

## AME - Approval Management Engine

Opis .....	1
Główne idee .....	1
Atrybuty .....	1
Warunki (kiedy reguła ma zastosowanie) .....	1
Grupy osób (jakie osoby wówczas zatwierdzają) .....	1
Reguły .....	3
API .....	4
Zmienna pomocnicza .....	4
Zainicjowanie API .....	4
Pobranie pierwszego / kolejnego zatwierdzającego .....	4
Przykłady .....	5
W razie problemów .....	5
Zagadnienia otwarte .....	6

## Opis

**AME** to generyczny mechanizm budowania ścieżek powiadomień.

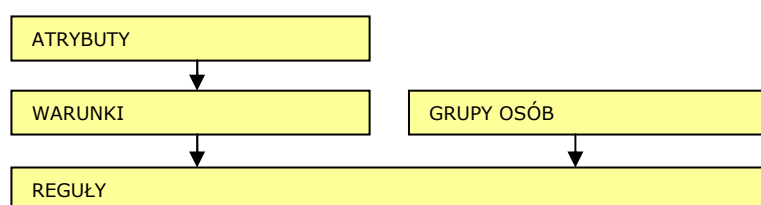
Mechanizm wykorzystują m.in. moduł zobowiązania ( workflow APINV ), moduł GMD ( workflowy GMD%), expenses, zaopatrzenie w wersji >=11.5.10.

**AME** składa się z następujących elementów:

1. graficzny interfejs użytkownika do budowania ścieżek, zob. autoryzacja **Zarządzanie Zatwierdzeniami - analityk biznesowy** (Approvals Management Business Analyst)
2. graficzny interfejs użytkownika do testowania ścieżek. Umożliwia on zarówno podanie wartości atrybutów jak również podanie określonego id, dla którego wartości atrybutów zostaną wyliczone ( tzw. test na danych rzeczywistych ). Umożliwia podgląd ścieżki przed i po scaleniu ( ścieżka przed może zawierać powtarzające się nazwiska – wówczas zostaje tylko pierwsze wystąpienie ).
3. API do pobierania osób ze ścieżki zatwierzeń

## Główne idee

Ścieżki budowane są w oparciu o 4 komponenty:



## Atrybuty

- ➔ Predefiniowane + można tworzyć własne
- ➔ Statyczne / dynamiczne ( SQL ) – używaj zmiennej **:transactionId**, aby odwołać się do wartości atrybutu **event.\_key** procesu workflow

## Warunki (kiedy reguła ma zastosowanie)

- ➔ Zbudowane są z atrybutów.
- ➔ Ograniczenia: nie można porównać dwóch warunków, nie ma NULL, wszystkie warunki są AND ( nie ma OR ), dla stringów dostępny jest tylko operator IN (nie ma like '%'). Większość z tych ograniczeń można obejść przez odpowiednie zbudowanie atrybutów (w definicji atrybutów dynamicznych można używać dowolnej składni SQL).
- ➔ Warunki są krostypowe (dziwne ale tak jest)

## Grupy osób (jakie osoby wówczas zatwierdzają)

- ➔ Szeregowo/równoległe (trzeba też wybrać odpowiedni typ czynności dla reguły)

- ➔ Statyczne/dynamiczne ( SQL – tu możemy określić typ osoby zatwierdzającej: USER\_ID lub PERSON\_ID w nast. Sposób: select 'person\_id'||employee\_id from fnd\_user
- ➔ Grupy mogą się zagnieżdzać
- ➔ Użytkownik BOA = użytkownik bez przypisanego pracownika
- ➔ Również w tym miejscu możesz używać zmiennej :transactionId

Rysunek 1 Statyczna grupa osób

Atrybuty | Warunki | Typy czynności | Grupy osób zatwierdzających

Konfiguracja: Grupy osób zatwierdzających >

Aktualizacja grupy osób zatwierdzających: SZYMANIAK

\* Wskazuje wymagane pole

►Reguły korzystające z grupy osób zatwierdzających

Szczegóły

Nazwa SZYMANIAK  
 \* Opis SZYMANIAK

\* Numer porządkowy 1

Metoda głosowania Seria

Typ użycia Statyczne

Zapytanie

Dynamiczna grupa osób zatwierdzających wymaga instrukcji SQL.

