Implementing, Personalizing and Customizing Oracle Approvals Management (AME) for AP Invoice Approval Routing

Gerald Jones
Solution Beacon, LLC

Introduction

Oracle defines Oracle Approvals Management, otherwise referred to as AME, as a self-service application that allows users to define business rules that govern the approval process for transactions in Oracle applications integrated with AME. In other words, the AME engine provides the opportunity for an organization to transform their unique business cases into approval routing rules for a particular transaction.

This paper will discuss how AME can be utilized to create both simple and complex business cases involving the approval of Oracle Payables (AP) invoices. It also will discuss the basic components of the AME application (i.e. attributes, conditions and rules) that are required as part of AME setup for any integrating application. Finally, it will discuss how the invoice workflow in the Oracle Payables module has been integrated with AME to drive invoice approval routings. The current AME Implementation guide provides more in-depth information on the AME engine and other advanced features which is outside the scope of this paper.

The paper assumes that the current environment is using the E-Business Suite version 11.5.10.2 (or higher) and Oracle Approvals Management patched to mini-pack B.

AME Basics

The purpose of using AME is to develop specific business cases within the framework of the application to facilitate how the approval of a transaction will routed within an application. Organization business cases regarding approval routing translate into rules within AME. Sometimes, these rules are simple. For example a simple rule regarding approval of AP invoices may require any invoice greater than $100 to be approved by the requester's immediate manager or supervisor. However, many organizations have more complex rules regarding approval routings pertaining to AP invoices. An example of a complex approval routing requirement could be an organization requiring any invoice greater $10,000 that is matched to a purchase order including computer equipment items, to be approved by the IT manager along with the requesters two immediate supervisors. The flexibility of the AME application allows a unique approval requirement such as this to be developed. Whenever approval rules are developed in AME for an application such as Payables, it identifies all of the necessary approvers for a given transaction (invoice) and notifies them through workflow notifications.

Advantages of AME

The AME application provides many advantages for the business user looking to develop approval rules for their Payables application. One such advantage is that business cases (rules) can be setup in AME without the business user having to write programming code or customizing the application. Another advantage of AME is that the approval routing can leverage some of the hierarchical structures that already exist in Oracle applications such as HR employee/supervisor or HR positions hierarchies. Additionally, if an organization has requirements for an invoice to be routed to a particular individual or

groups of individuals, AME provides the business user the ability setup specific hierarchies on which approval routings can be based.

One of the unique features of AME is its ability to respond to any changes that may occur in an organization during the approval process of a transaction. The changes can include organizational hierarchy changes (i.e. supervisor/manager), modifications to an AME rule or even changes to the values of the current transaction.

How does AME do this? Whenever an identified approver for a transaction responds with an approval, AME reconstructs a new approver list based on the most current conditions in the application including the current AME rules for the transaction. This includes the current values of the application and any changes to the approval hierarchy on which the transaction rule is based. For example, imagine the original rule in AME requires an invoice greater than $1,000 to be approved by the requester's two immediate supervisors. The approval for an invoice is initially sent to the most immediate supervisor. If a new rule is created prior to the final approval (by the second supervisor) requiring the review and approval by a tax accountant, a new approver list is built that includes the tax accountant approver after the first approval.

AME Components

In order for a business user to develop business scenarios in AME that determine approval routings, it is important to understand the different components within AME. These components are often required to be modified or created as part of the development of business cases. A brief description of these components will be discussed in the following paragraphs.

Transaction Types

A transaction type describes the type of transaction for which business cases (rules) and approval routings will be based. This can include Oracle Application transactions such as purchase orders, sales orders or accounting journals. For the sake of this discussion, the transaction type that is provided with AME is the *Payables Invoice Approval* transaction type. Oracle provides many seeded transaction types to satisfy many of the common transactions that occur within a particular application. The creation of new transaction type in AME is available for those business users that want to integrate custom applications with AME. However, Oracle does not encourage the development of new transaction types because of the significant programming effort involved to integrate with the AME application.

Attributes

Attributes within AME are business variables that represent the value of a data element of a given transaction. In the case of an AP invoice transaction, a typical attribute would be invoice amount or supplier name. Attributes can be thought of the as the 'building blocks' of business case development. The reason being is that the value of an attribute(s) for a transaction can ultimately determine whether a business case (approval rule) has been met because approval rules use conditions which in turn use attributes.

Attributes in AME can be created as being static or they can be dynamic in nature. Static attributes have a constant value that remains the same for each and every transaction associated with the attributes transaction type. Dynamic attributes use a SQL query to retrieve the value of an attribute at runtime whenever a transaction is created. Most attributes in a transaction type are dynamic.

There are several different attribute types that exist within AME. String attributes are alphanumeric in nature and can have a total length of 100 characters. Numeric attributes are considered to be any numeric value that is acceptable in PL/SQL. This includes numbers containing decimal or sign operators (+/-). AME requires that any numeric attribute that is dynamically generated to be converted to a

canonical form. This can be done by using the syntax **fnd_number.number_to_canonical** function as part of the dynamic SQL query. An example dynamic SQL query for a numerical attribute would be in the following syntax:

SELECT fnd_number.number_to_canonical(:requester_id)
FROM ap_invoices_all
WHERE invoice_id = :transactionId.

