Applications as the permission set.
 9. Review and Finish.

3. Set the user profile - AME:Installed

4. Configure transaction types.

An application administrator should review AME's configuration-variable values as soon as AME is installed and its security has been set up. AME has the following kinds of configuration variables:

- Single-Valued Configuration Variables AME's configuration variable *distributedEnvironment* has a single value for the entire application. This variable describe various aspects of AME's computing environment. You must set its value for AME to function properly.

- **Transaction-Type-Specific Variables**

Other AME configuration variables can have a default value, as well as a value for each transaction type. These variables are:

- adminApprover
- allowAllApproverTypes
- allowAllItemClassRules
- allowFyiNotifications
- currencyConversionWindow
- forwardingBehaviors
- productionFunctionality
- purgeFrequency
- repeatedApprovers
- rulePriorityModes

These variables determine many aspects of how AME generates a transaction type's approval processes and are similar to the mandatory attributes. The difference is, their values are always constant for all transactions in the transaction type. Ensure you are satisfied with these variables' default values before using AME.

Implementing the Transaction Type

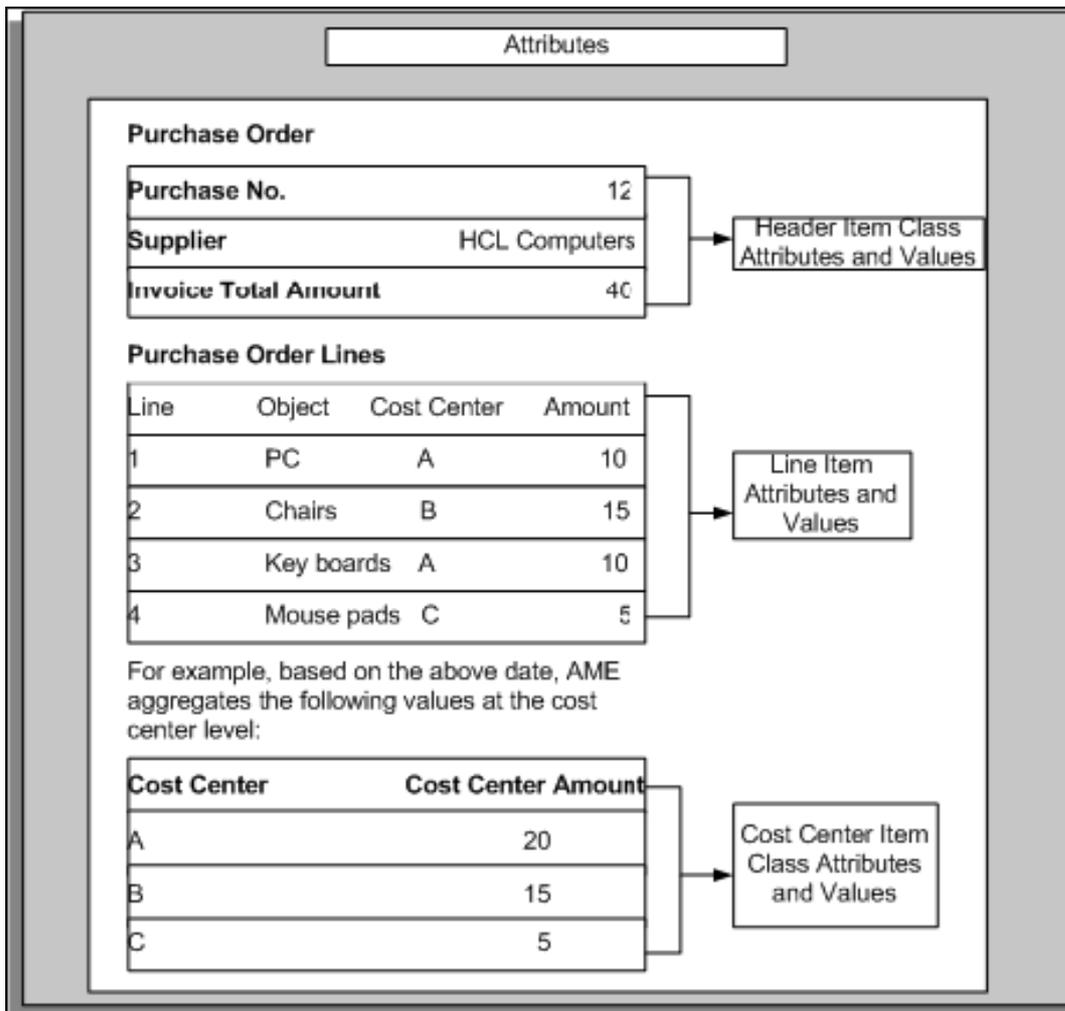
To implement the transaction type, you need to specify or create, if required, the following components of the approval process:

