

Workflow – knowledge

Skąd pobrać	2
Jak uruchomić / zatrzymać workflow z poziomu kodu	2
Przekazywanie parametrów pomiędzy klockami workflow.....	3
Jak odczytać / zapisać wartość atrybutu globalnego (ItemType)	4
Jak odczytać / zapisać wartość atrybutu lokalnego (funkcji lub procedury)	4
Jak zbudować procedurę(funkcję) dla workflow	4
Jak zbudować powiadomienie (notification)	8
Proste powiadomienie	9
Zaawansowane powiadomienie.....	9
Powiadomienie CLOB	11
Jak załączyć do powiadomienia plik (załącznik)	12
Jak załączyć do powiadomienia łącze do formularza Aplikacji.....	14
Odczytanie komunikatu wprowadzonego na powiadomieniu i zapisanie go w bazie danych	15
Reguły czasu wolnego.....	16
Inne wbudowane właściwości powiadomienia	17
Skąd wziąć standardowe klocki: start/stop/join itd.	17
Jak dodać klocki z innych item type	17
Jak zakończyć workflow błędem	17
Diagnostyka workflow	17
Jeżeli nie możesz za pomocą GUI odnaleźć procesu	18
Jak nadać uprawnienie administratora procesów workflow	18
Jak wyświetlić wszystkie atrybuty workflow	18
Proces zatwierdzania faktur	19
Informacje podstawowe	19
Wskazówki techniczne.....	23
Zabezpieczenie przed wielokrotnym uruchomieniem workflow dla dokumentu gdy dokument jest w trakcie przetwarzania workflow; ponowne uruchomienie odrzuconego workflow	26
FAQ.....	27
jak się tworzy rolę zawierającą dwóch użytkowników ?	29
Notification mailer	30
Jak skonfigurować workflow tak aby wysyłał maile	30
Diagnostyka notification mailer	30
Gdzie się ustawia język dla powiadomień.....	32
Jak przełączyć sposób obierania maili przez użytkowników na mail	33
Tabele workflow	34
Interakcja z użytkownikami	35
Popychanie workflow	35
Jak z poziomu kodu popchnąć proces workflow (zatrzymany z powodu błędu) lub zasymulować odpowiedź na powiadomienie	36
ERROR PAGE.....	36
Dziennik zdarzeń / log.....	37
Co to są zdarzenia / events.....	37
Uruchomienie zdarzenia.....	37
Uruchomienie zdarzenia z plikiem XML / czytanie XML.....	38
Inne informacje.....	39
Zastępstwa / reguły czasu wolnego	40
Procesy podrzędne	41
Zagadnienia otwarte	45

Skąd pobrać

<http://www.oracle.com/technology/software/products/workflow/htdocs/winclient.html>

Jak uruchomić / zatrzymać workflow z poziomu kodu

Sposób 1/ wywołanie procesu workflow bezpośrednio

```
declare
  ls_itemkey number;
begin
  select po_wf_debug_s.nextval into ls_itemkey from dual;
  wf_engine.createprocess('XXTEST', ls_itemkey, 'XXMAIN', 'user key ' || ls_itemkey, 'SYSADMIN');
  -- init attributes
  -- wf_engine.setItemAttrNumber ('XXTEST', ls_itemkey, 'APINV_AII', p_invoice_id);

  --opcjonalnie - ustawienie informacji o nadrzędnym procesie workflow
  wf_engine.SETITEMPARENT(itemtype => l_itemtype, itemkey => l_itemkey, parent_itemtype =>
    p_itemtype, parent_itemkey => p_itemkey, parent_context => NULL); --zob. gmd_qmtes_body

  wf_engine.startprocess('XXTEST', ls_itemkey);
  commit;
end;
```

File: smallDemo.wft

Ustawienie procesu nadrzędnego skutkuje tym, że z poziomu GUI można dokonać przejścia od procesu nadrzędnego do procesów podrzędnych:

Wybór procesu workflow i przegląd... (Historia operacji) Diagram statusu Odpowiedzi uczestników Workflow - szczegóły										Poprzednie 1-25 Dalej 26	
Wybierz	Status	Workflow - typ	Klucz pozycji	Nazwa procesu	Klucz użytkown.	Właściciel	Rozpoczęte ▾	Wykonane	Podrzedne procesy workflow		
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51655-634680-27/02/08 04:18:46	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:46	27-lut-2008 16:18:46			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51655-634681-27/02/08 04:18:46	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:46	27-lut-2008 16:18:46			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51655-634679-27/02/08 04:18:45	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:45	27-lut-2008 16:18:45			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51655-634678-27/02/08 04:18:45	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:45	27-lut-2008 16:18:45			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51654-634676-27/02/08 04:18:21	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:21	27-lut-2008 16:18:21			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51654-634675-27/02/08 04:18:21	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:21	27-lut-2008 16:18:21			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51654-634677-27/02/08 04:18:21	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:21	27-lut-2008 16:18:21			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	51654-634674-27/02/08 04:18:20	OPM - wykonywanie podprocesów testowania jakości			27-lut-2008 16:18:20	27-lut-2008 16:18:21			
☐	✓ Zakończony	OPM - wykonywanie testów jakości	8781	OPM - wykonywanie procesów testowania jakości	przetwarzanie próbki	Orlińska, Barbara	27-lut-2008 15:59:10	27-lut-2008 16:18:46			
☐	✓ Zakończony	OPM - wykonywanie	8780	OPM - wykonywanie procesów	przetwarzanie	Orlińska,	27-lut-2008			Służy do wyświetlania podrzędnych procesów	

Child Workflows: GMDQMTST, 8781, przetwarzanie próbek

Poniższe procesy workflow zostały wyznaczone jako podrzędne względem procesu "przetwarzanie próbki". Procesy te można przeglądać i zarządzać nimi, wybierając odpowiednie pozycje

WSKAZÓWKA Dane historyczne dotyczące operacji workflow są okresowo usuwane z systemu, dlatego ich przejrzanie może nie być już możliwe.

Wybór procesu workflow: (Historia operacji) Diagram statusu Odpowiedzi uczestników Workflow - szczegóły						
Wybierz	Status	Klucz pozycji	Klucz użytkown.	Workflow - typ	Właściciel	Rozpoczęte ▾
☐	✓ Zakończony	51655-634680-27/02/08 04:18:46		OPM - wykonywanie testów jakości		27-lut-2008 16:18:46
☐	✓ Zakończony	51655-634681-27/02/08 04:18:46		OPM - wykonywanie testów jakości		27-lut-2008 16:18:46
☐	✓ Zakończony	51655-634678-27/02/08 04:18:45		OPM - wykonywanie testów jakości		27-lut-2008 16:18:45
☐	✓ Zakończony	51655-634679-27/02/08 04:18:45		OPM - wykonywanie testów jakości		27-lut-2008 16:18:45
☐	✓ Aktywne	WF158823		System: Błąd		27-lut-2008 16:01:09
☐	✓ Aktywne	WF158822		System: Błąd		27-lut-2008 16:00:22

Sposób 2/ wywołanie zdarzenia

```
declare
  ls_itemkey number;
begin
  select po_wf_debug_s.nextval into ls_itemkey from dual;
  wf_event.raise('oracle.apps.gmd.qm.performtest', ls_itemkey, 'ala ma kota');
  commit;
end;
```

Zatrzymanie procesu workflow:

```
WF_ENGINE.AbortProcess(p_item_type, p_item_key);
```

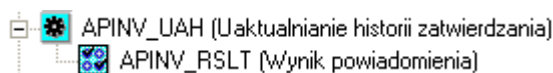
+ opcjonalnie

```
WF_PURGE.Total(pitem_type, pitem_key, sysdate, true, false, 1);
```

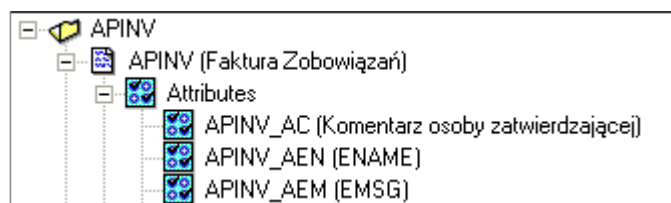
Przekazywanie parametrów pomiędzy klockami workflow

Idea funkcjonowania procesu workflow polega na tym, że istnieje zestaw atrybutów globalnych (współdzielonych), za pomocą których poszczególne procedury mogą przekazywać dane między sobą.

1. Każdy pojedynczy klocek workflow (funkcja, powiadomienie) posiada atrybuty lokalne- osiągalne tylko z poziomu danego klocka.