Currency attributes are used whenever the transactions of an organization involve multiple currency values. This allows for Oracle to use currency conversion between denominations when retrieving the value of an attribute. AME requires that any dynamic attribute setup as a currency attribute must include the following columns as part of the SQL query: numeric column, currency and conversion method. One caveat to mention regarding currency attributes; any AME conditions that are developed using a currency attribute must include a condition for each currency this particular transaction attribute value might have.

Boolean attributes have only two allowable values; true and false. Any dynamic attribute defined as a Boolean must return one of these two allowable results. AME provides a format string that can be used in the SQL query of a dynamic Boolean attribute.   The syntax format is in the form of either **ame_util.booleanAttributeTrue** or **ame_util.booleanAttributeFalse**.

Date attributes are commonly used on transaction data that contains a date value, such as invoice date. AME requires that date attributes be returned in the format 'YYYY:MON:DD:HH24:MI:SS'. AME provides a format string that can be used in the SQL query of a dynamic date attribute. The format string ame_util,versionDateFormatModel can be used to return the proper date format at runtime.

All transaction types currently defined in AME use several mandatory attributes that can be thought of as *runtime parameters* because they often determine various facets of AME runtime behavior.   These attributes can control AME behavior such as whether to allow an approver to appear multiple times on an approval hierarchy or whether to allow a requester to approver his/her own transactions (invoices). The following mandatory attributes are defined in AME for all transaction types:

ALLOW_DELETING_RULE_GENERATE_APPROVERS
ALLOW_REQUEST_APPROVAL
AT_LEAST_ONE_RULE_MUST_APPLY
REJECTION_RESPONSE
USE_RESTRICTIVE_ITEM_EVALUATION
EFFECTIVE_RULE_DATE
EVALUATE_PRIORITIES_PER_ITEM
USE_WORKFLOW
WORKFLOW_ITEM_KEY
WORKFLOW_ITEM_TYPE
REPEAT_SUBSTITUION

The AME Implementation guide provides additional detail on each of these mandatory attributes and how they are interpreted by AME.

Required Attributes are similar to mandatory attributes in that they determine runtime behavior of AME. The only difference being that required attributes are defined specific to a transaction type. In the case of the Payables Invoice Approval transaction types, the following require attributes are defined.

ALLOW_EMPTY_APPROVAL_GROUPS
FIRST_STARTING_POINT_PERSON_ID and SECOND_STARTING_POINT_PERSON_ID
INCLUDE_ALL_JOB_LEVEL_APPROVERS
JOB_LEVEL_NON_DEFAULT_STARTING_PERSON_POINT_ID
NON_DEFAULT_POSITION_STRUCTURE_ID

NON_DEFAULT_STARTING_POINT_POSITION_ID
SUPERVISORY_NON_DEFAULT_STARTING_POINT_PERSON_ID
TOP_POSITION_ID
TOP_SUPERVISOR_ID
TRANSACTION_REQUESTER_PERSON_ID

The AME Implementation guide provides details of each of these mandatory attributes and how they are interpreted by AME.

Both mandatory and required attributes come seeded with default values. They can be modified to meet the needs of a specific transaction type.

Conditions

The next major component of AME setup is conditions. Conditions are used to evaluate the value of attributes in a particular transaction. The result of a condition can either be true or false. Conditions are precursors to AME business rules. The result of a condition helps to determine whether a business case (rule) has been satisfied. The conditions within AME can be better thought of as the *IF* part of an approval rule. For example,

<u>*If invoice supplier is Vendor A*</u>*, then require approvals from Approver A, Approver B*

In this example, AME would retrieve and evaluate the value of the attribute *invoice_supplier* to determine if the value was equal to Vendor A.

There are three different types of conditions that exist in the AME application: Ordinary-Regular, Ordinary-Exception and List Modifier. Ordinary-Regular conditions associate an attribute with a defined value or range of values (e.g. invoice_amount > 100). Ordinary-Exception conditions are similar to Ordinary-Regular conditions in how they are defined, but differ in that they are limited to the types of rules with which they can be associated. This will be discussed later in the document. Finally, List-Modifier conditions provide the ability to create conditions based on the existence of a specific approver in an approver list that is built by AME for a specific transaction. For example, a List-Modifier condition could be defined as follows:

If Approver B is final approver, require approver up 1 level

This condition would evaluate to true if Approver B was the last approver in an approver list built by AME at runtime.

Action Types and Actions

Actions within the AME application describe the nature of what should be done in AME if a particular condition and rule is satisfied by a transaction. It is the actions that dictate the approver list that is generated by AME for the given transaction. Actions not only provide instruction as to who the approvers are, but how many approvers are requires for a given transaction and in what order should they be notified.

Action types are groupings of actions that have a similar functionality such as the approval hierarchy that should be traversed when building an approver list. An example of this would be actions that pertain to building an approval based solely on the supervisor tree in HR. The multiple actions for this action type would all pertain to traversal of the supervisor hierarchy, but would express in terms of how many levels to traverse. Typically, each action that describes how many levels of a hierarchy to move up would be defined separate unto itself. All of the defined actions would be grouped into an action type based on their relationship to the hierarchy being used.

Although the defining of a new action type is possible in the application, AME provides a broad number of action types that should satisfy most approval requirement of an organization. The action types that are defined in AME are categorized and can be grouped into four different types of hierarchies they navigate: Chain-of-Authority, List Modification, Substitution, Approver Groups & Production.