1. Create item-class use (Optional).
2. Create transaction attributes (Optional). In AME, an attribute is a named business variable such as TRANSACTION_AMOUNT, whose value AME fetches at run time, when it constructs transactions'approver lists.
3. Create conditions (Optional). In AME, a condition specifies a list or range of attribute values required to make a rule apply to a transaction. For example:
Example USD1000 < TRANSACTION_AMOUNT < USD5000
4. Create approver groups (Optional).
You can create AME rules to include one or more approver groups in a transaction's approver list.
5. Prepare to use the action types. You add action types to a transaction type using the Action Types tab.
 - **Predefined action and approver types**
 - **Custom action and approver types**
 - **Add approvals to existing approval types**
 - **Preparing to use the Job-Level approval types**
6. Define approval rules.
7. Test approval rules.
8. Create custom transaction types.
9. Configure Oracle applications to use AME.

Running the Approvals Management Post Upgrade Process : Management Post Upgrade Process concurrent job

Attributes

Attributes are business variables with a single value for a particular transaction.



Eg select item_quantity from some_application_table where transaction_id = :transaction

Mandatory Attributes

ALLOW_DELETING_RULE_GENERATED_APPROVERS
 ALLOW_REQUESTOR_APPROVAL
 AT_LEAST_ONE_RULE_MUST_APPLY
 REJECTION_RESPONSE
 USE_RESTRICTIVE_ITEM_EVALUATION
 USE_WORKFLOW
 WORKFLOW_ITEM_KEY
 WORKFLOW_ITEM_TYPE

Required Attributes

ALLOW_EMPTY_APPROVAL_GROUPS
 FIRST_STARTING_POINT_PERSON_ID
 INCLUDE_ALL_JOB_LEVEL_APPROVERS
 TRANSACTION_REQUESTOR_PERSON_ID

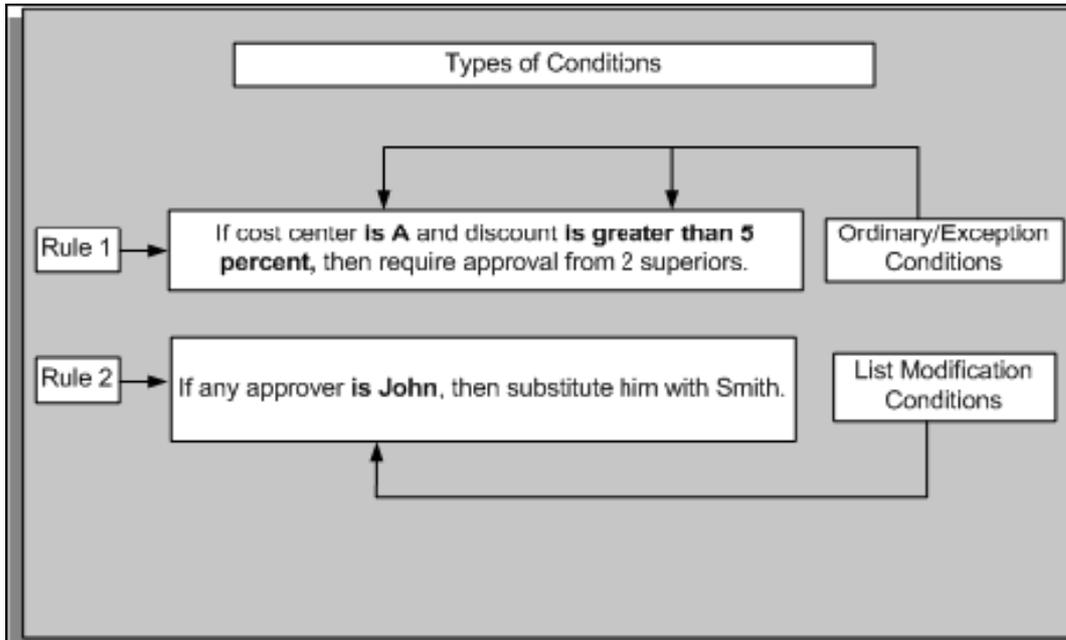
How does AME use Attributes?

When AME starts to calculate a transaction's approver list at run time, it does the following:

1. It fetches the values of each attribute that is active for the transaction type.
2. After fetching all of the active attributes' values, AME checks whether each of a transaction type's rules applies to the transaction.