Zatwierdzenie

Członkowie grupy

Należy wprowadzić członków statycznej grupy osób zatwierdzających.

Typ osoby zatwierdzającej	Osoba zatwierdzająca	Numer porządkowy	Kasowanie
Kadry - osoby	Kadry - osoby: SZYMANIAK, MARIUSZ	1	
Kadry - osoby	Kadry - osoby: Orlińska, Barbara	2	

Dodawanie kolejnego wiersza

Rysunek 2 Dynamiczna grupa osób (użytkownicy przekazywani przez per\_all\_people\_f.person\_id)

Atrybuty | Warunki | Typy czynności | Grupy osób zatwierdzających

Konfiguracja: Grupy osób zatwierdzających >

Aktualizacja grupy osób zatwierdzających: SZYMANIAK

\* Wskazuje wymagane pole

►Reguły korzystające z grupy osób zatwierdzających

Szczegóły

Nazwa SZYMANIAK  
 \* Opis SZYMANIAK

\* Numer porządkowy 1

Metoda głosowania Seria

Typ użycia Dynamiczne

Zapytanie

```
select 'person_id'||employee_id
from fnd_user
where user_name like 'BORLINSK1%'
and employee_id is not null
```

Dynamiczna grupa osób zatwierdzających wymaga instrukcji SQL.

Zatwierdzenie

Rysunek 3 Dynamiczna grupa osób (użytkownicy przekazywani przez fnd\_user.user\_id)

Nazwa **SZYMANIAK**

\* Opis SZYMANIAK

\* Numer porządkowy 1

Metoda głosowania Seria

Typ użycia Dynamiczne

Zapytanie  

```
select 'user id:'||created_by from gmd_results where result_id = :transactionId
```

Dynamiczna grupa osób zatwierdzających wymaga instrukcji SQL.

Zatwierdzenie

## Reguły

- Łączą warunki z grupami osób oraz z action type.
- Jak utworzyć prostą regułę: Action type określa co reguła ma zrobić, np. utworzyć początkową listę zatwierdzeń, końcową listę zatwierdzeń, zasadniczą listę zatwierdzeń, coś innego. Jeżeli chcesz po prostu utworzyć listę zatwierdzeń wykonaj:
  - Wybierz typ czynności zgodnie z rysunkiem (nie zapomnisz o tym, bo bez tego nie można utworzyć reguły)

Konfiguracja: Typy czynności >

### Użycie istniejącego typu czynności: Wybór typów czynności

Należy wybrać co najmniej jeden typ czynności i kliknąć przycisk "Kontynuuj" w celu użycia go w ramach bieżącego typu transakcji.

☞ Wskazuje już dostępne w ramach typu transakcji Process Quality Lot Expiry Sample Creation.

#### Wyszukiwanie

Typ reguły Wszystkie Wykonaj

Tworzenie

Wybierz wszystko | Nic nie wybieraj

Wybierz	Nazwa <sup>△</sup>	Opis	Typ reguły
<input type="checkbox"/>	<a href="#">bez upoważnienia do ostatecznego zatwierdzenia</a>	rozszerzanie łańcucha upoważnień poza osobę zatwierdzającą	Modyfikacja listy
<input type="checkbox"/>	<a href="#">bezwzględny poziom typu stanowiska</a>	łańcuch upoważnień na podstawie bezwzględnego poziomu typu stanowiska	Łańcuch upoważnień
<input checked="" type="checkbox"/>	<a href="#">grupa zatwierdzeń w łańcuchu upoważnień</a>	łańcuch upoważnień zawiera grupę zatwierdzeń	Łańcuch upoważnień
<input type="checkbox"/>	<a href="#">kierownik, a następnie osoba ostatecznie zatwierdzająca</a>	łańcuch upoważnień zawiera kierownika osoby zlecającej, a następnie osobę ostatecznie zatwierdzającą	Łańcuch upoważnień
<input type="checkbox"/>	<a href="#">podwójny łańcuch upoważnień</a>	łańcuch upoważnień zawiera dwa łańcuchy podrzędne, każdy na podstawie poziomu typu stanowiska	Łańcuch upoważnień

- Wybierz typ reguły – tworzenie listy. Krok dodawanie warunków: Nie musisz dodawać żadnych warunków.