Action types and their associated actions tend to be one of the more complicated concepts to understand in AME. The following tables are intended to provide a basic understanding of action types as they are categorized in AME.

Chain-of-Authority Action Types

These action types typically utilize either the supervisor or position hierarchy defined in Oracle HR to generate an approver list for a give transaction.

Action Type	Description	Example
Absolute-job-level	Ascends the HR supervisor hierarchy until an approver with the appropriate job level is found.	Require approvals up to job level 6
Relative-job-level	Ascends the HR supervisor hierarchy until an approver with a number of job level above the job level of the requester of a transaction	Require approvals at least 4 levels up
Manager-the-Final-Approver	Ascends the HR supervisor hierarchy, but only the immediate supervisor and final approver on approval list are require to approve	Require approval up to first supervisor and CEO only
Final-Approver-Only	Ascends the HR supervisor hierarchy, but only requires approval from the person that is last on the approver list	Require approval from division manager only
Dual-Chains-of –Authority	Ascends the HR supervisor hierarchy and builds two separate list chain of approvers. Must provide approval action for each chain.	Require approval from previous employee supervisor and current employee supervisor (i.e. during employee transfer)
Line-Item-Job-Level	Ascends the HR supervisor hierarchy. Enables approval chains to be built based on line level item in a transaction.	Require approval from manager of an accounting code cost center distribution segment on an invoice distribution line.
Supervisory-Level	Ascends the HR supervisor hierarchy based on a fixed number of required approvers. There is no correlation or dependency on job level.	Require approvals up to 3 supervisors

Action Type	Description	Example
HR-Position	Ascends the HR position hierarchy up to a specified position.	Require approvals up Accounting Manager
Position-Level	Ascends the HR position hierarchy up a specified number of positions.	Require approvals up to position at level 4

List-Modification Action Types

Like the Chain-Of-Authority action type, the List-Modification action type also traverses the HR supervisor hierarchy structure. However, its intent is to modify the inherit approver list by either granting or revoking authority limits. It accomplishes this by either extending or removing individuals from an approver chain that might normally be built for a transaction. This modification of the approver chain is based on the target approver specified when using the action type. An example of this would be allowing an IT manager be the final approver of a computer purchase even though the amount of the transaction might normally require approval up to say the CFO level. However, because the IT manager may be more knowledgeable of the purchase, the approval list is shortened to allow them to be the final approver of the transaction.

Action Type	Description	Example
Final-Authority	Grants final authority to an approver that typically does not have signing authority by ending the approval chain when the approver chain reaches the designated approver.	If Approver is IT Manager, allow final approval
Non-Final Authority	Revokes final approval authority to an approver that normal has sufficient signing authority by extending an approval chain beyond the final approver until a targeted or designated approver is reached.	If Approver is Harry, require approval up to one supervisor

Substitution Action Types

Substitution action types allow for a target approver to be replaced by another designated approver whenever a transaction generates an approval chain involving the target approver. An example would be routing approvals to an employee in the absence of another.

Approver Group Action Types

Approver group action types allow an approver list to be built based on a predefined list of members that are part of the approval group. This list can be static, containing the specific names of individuals responsible for approvals of a given transaction. The list can also be dynamically build based on a SQL statement.

Action Type	Description	Example
Approver-Group Chain-of-Authority	Chain-of-authority list is built in the same manner as the List-Creation or List-modification action types. However, instead of using the HR supervisor or position structure, it uses a predefined approver group list to generate the chain-of-authority	If transaction item contains building material, require approvals from Mark, then Chris and then Sharon.
Pre and Post Chain-Of-Authority	Inserts an approver list either before or after the normal approver chain that might be generated for a given transaction.	If transaction includes sales tax, require approval by Sales Tax group, then require approval up to job level 3

Rules

Rules could arguably be considered the essential component when defining business cases for an organization. Whenever a transaction is initiated, the rules are evaluated to determine what approval path the transaction will follow when submitted for approval. Rules are defined by associating the conditions that ultimately determine whether a particular business case has been satisfied. Additionally, rules are associated with action types to determine the approval action and thus the approver list that should be generated for the transaction.

Similar to action types, rules use rule types to determine the type list to build if the results of the rule are true. The following table briefly explains the current rule types for the Payables Invoice Approval transaction type in AME

Rule Type	Description
List-Creation	Builds a chain-of-authority list of approvers that ascend some organizational hierarchy to generate the chain of approvers.
List-Creation Exceptions	Builds a chain-of-authority list of approvers as well. However, it is used often used to suppress a list-creation rule so as to require approvals from a certain group of approvers if an additional condition is met. An example is a rule to force approval by a particular group of approvers based on cost center of an invoice distribution account.
List Modification	Allows for rule to grant or revoke final approval based on a transaction's condition.
Substitution	Allows for delegation of an approval authority to another approver. Example is when someone is on vacation and they want to assign an individual to handle their approvals.
Pre/Post Approval Rules	Allows for additional approvers outside of a transaction's generated chain-of-authority to be added to the list of approvers
Combination Rules	Combines actions from action types having different rule types. Often used when several rules apply to the same business transaction.
Production Rules	Outputs runtime values to a particular integrating transaction.