Conditions

The *if* part of an approval rule consists of zero or more conditions. A condition is a statement that is either true or false, for a given transaction.

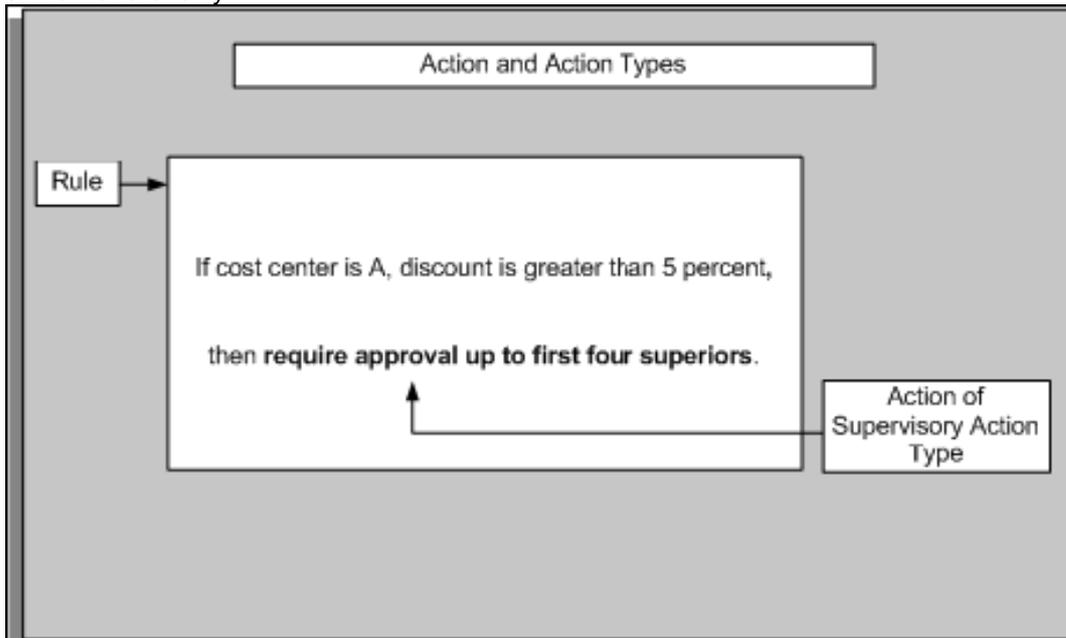


There are two types of conditions:

- Regular, which has Ordinary and Exception conditions eg $1,000 \leq \text{TRANSACTION_AMOUNT} < 2,000$
- List-modifiers : A list-modification condition checks for the presence of a given target approver at a specific set of allowed locations in the default approver list. When the target approver is present in an allowed location, we say that the default approver list satisfies the condition.

Actions

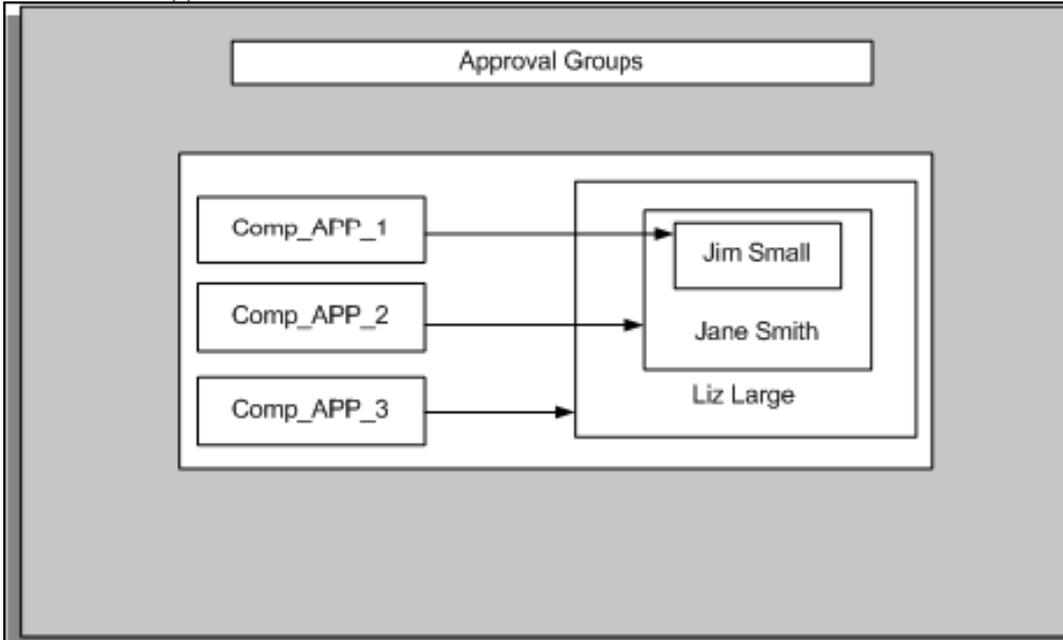
An *action* is an instruction to AME to modify a transaction's approval process in the manner you specify. Every action belongs to an action type. An action type is a collection of actions having similar functionality