**Tworzenie nowej reguły: Wprowadzanie szczegółów reguły**

\* Wskazuje wymagane pole

\* Nazwa: Lista powiadamianych

\* Typ reguły: Tworzenie listy

Klasa pozycji: Nagłówek

\* Data początkowa: 28-lut-2008 (przykład: 28-lut-2008)

\* Data końcowa: 31-gru-4712

- o Wybierz typ czynności i grupę zatwierdzeń zgodnie z rysunkiem



**Tworzenie nowej reguły: Dodawanie czynności**

Anuluj

Dodawanie czynności

Typ czynności: grupa zatwierdzeń w łańcuchu upoważnień

\*Czynność: Wymagane zatwierdzenie przez osobę z grupy Uzytkowr

## API

Elementy składowe API:

### Zmienna pomocnicza

Approver `ame_util.approverRecord;`

### Zainicjowanie API

Ścieżka zatwierdzeń jest identyfikowana za pomocą trójki elementów:

Nazwa zmiennej	Opis	Przykład
applicationIdIn	Aplikacja	<code>select application_id from fnd_application where application_short_name='GMD'</code>
transactionTypeIn	Typ transakcji (unikalny identyfikator tabeli, z której prześlemy ID)	<code>'GMDQMPF'</code>
transactionIdIn	Identyfikator transakcji	<code>select result_id from gmd_results</code>

Te trzy elementy są przekazywane podczas każdego odwołania do AME.

```
ame_api.clearAllApprovals(
    applicationIdIn => l_application_id,
    transactionIdIn => l_result_id,
    transactionTypeIn => l_transaction_type);
```

### Pobranie pierwszego / kolejnego zatwierdzającego

```
ame_api.getNextApprover(
```

```

applicationIdIn => l_application_id,
transactionIdIn => l_result_id,
transactionTypeIn => l_transaction_type,
nextApproverOut => Approver);

if(Approver.user_id is null and Approver.person_id is null) then
  /* No Approval Required */
  ...
end if;

if(Approver.person_id is null) then
  select user_name into l_user from fnd_user
  where user_id=Approver.user_id;
else
  select user_name into l_user from fnd_user
  where user_id=ame_util.PERSONIDTOUSERID(Approver.person_id);
end if;

... l_user zawiera nazwę zatwierdzającego ...

-- zaznaczamy, że zatwierdzający został pobrany
Approver.approval_status := ame_util.approvedStatus;
ame_api.updateApprovalStatus(
  applicationIdIn => l_application_id,
  transactionIdIn => l_result_id,
  approverIn => Approver,
  transactionTypeIn => l_transaction_type,
  forwarderIn => ame_util.emptyApproverRecord);

```

## Przykłady

przykład użycia : pakiet GMD\_QMTES

## Obiekty w bazie danych

### Widoki

```

select * from AME_TRANSACTION_TYPES_V where transaction_type_id like 'GMD%'

select * from AME_ATTRIBUTES_VL

select * from AME_RULES--_VL

select * from AME_ACTIONS_VL

select * from AME_ACTION_TYPES_VL

select * from AME_APPROVAL_GROUPS_VL

select * from AME_CONFIG_VARS_VL

select * from AME_ITEM_CLASSES_VL

select * from AME_CALLING_APPS_VL

```

### Inne obiekty

```

select object_name
from all_objects
where upper(object_name) like upper('AME%')
  and upper(owner) like upper('%%')
-- and upper(object_type) in ('VIEW')

```

```
and status in ('INVALID','VALID')
order by object_type
```

## W razie problemów

Przypisanie autoryzacji AME do usera w 11i10 (wersja AME.B), opisuje Note: 466876.1 (AME Responsibilities Is Not Working Any More After Upgrade)

## Zagadnienia otwarte

1. W kodzie występuje tylko jedna wartość do identyfikowania obiektu - transactionid (jest to np. id faktury). W jaki sposób identyfikuje się poziom linii podczas generowania ścieżki dla faktury.
2. Warunki są krostypowe ?