Invoice Approval Workflow and AME

So how does Oracle Payables integrate with Oracle Approvals Management? The simplest answer is to mention that they integrate through use of the AP Invoice Approval workflow. Whenever AP is configured to use workflow, all invoices (manual and imported) are subject to invoice approval. This is done by initially setting the approval status of the invoice to *Required*. Once the invoice is validated and approval is initiated for the invoice either online or via the *Invoice Approval Workflow* concurrent program, the invoice falls into the workflow cycle. The approval logic can best be explained by reviewing the Invoice Approval workflow.
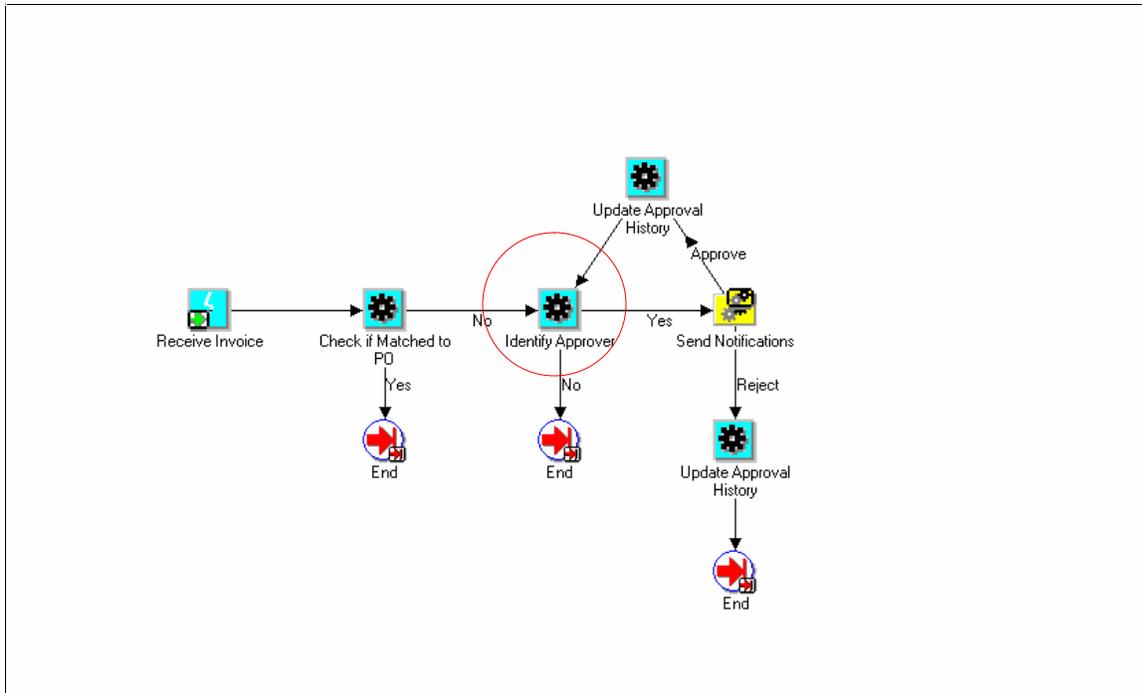
Approval Logic

When an invoice transaction falls into the approval workflow, the workflow determines if the invoice transaction is fully matched to a purchase order. If it is, then the workflow ends and the approval status of the invoice is updated to *Not Required.* However, if the invoice is not matched to a purchase order, then the workflow tries to identify the first or next individual responsible for review and approval of the invoice. The workflow node *Identify Approver* is where AP and AME are integrated. It is at this point that the workflow calls AME to determine either a) does the invoice initially meets any of the currently defined rules in AME for invoice approvals or b) are there any additional approvers left on the approval chain hierarchy.

For any AME rule satisfied by the invoice transaction, AME attempts to build an approver list based on the applied rule and the associated action type and actions that define the appropriate approvers. If a successful approver list is built, then the workflow sends a notification to the first approver in the list. The workflow itself remains active and continues to call AME as long as:

There are more approvers left on the approver chain
The workflow has not been rejected by any approver
The workflow has not expired due to non-responses by an approver

As you can see, the components that are defined in AME, especially the business rules have a direct impact on the approval routings within AP invoice workflow. It is very important to plan and define your rules carefully to ensure that the organizational approval requirements are met and approval routings flow as intended.

One thing that is important to note is the behavior of AME and workflow for invoices that do not satisfy any predefined rule. By default, the approval status of any new invoices is set to 'Required'. One the invoice is sent for approval either manually by the user online or via the Invoice Approval Workflow program, the approval status of the invoice changes to 'Initiated'. When the workflow begins, if the invoice transaction does not initially satisfy any approval rules in AME, the workflow ends and the status of the invoice remains 'Initiated'. This is the behavior of the AP Invoice workflow delivered with Oracle. It is important to mention this because whenever an organization decides to require approvals of invoices, the invoices cannot be paid until the invoice is approved. So for any invoices that fall into the category of not satisfying any approval rules, this could potentially prevent these invoices from ever being paid. There are a couple of alternatives an organization could choose to resolve this issue. The first of which is to modify the AP Invoice workflow to deal with any invoices that do not initially meet the conditions of an approval rule. Modification to the workflow is beyond the scope of this document. The second alternative has to do with the setting of the mandatory attribute *AT_LEAST_ONE_RULE_MUST_APPLY*. Setting the value of this attribute to True will cause the workflow to raise an exception for any invoice transactions that do not satisfy at least one defined rule. In this case, an organization could at least be aware that their rules defined in AME do not cover any all business cases that exist in regards to invoices transactions.