2. Proces jako całość posiada atrybuty globalne – osiągalne z każdego poziomu.



3. Wartości domyślne atrybutów lokalnych mogą być inicjowane przez atrybuty globalne (a atrybuty globalne mogą zostać ustawione w odpowiedni sposób przez poprzednie klocki).

Właściwości: Navigator Control

Function | Details | Roles | Access | Node | Node Attributes

Attribute

Name: XX Status

Type: Item Attribute

Value: []

Edit

Name	Value Type	Value	Type	Description
XX Status	Item Attribute		Lookup	XX Status

OK Anuluj Zastosuj Pomoc

Ten mechanizm jest wykorzystywany do przekazywania parametrów pomiędzy elementami workflow.

Jak odczytać / zapisać wartość atrybutu globalnego (ItemType)

```
h_profile_option_name := Wf_Engine.GetItemAttrText(itemtype, itemkey, 'ACCOUNT_TYPE');
h_profile_option_name := Wf_Engine.GetItemAttrNumber(itemtype, itemkey, 'ACCOUNT_TYPE');
h_profile_option_name := Wf_Engine.GetItemAttrDate(itemtype, itemkey, 'ACCOUNT_TYPE');

wf_engine.SetItemAttrText(itemtype, itemkey, 'ACCOUNT_TYPE', l_result_value);
wf_engine.SetItemAttrNumber(itemtype, itemkey, 'ACCOUNT_TYPE', l_result_value);
wf_engine.SetItemAttrDate(itemtype, itemkey, 'ACCOUNT_TYPE', l_result_value);
```

Jak odczytać / zapisać wartość atrybutu lokalnego (funkcji lub procedury)

```
l_profile_option_name := wf_engine.GetActivityAttrText(itemtype, itemkey, actid,
'PROFILE_OPTION_NAME');

wf_engine.SetActivityAttrText(itemtype, itemkey, actid, 'PROFILE_OPTION_NAME',
l_result_value);
```

Jak zbudować procedurę(funkcję) dla workflow

1. Powiązana procedura/ funkcja PLSQL musi mieć ZAWSZE następujący nagłówek

```
procedure GET_RET_DFF
( itemtype in varchar2
, itemkey in varchar2
, actid in number
, funcmode in varchar2
, result out varchar2)
```

2. Parametry odczytuje się za pomocą funkcji `wf_engine.GetActivityAttrText`, `Wf_Engine.GetItemAttrText`, a zapisuje za pomocą ich odpowiedników (`get -> set`).
3. Procedura musi zwrócić wartość (zmienna result) w formacie „COMPLETE:”

Słowo COMPLETE (używane najczęściej) pełna lista wygląda następująco:

(źródło: WF_ENGINE)

-- Item activity statuses

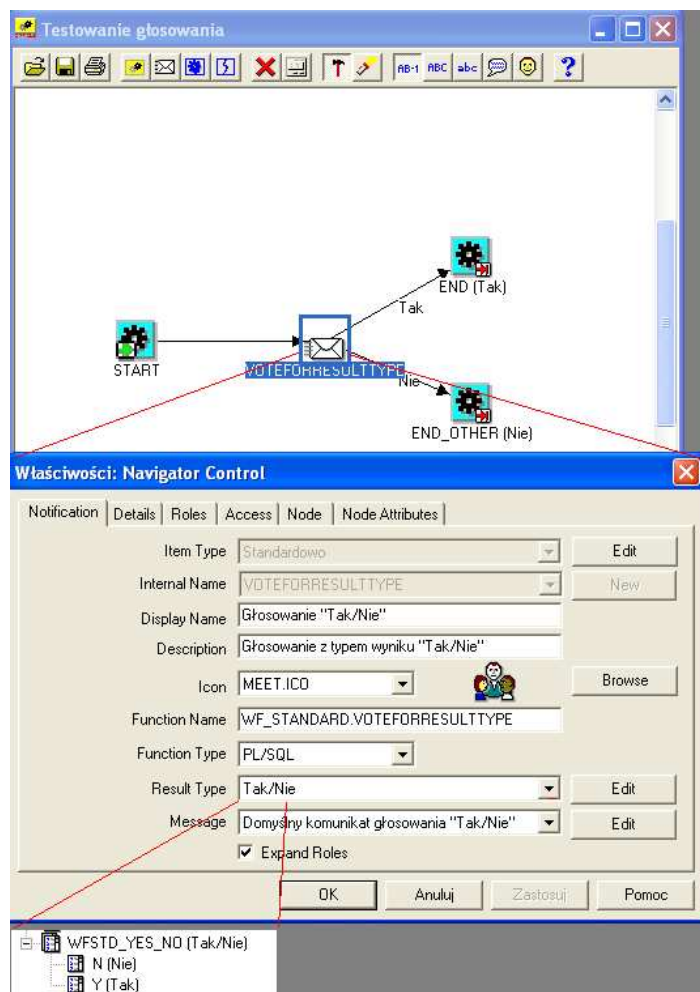
eng_completed	varchar2(8) := 'COMPLETE';	-- Normal completion
eng_active	varchar2(8) := 'ACTIVE';	-- Activity running
eng_waiting	varchar2(8) := 'WAITING';	-- Activity waiting to run
eng_notified	varchar2(8) := 'NOTIFIED';	-- Notification open
eng_suspended	varchar2(8) := 'SUSPEND';	-- Activity suspended
eng_deferred	varchar2(8) := 'DEFERRED';	-- Activity deferred
eng_error	varchar2(8) := 'ERROR';	-- Completed with error

„DEFERRED:” oznacza, że proces jest wstrzymany i będzie kontynuowany dopiero po wybraniu polecenia ponów w interfejsie graficznym użytkownika lub po uruchomieniu programu współbieżnego „Proces workflow w tle” z parametrem procesy odroczone=tak.

3. Funkcja musi zwrócić wartość (zmienna result) w formacie :

COMPLETE:<wartość z lookupa> (lub DEFFER)

Lookup musi być podłączony do definicji funkcji zgodnie z rysunkiem.



create or replace package body Xx_Faflex_Pkg_Wf as

procedure GET_RET_DFF

(itemtype in	varchar2
, itemkey in	varchar2
, actid in	number
, funcmode in	varchar2

```

, result out varchar2)
is
cursor c_wf (b_curr_dist_id number)
is
select fl.ATTRIBUTE1 LOOKUP_ATTR1
, fl.ATTRIBUTE2 LOOKUP_ATTR2
, fl.ATTRIBUTE3 LOOKUP_ATTR3
, fl.ATTRIBUTE4 LOOKUP_ATTR4
, fl.ATTRIBUTE5 LOOKUP_ATTR5
, fl.ATTRIBUTE6 LOOKUP_ATTR6
, fl.ATTRIBUTE7 LOOKUP_ATTR7
, fl.ATTRIBUTE8 LOOKUP_ATTR8
, fl.ATTRIBUTE9 LOOKUP_ATTR9
, fl.ATTRIBUTE10 LOOKUP_ATTR10
, fl.ATTRIBUTE11 LOOKUP_ATTR11
, fl.ATTRIBUTE12 LOOKUP_ATTR12
, fl.ATTRIBUTE13 LOOKUP_ATTR13
, fl.ATTRIBUTE14 LOOKUP_ATTR14
, fl.ATTRIBUTE15 LOOKUP_ATTR15
, fr.ATTRIBUTE1 RET_ATTR1
, fr.ATTRIBUTE2 RET_ATTR2
, fr.ATTRIBUTE3 RET_ATTR3
, fr.ATTRIBUTE4 RET_ATTR4
, fr.ATTRIBUTE5 RET_ATTR5
, fr.ATTRIBUTE6 RET_ATTR6
, fr.ATTRIBUTE7 RET_ATTR7
, fr.ATTRIBUTE8 RET_ATTR8
, fr.ATTRIBUTE9 RET_ATTR9
, fr.ATTRIBUTE10 RET_ATTR10
, fr.ATTRIBUTE11 RET_ATTR11
, fr.ATTRIBUTE12 RET_ATTR12
, fr.ATTRIBUTE13 RET_ATTR13
, fr.ATTRIBUTE14 RET_ATTR14
, fr.ATTRIBUTE15 RET_ATTR15
from FA_RETIREMENTS fr
, FA_DISTRIBUTION_HISTORY fdh
, FA_LOOKUPS fl
, ( select ASSET_ID
-- , BOOK_TYPE_CODE
, MAX(TRANSACTION_HEADER_ID_IN) MAXTRANS
from FA_RETIREMENTS
group by ASSET_ID, BOOK_TYPE_CODE
) B
where fdh.DISTRIBUTION_ID = b_curr_dist_id
and fr.ASSET_ID = fdh.ASSET_ID
-- AND fdh.BOOK_TYPE_CODE = fr.BOOK_TYPE_CODE
and fdh.DATE_INEFFECTIVE is null
and fl.LOOKUP_CODE = fr.RETIREMENT_TYPE_CODE
and fl.LOOKUP_TYPE = 'RETIREMENT'
and fdh.TRANSACTION_HEADER_ID_OUT is null
and fr.TRANSACTION_HEADER_ID_IN = B.MAXTRANS
and fr.RETIREMENT_ID not in
( select fdh.RETIREMENT_ID
from FA_DISTRIBUTION_HISTORY fdh
where fdh.RETIREMENT_ID = fr.RETIREMENT_ID
)
union
-- Not Partial Cost (Full and Partial Unit Cost Retirement Types)
select fl.ATTRIBUTE1 LOOKUP_ATTR1
, fl.ATTRIBUTE2 LOOKUP_ATTR2
, fl.ATTRIBUTE3 LOOKUP_ATTR3
, fl.ATTRIBUTE4 LOOKUP_ATTR4
, fl.ATTRIBUTE5 LOOKUP_ATTR5
, fl.ATTRIBUTE6 LOOKUP_ATTR6
, fl.ATTRIBUTE7 LOOKUP_ATTR7
, fl.ATTRIBUTE8 LOOKUP_ATTR8
, fl.ATTRIBUTE9 LOOKUP_ATTR9
, fl.ATTRIBUTE10 LOOKUP_ATTR10
, fl.ATTRIBUTE11 LOOKUP_ATTR11
, fl.ATTRIBUTE12 LOOKUP_ATTR12
, fl.ATTRIBUTE13 LOOKUP_ATTR13
, fl.ATTRIBUTE14 LOOKUP_ATTR14
, fl.ATTRIBUTE15 LOOKUP_ATTR15
, fr.ATTRIBUTE1 RET_ATTR1
, fr.ATTRIBUTE2 RET_ATTR2
, fr.ATTRIBUTE3 RET_ATTR3
, fr.ATTRIBUTE4 RET_ATTR4