Action type Properties

Name, Action-Type Handlers, Rule Types, Allowed Approver Types, Voting Methods..

Approver Groups An approver group can either be an ordered set of one or more approvers (persons and/or user accounts) or it can be a list, which is dynamically generated at rule evaluation time. A typical pre- or post-approval rule adds an approver group's members (in order) to a transaction's approver list.



Example

COMP_APP_1 = {Jim Small} COMP_APP_2 = {COMP_APP_1, Jane Smith} COMP_APP_3 = {COMP_APP_2, Liz Large}

AME would evaluate the membership of COMP_APP_3 to be (Jim Small, Jane Smith, Liz Large), in that order.

If ITEM_CATEGORY in {COMPUTER_HARDWARE} and ITEM_AMOUNT <= 1,000 USD then require post-approval from the COMP_APP_1 group.

Example

If ITEM_CATEGORY in {COMPUTER_HARDWARE} and 1,000 USD < ITEM_AMOUNT <= 10,000 USD then require post-approval from the COMP_APP_2 group.

Example

If ITEM_CATEGORY in {COMPUTER_HARDWARE} and 10,000 USD < ITEM_AMOUNT then require post-approval from the COMP_APP_3 group.

ALLOW_EMPTY_APPROVAL_GROUPS. When this attribute has the value 'true', AME allows an approver group not to have any members at run time.

when to use an approver group:

Using Approver groups for Pre- and Post-Approvals

Using Approver groups for Chain of Authority Approvals

Approver Group Properties **Name, Description, Voting Regime, Active List(static & Dynamic member list)..**

Select 'PER:'||person_id from invoice_table where transaction_id = :transactionId and item_class = :itemClass and item_id = :itemId

Rules : Creating rules and rule use are the main steps in the AME implementation process. Rule associate one or more conditions with an approval in an if-then statement. Before you can create rules, you must create conditions for the rules to use. You may need to create (or have a system administrator create) some custom attributes and/or approvals. You may also need to create some approver groups.

Rule Types : There are eight rule types in AME. Seven of them provide rules that participate in generating transactions' approver lists; these are the approver-generating rule types. The eighth generates productions.

- List-creation rules If TRANSACTION_AMOUNT < 1000 USD, then require approvals up to at least job level 2.
- List-creation exceptions If TRANSACTION_AMOUNT < 500 USD and Exception: COST_CENTER is in {0743}, then require approvals up to at least job level 1.
- List-modification rules If TRANSACTION_AMOUNT > 1000 USD and the final approver is: Kathy Mawson, then require approvals at least one level up.
- Substitutions If TRANSACTION_AMOUNT < 500 USD and CATEGORY in {MISCELLANEOUSOFFICE EXPENSES} and any approver is: John Doe, then substitute Jane Smith for the approver.
- Pre-list approval-group rules If TRANS_AMOUNT < 1000 USD and CATEGORY_NAME in {Marketing Event}, then require pre-approval from Marketing Approvals Group.
- Post-list approval-group rules
- Combination Rules
- Production Rules

Rule Properties : **Actions, Start and End Dates, Item Class**

The purpose of **rule priorities** is to prioritize a transaction type's rules and, at run time, remove from the set of rules that would otherwise apply to a transaction, those rules of insufficient priority. A *rule priority* is a positive integer associated with a rule within a transaction type.

Planning Your Test Cases (test on AME work Bench)

While planning your implementation strategy, your implementation document, showing your approval policies, should specify test cases sufficient to verify that your transaction type does the following things according to your business rules:

- Fetch correct item IDs.
- Fetch correct attribute values.
- Evaluate the rule use correctly.
- Process the rule use priorities correctly.
- Process production rules and actions correctly.
- Produce the correct default approver list.
- For real transactions, handle inserted approvers correctly.
- For real transactions, handle suppressed approvers correctly.
- Process repeated approvers correctly.
- Parallelize a transaction's approval process correctly.
- Process forwardings correctly.