Implementing AME for AP Invoice Approval

In order to use AME to facilitate AP invoice approval routings, there are some setup steps that must be completed in both the AME application as well as within the Payables applications.

AME Setup

The first step in implementing AME is to install the AME application. As of release 11.5.9, AME comes seeded and is installed as part of the overall applications install. The setups that are described in this document are for the latest version of the AME applications. (As of this writing, AME.B is the latest RUP (Roll-Up Patch) available).

You can find the most recent patch for AME at site by using the Simple Search function under the Patches & Updates tab on Metalink.

The next step in setting up the application is to set up AME security. The current version of AME uses Oracle Role Based Access Model (RBAC) which is part of the new User Management model to provide access to the various AME component functions. An AME role must be attached to the user account of any person utilizing AME to develop application business rules. The following roles are predefined in AME.B.

Approvals Management Administrator
Approvals Management Analyst
Approvals Management System Viewer
Approvals Management System Administrator
Approvals Management Process Owner
Approvals Management Business Analyst[1]

Granting a role to a user does not automatically provide access to the setup components within AME. As part of the RBAC model, once a role has been granted to a user, specific access must be granted, to access functions within the role in order to 'activate' access to the functions. In terms of AME, this means granting access to either all or a specific transaction type. For example, if a business user is responsible only for the setup of the *Payables Invoice Approval* transaction type, then a specific access to this transaction type can be granted, thus allowing the user to only access and modify components of this one transaction type.

---

[1] This role comes inherit will access to all of the components needed to develop AME rules. This includes attributes, conditions, rules and the testing workbench.

After the roles are granted and access is established, the profile option *AME:Installed* must be set at the application level for *Payables.* This profile option change can be made under the System Administrator responsibility in the applications.
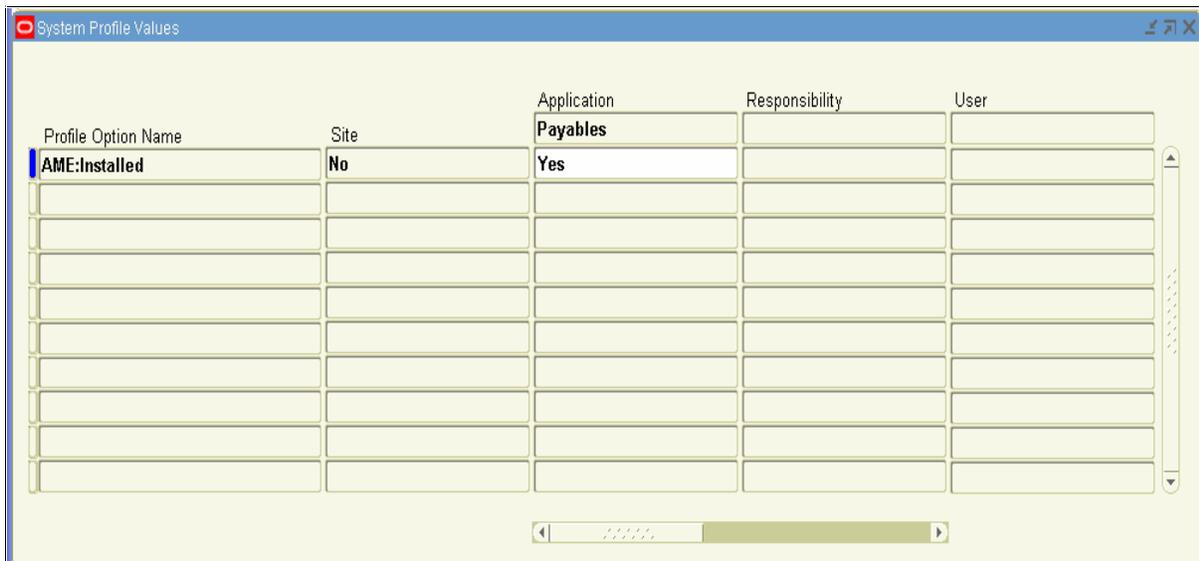
AP Setup