```

```

, fr.ATTRIBUTE5 RET_ATTR5
, fr.ATTRIBUTE6 RET_ATTR6
, fr.ATTRIBUTE7 RET_ATTR7
, fr.ATTRIBUTE8 RET_ATTR8
, fr.ATTRIBUTE9 RET_ATTR9
, fr.ATTRIBUTE10 RET_ATTR10
, fr.ATTRIBUTE11 RET_ATTR11
, fr.ATTRIBUTE12 RET_ATTR12
, fr.ATTRIBUTE13 RET_ATTR13
, fr.ATTRIBUTE14 RET_ATTR14
, fr.ATTRIBUTE15 RET_ATTR15
from FA_RETIREMENTS fr
, FA_TRANSACTION_HEADERS fth
, FA_LOOKUPS fl
, FA_DISTRIBUTION_HISTORY fdh
where fdh.DISTRIBUTION_ID = b_curr_dist_id
and fth.TRANSACTION_HEADER_ID = fr.TRANSACTION_HEADER_ID_IN
and fdh.ASSET_ID = fr.ASSET_ID
and fdh.RETIREMENT_ID = fr.RETIREMENT_ID
and fl.LOOKUP_CODE = fr.RETIREMENT_TYPE_CODE
and fl.LOOKUP_TYPE = 'RETIREMENT';
--
r_wf c_wf%rowtype;
--
h_account_type          varchar2(30);
h_distribution_id        number;
l_SegName               varchar2(30);
l_result_attr           varchar2(240);
l_result_value          varchar2(240);

begin <<GET_RET_DFF>>
DBMS_OUTPUT.PUT_LINE('Get_RET_DFF');
--
if (funcmode = 'RUN')
then
h_account_type := wf_Engine.GetItemAttrText(itemtype,itemkey,'ACCOUNT_TYPE');
DBMS_OUTPUT.PUT_LINE(h_account_type);
h_distribution_id := wf_Engine.GetItemAttrNumber(itemtype,itemkey,'DISTRIBUTION_ID');
DBMS_OUTPUT.PUT_LINE(h_distribution_id);
l_result_attr := wf_engine.GetActivityAttrText(itemtype, itemkey, actid,
'RETURN_VALUE_ATTRIBUTE');
if h_distribution_id is not null
then
begin
open c_wf(h_distribution_id);
fetch c_wf into r_wf;
if c_wf%FOUND
then
close c_wf;
l_SegName := wf_engine.GetActivityAttrText(itemtype, itemkey, actid, 'ATTRIBUTE_SYMBOL');
if l_SegName = 'RET_ATTR1' then l_result_value := r_wf.ret_attr1; end if;
if l_SegName = 'RET_ATTR2' then l_result_value := r_wf.ret_attr2; end if;
if l_SegName = 'RET_ATTR3' then l_result_value := r_wf.ret_attr3; end if;
if l_SegName = 'RET_ATTR4' then l_result_value := r_wf.ret_attr4; end if;
if l_SegName = 'RET_ATTR5' then l_result_value := r_wf.ret_attr5; end if;
if l_SegName = 'RET_ATTR6' then l_result_value := r_wf.ret_attr6; end if;
if l_SegName = 'RET_ATTR7' then l_result_value := r_wf.ret_attr7; end if;
if l_SegName = 'RET_ATTR8' then l_result_value := r_wf.ret_attr8; end if;
if l_SegName = 'RET_ATTR9' then l_result_value := r_wf.ret_attr9; end if;
if l_SegName = 'RET_ATTR10' then l_result_value := r_wf.ret_attr10; end if;
if l_SegName = 'RET_ATTR11' then l_result_value := r_wf.ret_attr11; end if;
if l_SegName = 'RET_ATTR12' then l_result_value := r_wf.ret_attr12; end if;
if l_SegName = 'RET_ATTR13' then l_result_value := r_wf.ret_attr13; end if;
if l_SegName = 'RET_ATTR14' then l_result_value := r_wf.ret_attr14; end if;
if l_SegName = 'RET_ATTR15' then l_result_value := r_wf.ret_attr15; end if;

if l_SegName = 'LOOKUP_ATTR1' then l_result_value := r_wf.lookup_attr1; end if;
if l_SegName = 'LOOKUP_ATTR2' then l_result_value := r_wf.lookup_attr2; end if;
if l_SegName = 'LOOKUP_ATTR3' then l_result_value := r_wf.lookup_attr3; end if;
if l_SegName = 'LOOKUP_ATTR4' then l_result_value := r_wf.lookup_attr4; end if;
if l_SegName = 'LOOKUP_ATTR5' then l_result_value := r_wf.lookup_attr5; end if;
if l_SegName = 'LOOKUP_ATTR6' then l_result_value := r_wf.lookup_attr6; end if;
if l_SegName = 'LOOKUP_ATTR7' then l_result_value := r_wf.lookup_attr7; end if;
if l_SegName = 'LOOKUP_ATTR8' then l_result_value := r_wf.lookup_attr8; end if;
if l_SegName = 'LOOKUP_ATTR9' then l_result_value := r_wf.lookup_attr9; end if;
if l_SegName = 'LOOKUP_ATTR10' then l_result_value := r_wf.lookup_attr10; end if;
if l_SegName = 'LOOKUP_ATTR11' then l_result_value := r_wf.lookup_attr11; end if;

```

```

        if l_SegName = 'LOOKUP_ATTR12' then l_result_value := r_wf.lookup_attr12; end if;
        if l_SegName = 'LOOKUP_ATTR13' then l_result_value := r_wf.lookup_attr13; end if;
        if l_SegName = 'LOOKUP_ATTR14' then l_result_value := r_wf.lookup_attr14; end if;
        if l_SegName = 'LOOKUP_ATTR15' then l_result_value := r_wf.lookup_attr15; end if;

        wf_engine.SetItemAttrText(itemtype, itemkey, l_result_attr, l_result_value);

        result := 'COMPLETE: ';
    else
        close c_wf;
        --
        DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND');
        result := 'COMPLETE: ';
    end if;
    return;
exception
    when OTHERS then
        DBMS_OUTPUT.PUT_LINE(sqlerrm);
        result := 'COMPLETE: ';
        return;
    end;
    return;
else
    DBMS_OUTPUT.PUT_LINE('h_distribution_id IS NULL');
    result := 'COMPLETE: ';
    return;
end if;
elsif (funcmode = 'CANCEL') then
    result := 'COMPLETE: ';
    return;
else
    result := '';
    return;
end if;

exception
    when OTHERS then
        wf_Core.context('XX_FAFLEX_PKG_WF', 'GET_RET_DFF',
            itemtype, itemkey, TO_CHAR(actid), funcmode);
        raise;
end GET_RET_DFF;

end Xx_Faflex_Pkg_Wf;
/

--

procedure Noop(itemtype in varchar2,
               itemkey   in varchar2,
               actid     in number,
               funcmode  in varchar2,
               resultout in out nocopy varchar2)
is
begin
    resultout := wf_engine.eng_completed || ':' || wf_engine.eng_null;
exception
    when others then
        wf_Core.context('Wf_Standard', 'Noop', itemtype,
            itemkey, to_char(actid), funcmode);
        raise;
end Noop;