Administration

The Administrator Dashboard is available only to users with the Application Administrator responsibility. You can use this dashboard to maintain AME's configuration variables and transaction types. Additionally, you can view and analyze AME's runtime exceptions. An AME application administrator has many tasks. As an application administrator, you can perform the following application and transaction-type administrative tasks:

- Set the configuration variables' default values (application-wide values)
- Set the transaction type's configuration variables' values
- Create a transaction type
- Update a transaction type
- Delete a transaction type
- View the exception log
- Clear the exception log

- Run the Setup report

Example of Setting Approval rules

Step1

is page displays the transaction types to which you have access and the rules attached to these transaction types.

Transaction Types

Name	Application	Rules	Test	Setup
AMW Audit Procedure Approval	Internal Controls Manager			
AMW Control Approval	Internal Controls Manager			
AMW Risk Approval	Internal Controls Manager			
BOM EPES Bill of Materials Creation	Bills of Material			
BOM EPES Bill of Materials Update	Bills of Material			

Recently Updated Rules

Attribute	Description	Category	Transaction Type	Rule Class	Rule Type	Active	Default
ALLOW_APPROVAL_REQUEST_APPROVAL	Whether to let the calling application (or its end users) define approvers generated by the rules.	Mandatory	State	Header	Boolean		
ALLOW_REQUEST_APPROVAL	Whether to allow requests to approve that run transactions (when the rule is on).	Mandatory	State	Header	Boolean		
ALSO_SELECT_LINE_ITEM_MUST_EXIST	Whether to require that at least one rule apply to each transaction.	Mandatory	State	Header	Boolean		
EXPIRE_ON_RULE_DATE	The date that determines which rules are active.	Mandatory	State	Header	Date		
EXPIRE_ON_TRANSACTION_DATE	Whether to evaluate rule priorities per line item under	Mandatory	State	Header	Boolean		

1. Select the transaction type
OKL CS Transfer Assumption Reque

2. Define the components
Approval Management uses these components within the approval rules.
Attributes
Define attributes to fetch business facts for a specific transaction.
Conditions
Define conditions to evaluate attributes within rules. If all conditions in a rule are true then the rule is active for the transaction.
Action Types
Enable action types to specify the action to take if a rule is active for a transaction.
Approver Groups
Define approver groups to contain approvers who are usually members of peer groups such as payroll or expenses department.

3. Define the approval rules
Approval Rules determine the approvers or FYI notification recipients required for a business transaction.
Rules

4. Test Workbench
Define test cases or test real transactions to verify the approval setup, rules and associated approvers.
Test Workbench

Step2

Attributes

This page enables you to view attributes present in your transaction type, create new, and reuse existing attributes to specify the business components of a transaction such as total amount, a percent discount, and a person's salary.

Search

Attribute Category: All
 Rule Class: All
 Date Type: All

Create New Condition

Condition Type: **Expression**

Attribute: **TRANSACTION DATE**

Expression: **RANSAC_TRANSACTION_DATE > 13-Sep-2007**

Buttons: Cancel, Create Another (1), Apply

Step3

Requ... of Type

Use Existing Action Type: Select Action Types

Select one or more action types and click Continue to use them in the current context type.

Selected a type available in OKL CS Transfer Assumption Request

Search

Rule Type: All

Select Name	Description	Rule Type	View Actions	Update	Delete
<input type="checkbox"/> chain_of_authority_level	chain of authority based on absolute level	Chain Of Authority			
<input type="checkbox"/> chain_of_authority_group	chain of authority based on approver group	Chain Of Authority			
<input type="checkbox"/> chain_of_authority_level_and_group	chain of authority based on subchains such as level and job level	Chain Of Authority			

Step4

[Details](#)
[Mandatory Attributes](#)
[Attributes](#)
[Conditions](#)
[Rules](#)
[Action Types](#)
[Approval Group](#)
[Item Class](#)
[Configuration Variables](#)

Setup Report

[+ Add transaction type](#)
[Print this Page](#)

Transaction type:

Details

Transaction Type Key	Application Name	Internal Id
No data exists.		

Mandatory Attributes [Return to Top](#)

Attribute	Usage Type	Value/Query
No data exists.		

[Attributes](#) [Return to Top](#)
[Conditions](#) [Return to Top](#)
[Rules](#) [Return to Top](#)
[Action Types](#) [Return to Top](#)
[Approval Group](#) [Return to Top](#)
[Item Class](#) [Return to Top](#)
Configuration Variables [Return to Top](#)

[General Configuration Variables](#)