In addition to the setup steps that must be followed in the AME application, there are some additional steps that must be done in the Payables application to enable AP invoice approval routing. Fortunately, all of these setups are done from one form within the application module, the Payables Options form.
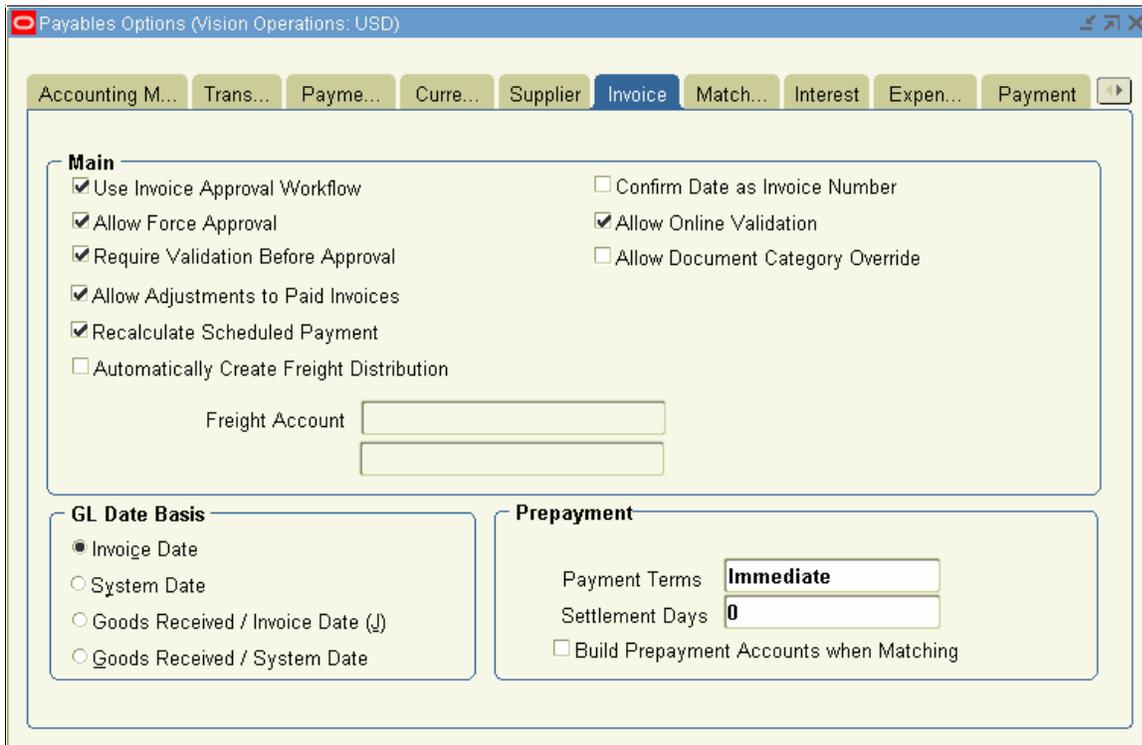
The Payable Options form is typically located under the Payables manager or equivalent responsibility in the applications.

There are three options on this form that dictate how Invoice Approval is facilitated in the Payables applications.

The first option *Use Invoice Approval Workflow* is the primary option because it informs the Payables application to force all invoices to go through the invoice approval workflow. As mentioned previously, when this option is enabled, all invoices are set to *Required* and must initially fall into the workflow cycle. The next option is *Allow Force Approval*. This allows a user to automatically set the approval status of an invoice to *Approved*, which allows an invoice to be automatically approved without having go through the workflow cycle. The last option, *Require Validation Before Approval* requires that an invoice be fully validated before it can be placed in the workflow approval cycle.

Defining Business Case Scenarios

In the current version of the AME application (AME.B), the Approvals Management Business Analyst role provides access to the Business Analyst Dashboard

The dashboard can be thought of as a 'birds eye view' of the AME application.
Along with displaying an overview of the various transaction types that are currently defined, the dashboard also displays any rules that have recently been defined, updated or deleted along with any

rules that are slated to become active at a future date. More importantly, the dashboard provides links to all of the setup components required when defining new business case rules in AME, including attributes, conditions, approver groups and rules.

Whenever a business user begins the process of defining rules that represent organization business cases, it is important to have an understanding of the transaction type of which business rules will be based. As part of this understanding, a user should determine two important elements of the transaction type:

- What does the transaction type's transaction id represent?
- How does the transaction type determine the requester of a transaction?

The answer to the first question would require some research (i.e. Metalink, Application specific guides, etc.) to discover what value in a particular transaction is used to represent the transaction id. In the case of the *Payables Invoice Approval* transaction type, the invoice_id in AP_INVOICES_ALL is used as the transaction type. The importance of knowing the value of the transaction id lies in the fact that most of the dynamic attributes use the transaction id as part of the WHERE clause of the SQL statement used to retrieve the their value. Remember, an attribute must return a single value. In the case of invoice transaction, using the invoice_id will ensure that a single value will be retrieved.

As far as determining the requester initiating a transaction, there is a required dynamic attribute defined for most if not all transaction types that contains the logic to retrieve this value. The required attribute is *TRANSACTION_REQUESTOR_PERSON_ID*. In the Payables Invoice Approval transaction type, the value of this attribute is retrieved by the following SELECT statement:

```
select requester_id
from ap_invoices_all
where invoice_id = :transactionId
```

This means that the person populated in the requester field on the invoice header in Payables will be flagged as the initiator of a transaction. Any approver lists that are built from an invoice transaction will begin using the requester id as the basis.

For each of the following business case demonstrations, the paper assumes the HR supervisor hierarchy is used as the basis for building approval lists. Additionally, there is the assumption that only one currency (USD) is used.

Business Case # 1: Require Approvals up 1 level from invoice requester for any invoice $100 or greater.

For this demonstration, the following components need to be defined:

Attributes: Total Invoice Amount
Condition: Total Invoice Amount >= $100
Rule: If Total Invoice Amount >= $100, then require approvals up to the first supervisor

Attribute: Total Invoice Amount (SB_INVOICE_AMT)

Since this is an amount field, the SQL statement must return the number using the fnd_number_to_canonical function.

Condition: Total Invoice Amount >= $100 (SB_INVOICE_AMT is greater than or equal to 100)

Any condition that uses a numeric attribute as its basis must provide a lower and/or upper limit. In the business case, we are only concerned with invoices that are $100 or more.

Rule: Total Invoice Amount >= $100, then require approvals up to the first supervisor (SB Invoice Rule (> $100)

Several items on the figure above are highlighted to show that the rule is consistent with the original business case requirement to force an approval by the immediate supervisor of the requester for any invoice $100 or greater.