```

Jak zbudować powiadomienie (notification)

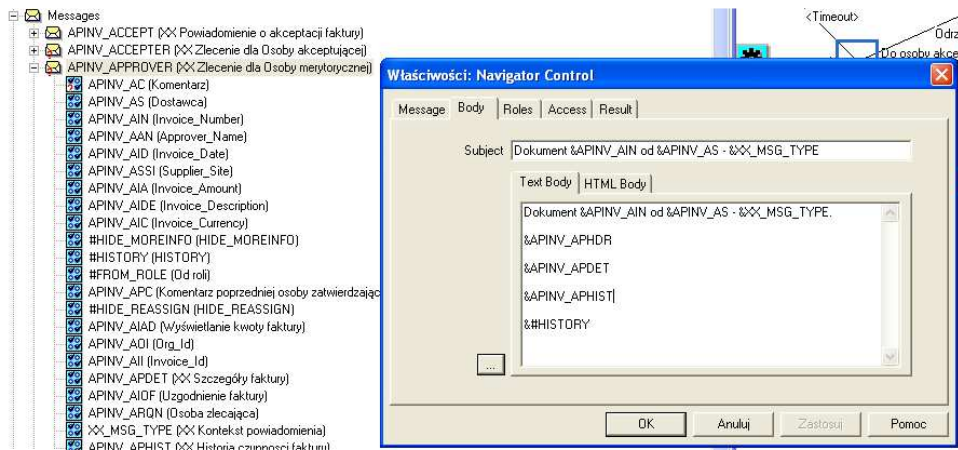
Powiadomienie zapewnia interakcję z użytkownikiem:

- Użytkownik może dostawać powiadomienie w oknie Aplikacji lub za pomocą maila.
- Powiadomienie może mieć postać zwykłego tekstu lub html

Proste powiadomienie

składa się z:

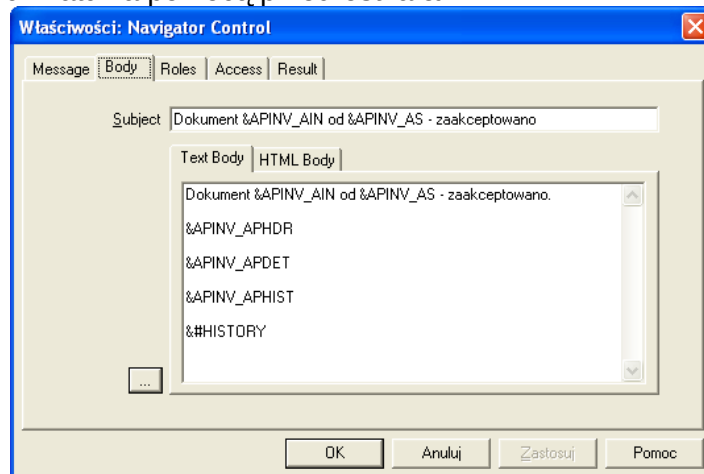
1. Komunikatu w postaci: temat + treść. Komunikat definiuje się jako osobny obiekt (message) i podłącza do powiadomienia we *właściwości message*. Wygląd powiadomienia definiuje się za pomocą powiązanego z powiadomieniem komunikatu (message) – sekcja body:



- Wskazanie osoby dla której powiadomienie jest przeznaczone – *pole Performer*. W tym polu trzeba wpisać nazwę roli (z reguły jest to nazwa użytkownika)

Zaawansowane powiadomienie

1. Dynamiczna zmiana odbiorcy powiadomienia – pobieraj nazwę z atrybutu globalnego o typie role.
2. Powiadomienie ze zmiennymi komunikatami – odwołuj się do atrybutów komunikatu za pomocą przedrostka &.

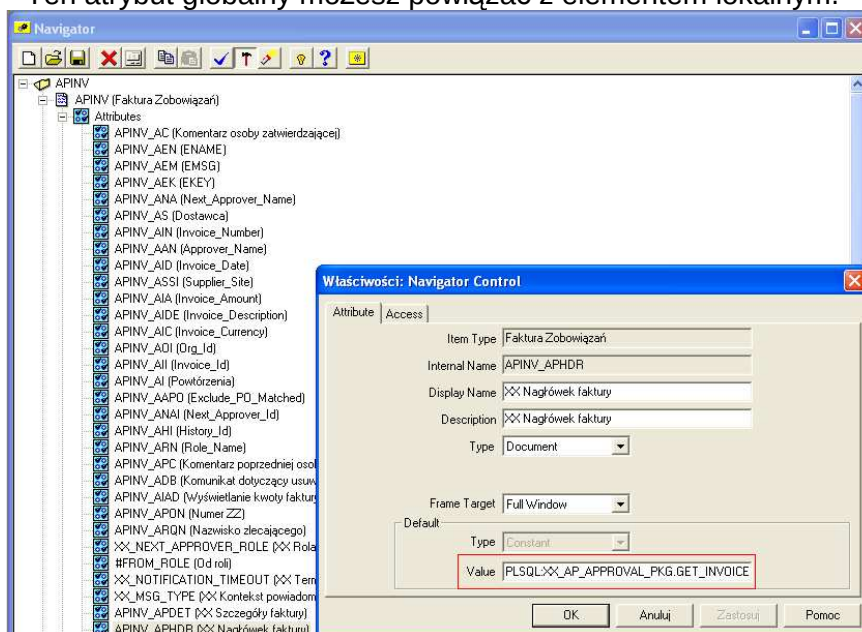


3. Wiadomość sformatowana - Atrybut globalny o typie *Document* może być powiązany z funkcją PLSQL, która może wygenerować opis dla elementu w formacie HTML, jak przedstawiono poniżej. Funkcja PLSQL, podobnie jak poprzednio, musi mieć ściśle określoną strukturę nagłówka.

VALUE=

PLSQL:XX_AP_APPROVAL_PKG.GET_INVOICE_HEADER/<document id>

Przykładowy kod źródłowy funkcji załączono w oddzielnym pliku. <dokument id> może być ustawiony dynamicznie – zob. procedura `init_attributes`. Ten atrybut globalny możesz powiązać z elementem lokalnym.



4. Powiadomienia mailowe – dostępne po skonfigurowaniu usługi "Workflow Notification Mailer"
5. Interakcja z użytkownikiem. Są dwie możliwości, z których można korzystać jednocześnie:
 - 1. Przyciski decyzyjne. Zgodnie z definicją w polu *Result Type*.

Preferencje Pomoc Zamykanie okna

[Powiadomienia](#) >

Dokument 20070517K od ORACLE POLSKA SP.Z O.O. - wymagana akceptacja

Odrzucenie
Zatwierdzenie

Do **Szymczak, Maciej**

Wysłane **17-05-2007 15:53:10**

ID **83127**

Dokument 20070517K od ORACLE POLSKA SP.Z O.O. - wymagana akceptacja.

- 2. Listy wartości i pola tekstowe dostępne na dole powiadomienia. W sekcji *body* możesz odwoływać się tylko do elementu o źródle (*source*) Send. Elementy o źródle Respond, oznaczone znakiem zapytania (powyżej) są elementami interakcji z użytkownikiem i wyświetlają się automatycznie w kolejności określonej w drzewie.

Odpowiedz

Komentarz

Kod DAO

Następna osoba akceptująca

Przyczyna odrzucenia

▼

Wszyscy pracownicy i użytkownicy

Błędna data

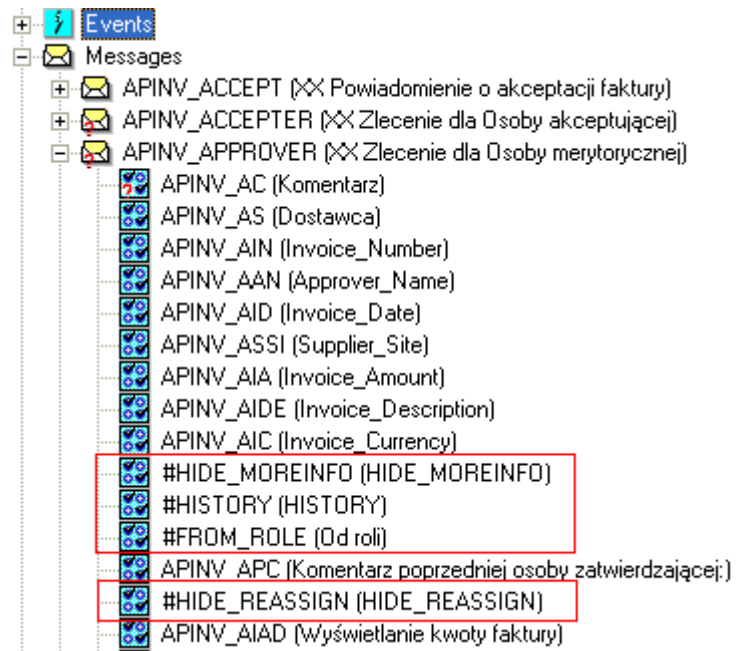
▼

🔍

Uwaga: Listy wartości nie są dostępne w przypadku powiadomień mailowych.

6. Akcje po przekroczeniu czasu oczekiwania (timeouts) – zob. sekcja `timeout`. Standardowo `timeout` nie uwzględnia dni wolnych – w załączonym przykładzie pokazano jak obejść tę niedogodność.

7. Możesz sterować tym, od kogo powiadomienie jest wysyłane, historią powiadomienia itd. Użyj w tym celu systemowych atrybutów obiektu message, np. #FROM_ROLE



8. Dodatkowe działania związane z powiadomieniem, np.
- odczytanie komunikatu wprowadzonego na powiadomieniu i zapisanie go w bazie danych
 - Sprawdzenie wymagalności pola w przypadku określonej akcji (np. odrzucenia) i wyświetlenie komunikatu błędnie.
- Z powiadomieniem można powiązać funkcję PL/SQL. Nagłówek powiązanej procedury plsql musi mieć strukturę jak w przypadku zwykłej funkcji workflow (opisanej w poprzednim punkcie) – przykład: XX_AP_APPROVAL_PKG.NOTIFICATION_ACCEPTER.

Powiadomienie CLOB

Zob. pakiet XXEX_004_PKG, projekt PZU

```
Wf_Engine.SetItemAttrText(itemtype,itemkey,'XXNOTIF_BODY',
'PLSQLCLOB:XXEX_004_PKG.GET_NOTIF_BODY/'||itemkey);
```

```
procedure get_notif_body
```

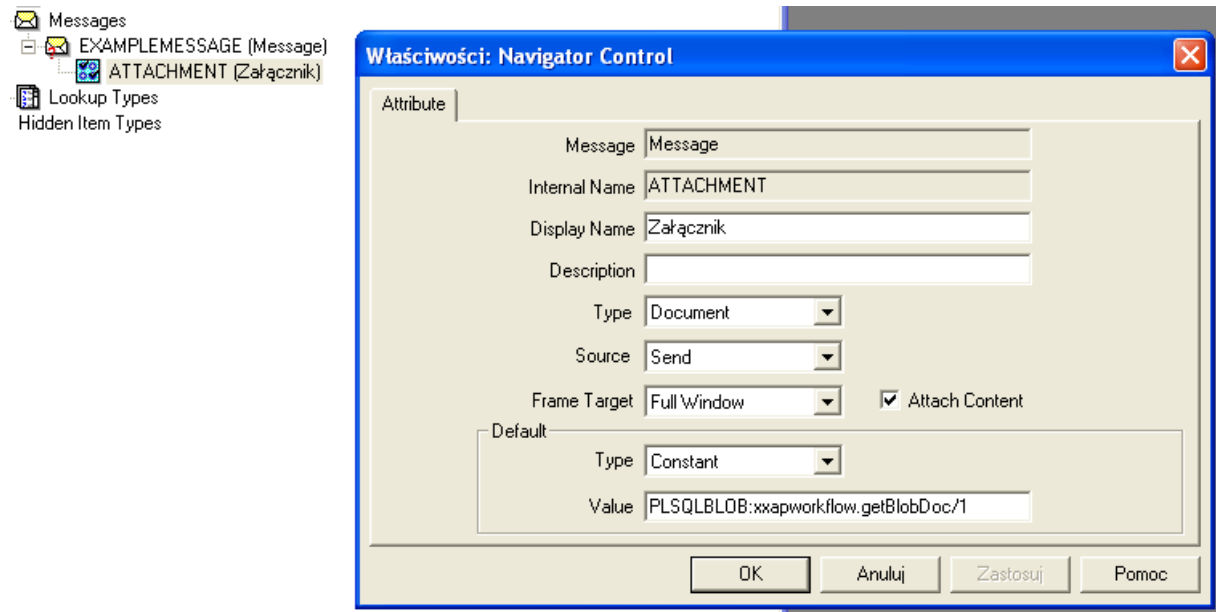
```
procedure writeappend (p_text varchar2) is
v_text varchar2(32000);
begin
v_text := replace(p_text,'<TD></TD>',' <TD>&nbsp;</TD>');
dbms_lob.writeappend(document, length(v_text), v_text);
end;
```

```
begin
dbms_lob.createtemporary(document, true, dbms_lob.session);
dbms_lob.open(document, dbms_lob.lob_readwrite);
```

Jak załączyć do powiadomienia plik (załącznik)

Można użyć CLOB lub BLOB.

Przykład na BLOB:



```
drop table xxext.xxapworkflow_files;
drop synonym xxapworkflow_files;

create table xxext.xxapworkflow_files (
  request_id number not null
, outfile      blob
, creation_date date default sysdate not null);

create index xxext.xxapworkflow_files_i1 on xxext.xxapworkflow_files (request_id);
create index xxext.xxapworkflow_files_i2 on xxext.xxapworkflow_files (creation_date);

create synonym xxapworkflow_files for xxext.xxapworkflow_files;

-- stores output file of concurrent program in xxapworkflow_files database table
procedure createBlobByRequestId ( p_request_id number) is
  l_outfile_name varchar2(500);
  l_path          varchar2(500);
  l_file_name     varchar2(500);
  src_lob         bfile;
  dest_lob        blob;
begin
  select outfile_name into l_outfile_name from fnd_concurrent_requests where request_id =
p_request_id;
  select xxmsz_tools.extractPath(l_outfile_name)
, xxmsz_tools.extractFileName(l_outfile_name)
  into l_path, l_file_name
  from dual;

  execute immediate 'create or replace directory OUTFILE_TOP as
''/d1/ITD2/appitdd/itd2comn/admin/out/ITD2_oratest5/''';
  src_lob := BFILENAME('OUTFILE_TOP', l_file_name);

  delete from xxapworkflow_files where request_id = p_request_id;
  delete from xxapworkflow_files where creation_date < sysdate - 7; --purge old data

  insert into xxapworkflow_files (request_id, outfile) values(p_request_id, empty_blob())
returning outfile into dest_lob;

  dbms_lob.open(src_lob, dbms_lob.lob_readonly);
```

```

dbms_lob.loadfromfile( dest_lob => dest_lob,
                      src_lob   => src_lob,
                      amount    => dbms_lob.getlength(src_lob) );
dbms_lob.close(src_lob);
commit;
end;

-- PLSQLBLOB:xxapworkflow.getBlobDoc/1
procedure getBlobDoc
(document_id in varchar2,
content_type in varchar2,
document in out nocopy blob,
document_type in out nocopy varchar2)
is
  l_docid pls_integer;
  l_errmsg varchar2(100) := 'The document is not found in the database';
  l_bdoc blob;
  l_data_type varchar2(100);
begin
  -- Determine the document to display from the
  -- Document ID
  --l_docid := to_number(document_id);
  l_docid := 7534745; --@@@tests
  -- Obtain the BLOB version of the document
  createBlobByRequestId ( l_docid );
  select outfile into l_bdoc from xxapworkflow_files where request_id = l_docid;
  -- Set the encoding as a part of the document_type
  document_type := 'application/pdf; encoding=base64; name=ZZ.pdf';
  -- Now copy the content to the document
  dbms_lob.Copy(document, l_bdoc, dbms_lob.getlength(l_bdoc));
exception
  when others then
    dbms_lob.WriteAppend(document, length(l_errmsg), l_errmsg);
    wf_core.context('WFMLR_TEST', 'getBLOBDoc', document_id);
    raise;
end getBlobDoc;