Business Case # 2: Require approvals from the Tax Approver group, along with the normal chain-of-authority for any invoices greater than $100 with Sales Tax distribution Lines.

For this demonstration, the following components need to be defined:

Attributes: Sales Tax Present on Invoice
        Total Invoice Amount (Already defined)
Conditions: Sales Tax Present on Invoice > 0
        Total Invoice Amount >= $100 (Already defined)
Approver Group: Sales Tax Group
Rule: If Sales Tax Present on Invoice > 0, then require pre-approval from the sales tax approver group, then up to the first supervisor

Attribute: Sales Tax Present on Invoice (SB_SALES_TAX_PRESENT)

This attribute could have been defined as a Boolean attribute to return either True or False. For simplicity it has been defined as a numeric field that returns the # of invoice distribution lines that are line type 'TAX' with a tax code equal to 'SALES TAX'. If the count returned is zero, then it is assumed the invoice has no sales tax lines. Any value greater than zero is assuming the invoice has at least one or more sales tax lines.

Condition: Sales Tax Present on Invoice > 0 (SB_SALES_TAX_PRESENT is greater than 0)

As mentioned in the previous example, any condition that is associated with a numeric attribute must define a lower and/or upper limit. In this case, we define a lower limit that the value retrieved in the SB_SALES_TAX_PRESENT attribute must be greater than 0 for the condition to be considered true.

Approver Group: Sales Tax Group (SB Sales Tax Group)

Important to remember is that an approver group allows an organization to setup hierarchies that include specific individuals required to be included in an approval list. This approver group has been defined to include one employee in the approver group to represent the sales tax group. Additional people can be added or removed as needed.

Rule: If Sales Tax Present on Invoice > 0, then require pre-approval from the sales tax approver group, then up to the first supervisor

The definition of this rule really demonstrates the flexibility of the AME application when defining complex or unique approvals. There are a couple of items worth noting regarding this setup.

The first of which is the rule type associated with the rule which is a *combination* rule type. This important to note because a combination rule type allows a rule to include different action types that may use different approval types as discussed earlier in the document. For this rule, a combination rule type was necessary to allow the pre-approval group to be notified (Sales Tax Group) prior to the notifying the immediate supervisor of the invoice requester.

The other item worth noting is that rule can use as many conditions as necessary to satisfy the most complex or unique approval requirements of an organization. For this rule, notice that we are using both the most recently defined condition that counts the number of sales tax distributions lines, but also uses the previously defined condition that verifies whether the invoice amount is greater than $100.

AME Testing Workbench

One of the very powerful features of the AME application is the Testing Workbench. The workbench provides the ability to test the business rules that have been defined in AME against test or *real* transactions in your Payables application database. This allows you to preview the results of your AME definitions to verify certain aspects such as:

* Are attribute values, particularly custom attributes retrieving values correctly?
* Does the invoice satisfy the appropriate rule?
* Is the proper approver chain(s) being generated for the transaction based on the rule chosen?

The testing workbench can be accessed from the AME Dashboard. The AME Dashboard can be found under the Approvals Management Business Analyst role discussed earlier in the document.

The first step in using the Test Workbench involves defining a new test case in AME. Defining a test case is simple as it involves supplying a name for the test case and description (optional).

After entering a name for the test case and description, choose the *Save for Later* button to save the test case definition. Although a *Run Test Case* button is available at this point, it is best to save the definition to the database first and then execute a test case after receiving the confirmation page below.

The best approach to demonstrating the AME Test Workbench is by entering a new invoice in the Payables application and executing a test against this invoice. The invoice will be created to test the second business case rule defined earlier in the document. Our test case will allow us to verify whether our business case rule has been defined properly and if the approver list is built correctly by AME.

The first step to test our business case is to create an invoice in Oracle Payables. The invoice has been created as follows.

The invoice has been created with a total amount of $160.88. The invoice has an identified requester and one of the distributions lines contains sales tax. In regards to the second business case defined in AME, this invoice should be sent for approval to any members in the tax approver group and then to the supervisor of the requester (Jones, Gerald K). In the HR application, Susan Behn has been assigned as Geralds' supervisor.

Now that an invoice has been created, we can execute a test from the workbench to see if our AME components have been defined correctly and produce the results we expect. To execute a test against this invoice, navigate to the testing workbench. From the workbench, choose a test case against which the testing will be conducted. Choose the *Run Real Transaction Test* button.

The next screen prompts the user for the transaction id which AME uses to evaluate previously defined rules and generates an approver list. This transaction id as mentioned previously is the invoice_id from AP_INVOICES_ALL

It is important to note that upon entering a transaction id, you must choose the *Go* button which retrieves information about your transaction. In particular, it retrieves the values of all attributes that have been defined for the current transaction type. The values of the header attributes are shown on the next page to demonstrate how AME retrieves and displays the values of the attributes. These values are consistent with the data entered for your invoice transaction.