```

CLOB:

Aby wysłać w powiadomieniu załącznik należy ustawić powiadomienia w następujący sposób :

1. w workflow utworzyć atrybut typu Document; frame target: Full Window
2. przypisać atrybutowi wartość w postaci:

PLSQLCLOB:<nazwa pakietu>.<nazwa_procedury>/<id dokumentu>

3. Przypisać atrybut do message: type: Document, Source: Send, Frame: Full Window, Attach Content = YES
4. Utworzyć procedurę wskazaną w pkt. 2, nagłówek:

PROCEDURE <nazwa_procedury> (document_id IN VARCHAR2, display_type IN VARCHAR2, document IN OUT NOCOPY CLOB, document_type IN OUT NOCOPY VARCHAR2);

w procedurze musisz obsłużyć:

- skopiowanie do zmiennej document zawartości załącznika
- przypisanie do zmiennej document_type ciągu oznaczającego odpowiednio typ pliku i jego nazwę

U siebie w pakietach (dla CLOB) posługiwałem się następującymi wywołaniami:

```

l_doc :=
'<html><head><title>test</title></head><body>test</body></html>'; -- html,

```

który ma być wysłany w załączniku

```
dbms_lob.createTemporary(lc_doc, FALSE, dbms_lob.CALL);  
dbms_lob.writeAppend(lc_doc, LENGTH(l_doc), l_doc);  
dbms_lob.copy(document, lc_doc, LENGTH(l_doc), 1, 1);  
document_type:='application/html; name='||'wniosek.html';
```

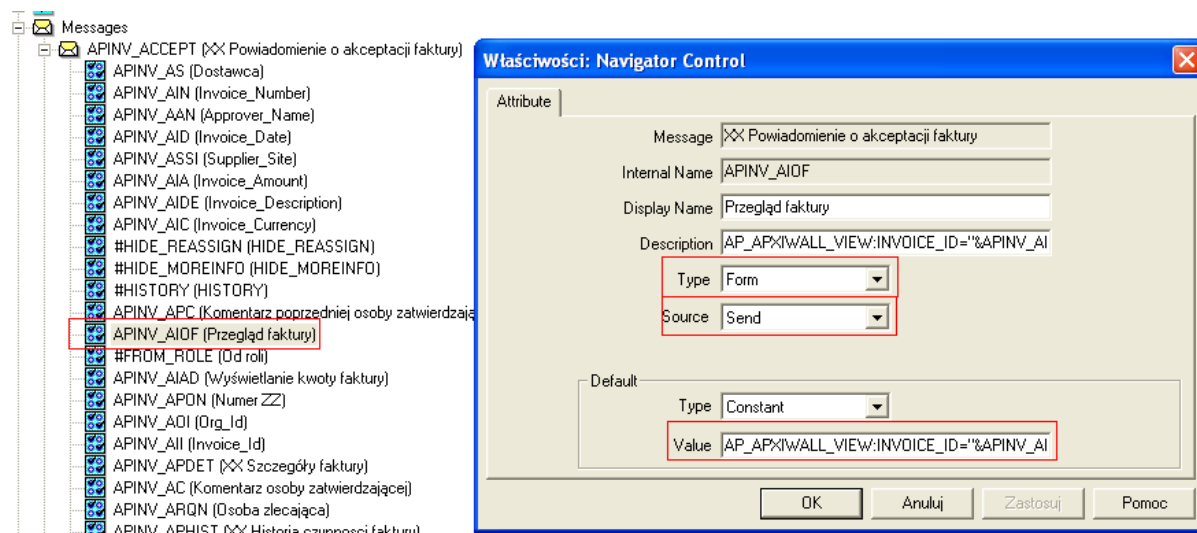
dla PDF oczywiście będzie parę zmian, jednak co do zasady rozwiązanie powinno być takie samo.

Jak załączyć do powiadomienia łącze do formularza Aplikacji

Adnotacje

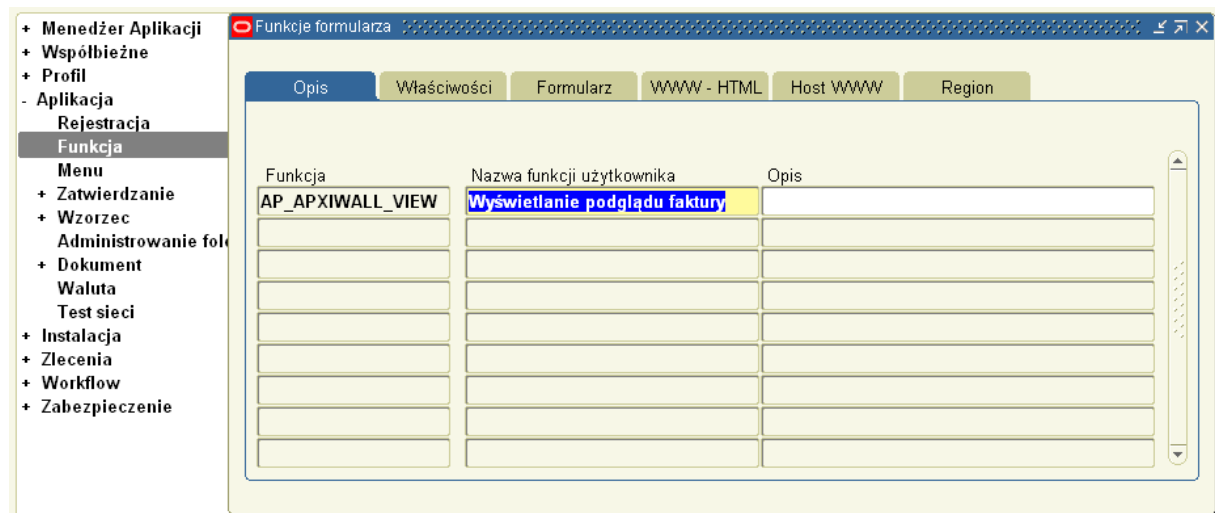
 [Przegląd faktury](#)

Dodaj do **message** atrybut zgodnie z rysunkiem



Value= AP_APXIWALL_VIEW:INVOICE_ID="%&APINV_AII" ORG_ID="%&APINV_AOI"
, gdzie:

&APINV_AII, &APINV_AOI – atrybuty workflow
AP_APXIWALL_VIEW – funkcja



przykład 2- podgląd ZZ/wydania:

```
IF l_document_type IN ('PO', 'PA') THEN
    l_open_form := 'PO_POXPOEP0:PO_HEADER_ID=' || '&' || 'DOCUMENT_ID' ||
        ' ACCESS_LEVEL_CODE="MODIFY"' ||
        ' POXPOEP0_CALLING_FORM="POXSTNOT"';

    wf_engine.SetItemAttrText ( itemtype => itemType,
                               itemkey  => itemkey,
                               aname    => 'OPEN_FORM_COMMAND' ,
                               avalue   => l_open_form);

ELSIF l_document_type = 'RELEASE' THEN
    l_open_form := 'PO_POXPOERL:PO_RELEASE_ID=' || '&' || 'DOCUMENT_ID' ||
        ' ACCESS_LEVEL_CODE="MODIFY"' ||
        ' POXPOERL_CALLING_FORM="POXSTNOT"';

    wf_engine.SetItemAttrText ( itemtype => itemType,
                               itemkey  => itemkey,
                               aname    => 'OPEN_FORM_COMMAND' ,
                               avalue   => l_open_form);

END IF;
```

Przykład 3- przegląd tabel int.PO

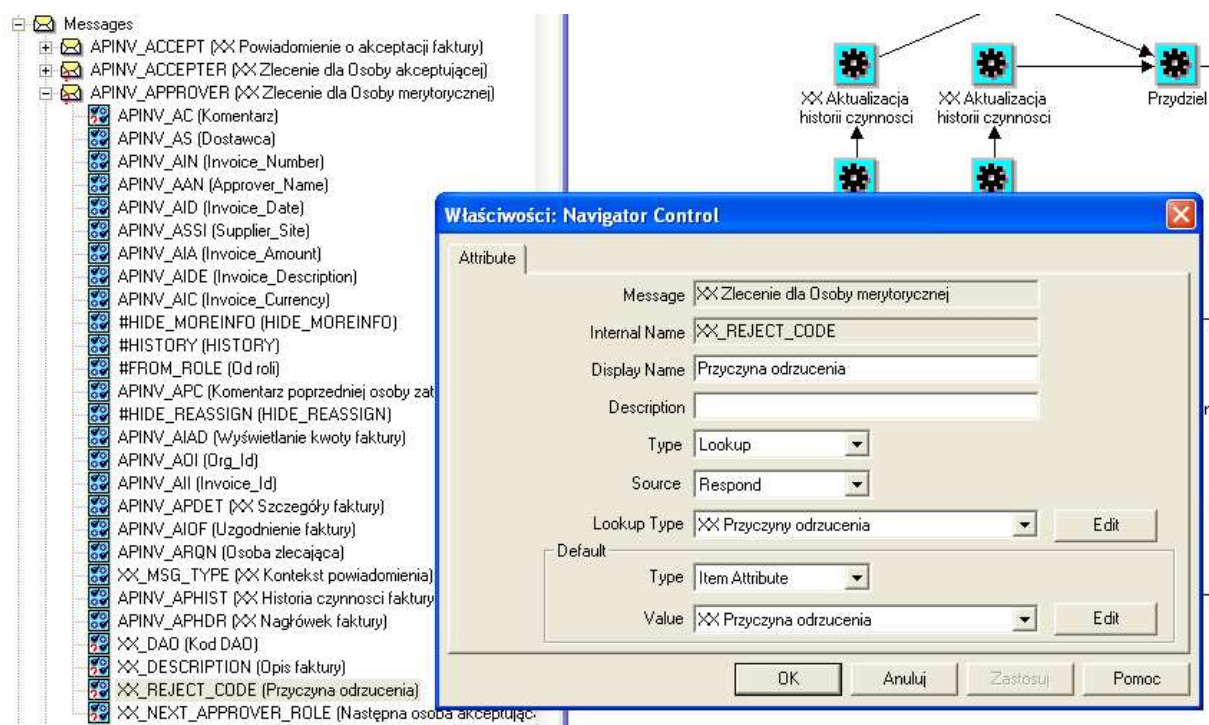
```
l_open_form := 'PO_POXPCATN:INTERFACE_HEADER_ID=' || '&' || 'INTERFACE_HEADER_ID' ||
    ' ACCESS_LEVEL_CODE="MODIFY"' || '&' || 'ACCESS_LEVEL_CODE';
```

Odczytanie komunikatu wprowadzonego na powiadomieniu i zapisanie go w bazie danych

Utwórz w message atrybut o właściwościach text, respond.

Jeżeli atrybut o typie **Respond** zostanie powiązany atrybutem globalnym, to atrybut globalny pełni dwie role:

- jego wartość jest wartością domyślną odpowiedzi (jak wskazuje nazwa na ramce - Default);
- wartość odpowiedzi jest zapisywana w tym atrybucie globalnym (mimo tego, że ramka sugeruje, że jest to tylko wartość domyślna). Wartość atrybutu globalnego może z kolei zostać zapisana w bazie danych przez funkcję wywoływaną za powiadomieniem.



Reguły czasu wolnego

Powiadomienia mogą być przekazywane do innych osób w dwóch przypadkach:

- 1/ zdefiniowano reguły czasu wolnego (automatyczne przekierowanie) – zob. select * from wf_routing_rules
- 2/ użyto przycisku reassign/delegate/transfer (manualne przekierowanie)

Uwagi:

- a) przycisk 2/ NIE są dostępne na powiadomieniach tylko, gdy systemowy atrybut #HIDE_REASSIGN ma ustawioną wartość Y (domyślnie nie ma tego atrybutu)
- b) reguła czasu wolnego RÓWNIEŻ NIE ZADZIAŁA, gdy atrybut #HIDE_REASSIGN ma ustawioną wartość Y

Są dwa rodzaje reguł czasu wolnego:

Nazwa	Kod techniczny (wf_routing_rules.action)	Opis
Oddelegowanie (przycisk Delegate)	FORWARD	Zastępujący dziedziczy uprawnienia od zastępowanego
Transfer (przycisk Transfer)	TRANSFER	Zastępujący NIE dziedziczy uprawnienia od zastępowanego (używa własnych uprawnień)
Dodatkowo: Ponowne przydzielanie (Przycisk Reassign)	FORWARD/ TRANSFER	To jest oddelegowanie lub transfer – tryb wybiera się na następnym ekranie

Inne wbudowane właściwości powiadomienia

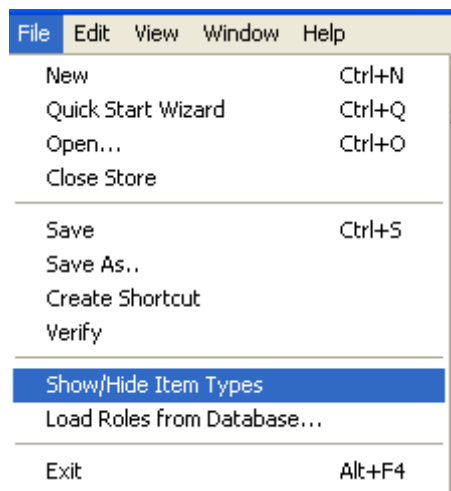
Szukaj z userguide opisu atrybutów specjalnego przeznaczenia, których nazwy zaczynają się od #, np. w **#HIDE_REASSIGN**

Skąd wziąć standardowe klocki: start/stop/join itd.

Szukaj item type „Standardowo”

Zob. też pakiet **WF_STANDARD**.

Jak dodać klocki z innych item type



Jak zakończyć workflow błędem

Do zweryfikowania

```
Fnd_Message.set_name('XXEXT', 'XXWF_APPROVER_REQUIRED'); --Wskaż Osobę merytoryczną w polu
Następna osoba akceptująca!
App_Exception.raise_exception;
result := Wf_Engine.eng_completed;
return;
```

lub

```
Wf_Core.context('Wf_Standard', 'ResetError', itemtype,
               itemkey, to_char(actid), funcmode);
Wf_Core.Token('COMMAND', cmd);
Wf_Core.raise('WFSQL_COMMAND');
```

lub

```
fnd_message.set_name('PA','WF_ACC_GENERATION');
fnd_message.set_token('KOMUNIKAT',l_message);
wf_engine.SetItemAttrText(itemtype=> itemtype,
                          itemkey=> itemkey,
                          aname=> 'ERROR_MESSAGE',
                          avalue=> fnd_message.get_encoded);
result := 'COMPLETE:FAILURE';
```

Diagnostyka workflow

Sprawdź, czy pakiety/funkcje do których odwołuje się workflow są valid / czy istnieją .
Nie => workflow nie uruchomi się, może też nie zwrócić żadnego komunikatu o błędzie.

GUI:

Autoryzacja:

Administrator systemu

lub

Administrator procesów typu workflow

Lub

Aplikacje WWW administratora procesów typu workflow

Funkcja: Workflow – administrator -> monitor statusu

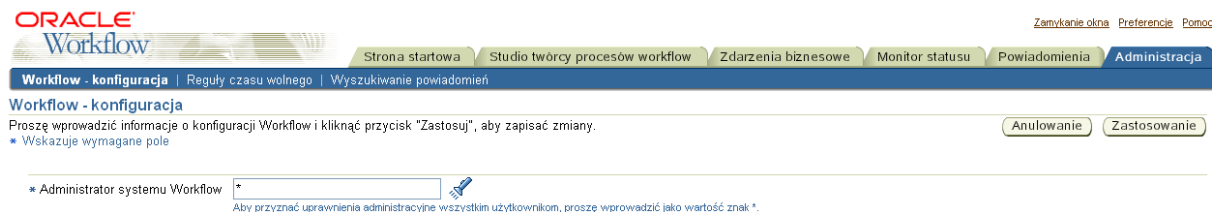
Loguj się na administrator procesów typu workflow (z reguły jest to użytkownik SYSADMIN)

Powyższe nie działa dla generatora kont dla ŚT – nie wiadomo, dlaczego.

Jeżeli nie możesz za pomocą GUI odnaleźć procesu

Zaloguj się jako SYSADMIN

Albo nadaj sobie uprawnienia administratora workflow:



albo wykonaj:

```
update wf_resources set text = '*' where name like 'WF_ADMIN_ROLE' --
z reguły TEXT=SYSADMIN
commit;
```

Diagnostyka procesu zatwierdzania zapotrzebowań, ZZ, zob. How approval list is prepared.doc.