SUPPLIER_INVOICE_ATTRIBUTE15		
SUPPLIER_INVOICE_ATTRIBUTE2		
SUPPLIER_INVOICE_ATTRIBUTE3		
SUPPLIER_INVOICE_ATTRIBUTE4		
SUPPLIER_INVOICE_ATTRIBUTE5		
SUPPLIER_INVOICE_ATTRIBUTE6		
SUPPLIER_INVOICE_ATTRIBUTE7		
SUPPLIER_INVOICE_ATTRIBUTE8		
SUPPLIER_INVOICE_ATTRIBUTE9		
SUPPLIER_INVOICE_BASE_AMOUNT		
SUPPLIER_INVOICE_CREATION_DATE	27-Feb-2007	
SUPPLIER_INVOICE_CURRENCY_CODE	USD	
SUPPLIER_INVOICE_DATE	27-Feb-2007	
SUPPLIER_INVOICE_DESCRIPTION	AME Workbench Test	
SUPPLIER_INVOICE_EXCHANGE_RATE_TYPE		
SUPPLIER_INVOICE_EXPENDITURE_ORGANIZATION_NAME		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE1		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE10		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE11		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE12		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE13		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE14		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE15		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE16		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE17		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE18		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE19		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE2		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE20		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE3		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE4		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE5		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE6		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE7		
SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE8		

		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE19		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE2		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE20		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE3		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE4		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE5		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE6		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE7		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE8		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE9		
		SUPPLIER_INVOICE_GLOBAL_ATTRIBUTE_CATEGORY		
		SUPPLIER_INVOICE_GOODS_RECEIVED_DATE		
		SUPPLIER_INVOICE_LAST_UPDATE_DATE	27-Feb-2007	
		SUPPLIER_INVOICE_NUM	AMETEST	
		SUPPLIER_INVOICE_PAYMENT_CURRENCY_CODE	USD	
		SUPPLIER_INVOICE_PAYMENT_METHOD_LOOKUP_CODE	CHECK	
		SUPPLIER_INVOICE_PAY_GROUP_LOOKUP_CODE	Standard	
		SUPPLIER_INVOICE_PROJECT_RELATED	N	
		SUPPLIER_INVOICE_RECEIVED_DATE		
		SUPPLIER_INVOICE_SOURCE	Manual Invoice Entry	
		SUPPLIER_INVOICE_TYPE_LOOKUP_CODE	STANDARD	
		SUPPLIER_INVOICE_USSGL_TRANSACTION_CODE		
		SUPPLIER_INVOICE_USSGL_TRX_CODE_CONTEXT		
		SUPPLIER_INVOICE_VAT_CODE		
		TOP_SUPERVISOR_PERSON_ID		
		TRANSACTION_DATE		
		TRANSACTION_GROUP_ID		
		TRANSACTION_ORG_ID	204	
		TRANSACTION_REQUESTOR_PERSON_ID	Jones, Gerald K	
		TRANSACTION_REQUESTOR_USER_ID		
		TRANSACTION_SET_OF_BOOKS_ID	1	
⊕	▶ Line Item Attributes			

(Cancel) (Save for Later (1)) (Run Test Case (2))

After reviewing the values retrieved for the various attributes of the transaction, choose the *Run Test Case* button to execute and evaluate the rules and action defined for the transaction.

ORACLE Approvals Management

Return to Dashboard   Home   Logout   Preferences   Help   Personalize Page

Transaction Type: Payables Invoice Approval

Rules | Test Workbench | Setup

Test Workbench > Real Transaction Test >

Run Real Transaction

To review the process stages that Approvals Management took to arrive at the applicable rules and approver list, click View Approval Process Stages.   (View Approval Process Stages (1))

Transaction ID   **64007**

Applicable Rules

Details	Description	Class	RuleType	Category
▶Show	SB Sales Tax Approval Rule	Header	Combination	Approver

Final Processed Approver List

Details	Order Number	Approver Type	Approver	Category	Status
▶Show	1	HR People	Berry-Jones, Halle	Approver	
▶Show	2	HR People	Behn, Susan	Approver	

Return to Test Workbench   (View Approval Process Stages (1))

Rules | Test Workbench | Setup | Return to Dashboard | Home | Logout | Preferences | Help | Personalize Page

About this Page   Privacy Statement

Based on the results of the test, it appears that the business test case was defined properly. The SB Sales Tax Approval Rule was applied to this transaction because the invoice had a sales tax distribution line. Additionally, the approval list has been built correctly. The first approver is the individual included in the Tax Approver group defined earlier in the test case. The next approver in the approval chain is the supervisor of the requester of the invoice.

As previously stated, the Testing Workbench can be a very useful during the process of implementing and developing AME rules for invoice approval routing. Having the ability to evaluate and see the results of your AME setups using *real* transaction prior to implementation in a production environment is quite valuable.

Conclusion

It was the intent of this paper to provide the reader with enough high level understanding of AME functionality and how organizations can use AME to control their approval requirements for invoice approval routing in Oracle Payables. As with any other Oracle application, mastery of the application comes through practice and experimentation. Hopefully, the paper has demonstrated how thorough planning of business case rules and further understanding of AME can allow business users to develop their most unique or complex approval requirements in this powerful application.

Bibliography

1. *Oracle Approvals Management Implementation Guide Rel 11i* Part No. B25324-01, Metalink Note 336901.1
2. *Oracle Approvals Management Developers Guide Release AME.B*, Metalink Note 289927.1
3. *How to Implement the Invoice Approval Workflow*, Metalink Note 269057.1
4. *Oracle Payables Users Guide 11i* Part No. A81180-07
5. *Oracle Approvals Management Not Enabled? What does it take to Enabled It?*, Metalink Note 413300.1
6. *About Oracle Approvals Management Minipack B*, Metalink Note 336004.1