SQL:

```
select * from WF_ITEM_ACTIVITIES_HISTORY_V --where item_type like 'FA%'
select * from wf_items where item_type = 'APINV'
```

Jak nadać uprawnienie administratora procesów workflow

Tylko administrator (SYSADMIN) może śledzić status przetwarzania procesów workflow innych użytkowników.

```
update wf_resources set text = '*' where name like 'WF_ADMIN_ROLE' --
z reguły TEXT=SYSADMIN
```

W miejsce * (gwiazdka oznacza dowolnego użytkownika) możesz również wpisać nazwę użytkownika lub nazwę autoryzacji.

Jak wyświetlić wszystkie atrybuty workflow

Przed procedurą End wstaw funkcję:

```
procedure showAttrs
( pitemtype in varchar2
, itemkey in varchar2
```

```

, actid      in      number
, funcmode   in      varchar2
, result     out      varchar2
) is

    procedure showAttributes (showNulls varchar2 default 'Y')
    is
        return_value varchar2(1000);
    begin
        xxmsz_tools.insertintoeventlog('START' , 'I', 'XXMSZ_EBS_TOOLS');
        for rec in
            ( select name, display_name
              from wf_item_attributes_vl
              where item_type = pitemtype --(select name from wf_item_types_vl where upper(display_name)
like upper(name))
            order by name)
        loop
            begin
                return_value := wf_engine.getitemattrtext(pitemtype,itemkey, rec.name );
                if showNulls = 'N' and return_value is null then
                    null;
                else
                    xxmsz_tools.insertintoeventlog(rec.name || ' = ' ||return_value || ' : ' ||
rec.display_name , 'I', 'XXMSZ_EBS_TOOLS');
                end if;
            exception
                when OTHERS then null;
            end;
        end loop;
        xxmsz_tools.insertintoeventlog('STOP' , 'I', 'XXMSZ_EBS_TOOLS');
    end;

begin
    showAttributes;
end;

```

Proces zatwierdzania faktur

Informacje podstawowe

Zatwierdzanie faktur odnosi się tylko faktur w module zobowiązania.

Faktury NAL nie podlegają procesowi zatwierdzania, tzn. tylko faktury ZOB podlegają procesowi zatwierdzania workflow - ze względu na pożądaną kontrolę procesu wydawania środków pieniężnych.

Typowy proces zatwierdzania wygląda następująco:

1. Rejestracja dokumentu (kancelaria)
2. Kontrola formalno-rachunkowa (księgowość)
3. Akceptacja- opis merytoryczny (Zaopatrzeniowiec).
4. Uzgodnienie z zamówieniem zakupu (Księgowość).
5. Akceptacja- zatwierdzenie do wypłaty (Przełożony zaopatrzeniowca)
6. Wypłata (księgowość)

Typowe kwestie do rozważenia podczas budowy workflow:

1. W zależności od parametrów faktury (z reguły chodzi o kwotę) na ścieżce może być różna liczba osób
2. Na każdym etapie może następować akceptacja lub odrzucenie; odrzucenie może polegać na anulowaniu dokumentu (przy zachowaniu odpowiednich procedur księgowych) lub na cofnięciu dokumentu do jednego z poprzednich etapów
3. Kontrola/akceptacja, w zależności od potrzeb może być wykonywana przez jednego lub wielu użytkowników – kolejno lub równolegle. Kryterium sukcesu może być zgoda wszystkich lub części uczestników procesu.
4. Ponowne uruchomienie odrzuconego workflow
5. Należy przewidzieć możliwość zastępowania użytkownika podczas jego nieobecności.

6. Czasy, po jakich dokument uznaje się za odrzucony/zaakceptowany itd. W praktyce stosuje się mechanizm polegający na tym, że po określonym czasie (po zajściu zdarzenia timeout) kontrola/akceptacja jest przekazywana kolejnemu użytkownikowi
7. Każda operacja zatwierdzania jest odnotowana w tabeli historii czynności zatwierdzania dokumentu.
8. Należy zapewnić zabezpieczenie przed wielokrotnym uruchomieniem workflow dla dokumentu gdy dokument jest w trakcie przetwarzania workflow
9. Ręczne zatwierdzenie workflow

Faktury mogą, lecz nie muszą podlegać zatwierdzaniu workflow – decyduje o tym pole wyboru w opcjach zobowiązań:

Opcje Zobowiązań (RAIFFEISEN BANK POLSKA S.A.)

Metody księgowania Transfer do KG Księgowanie płatności Waluta Dostawca **Faktura** Uzgadnianie

Główne

☒ Użyj oper. workflow do zatw. faktur

☐ Dozwolone wymuszenie zatwierdzania

☐ Wymagaj weryfikacji przed zatwierdzeniem

☒ Dozwolone korekty faktur zapłaconych

☒ Powtórne przeliczanie planowanej płatności

☐ Automatyczne tworzenie dekretacji frachtu

☒ Potwierdzenie daty jako numeru faktury

☒ Dozwolone zatwierdzenie w trybie online

☐ Dozwolona zmiana kategorii dokumentu

Konto frachtu

Podstawa daty KG

☒ Data faktury

☐ Data systemowa

☐ Data faktury/otrzymania towaru (H)

☐ Data system./otrzymania towaru (G)

Przedpłata

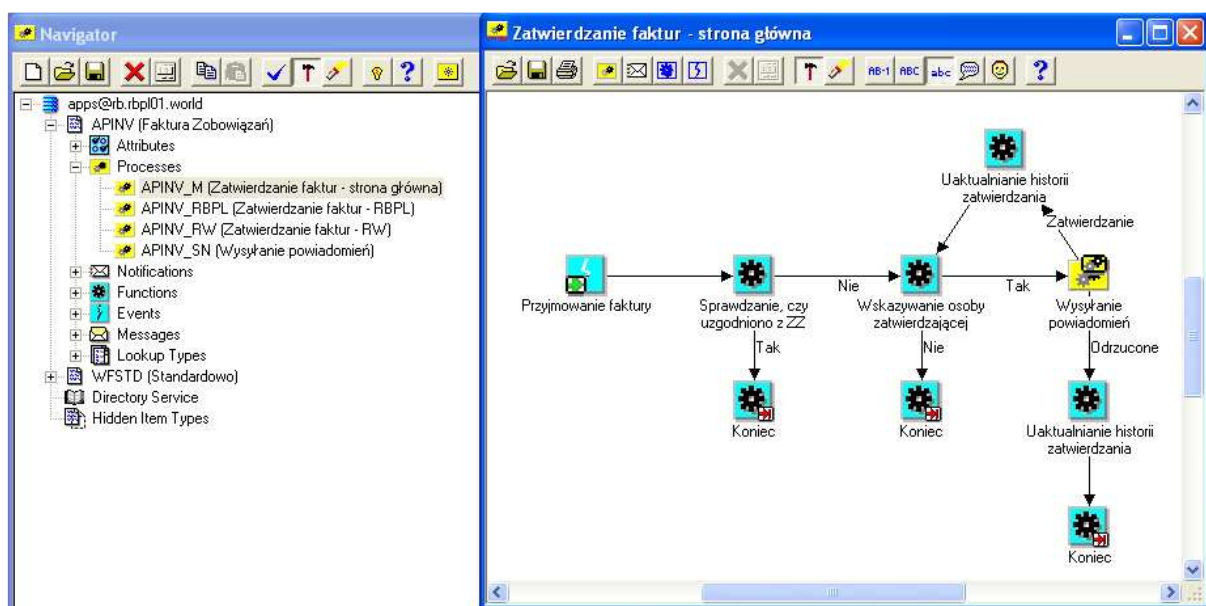
Warunki płatn.

Dni rozliczenia

☐ Twórz konta przedpłat po uzgodnieniu

Po włączeniu zatwierdzania, w oknie czynności faktur pojawia się pole wyboru do uruchomienia workflow, a status zatwierdzania faktury po wprowadzeniu to wymagane (jeżeli pole wyboru nie zostanie zaznaczone, wówczas status zatwierdzania ma wartość niewymagane):

Standardowo zatwierdzanie jest wykonywane przez workflow o nazwie „Faktura Zobowiązania”:



Standardowy proces workflow wybiera kolejne osoby w ścieżce zatwierdzania (zob. wskazanie osoby zatwierdzającej) zgodnie z konfiguracją określaną za pomocą standardowego modułu *Oracle Approvals Management*, dla typu transakcji *Payables Invoice Approval*. Moduł jest dostępny z poziomu autoryzacji *Zarządzanie Zatwierdzeniami: Administrator aplikacji*.

Jednak w dotychczasowych wdrożeniach moduł ten nie był wykorzystywany (ze względu na skomplikowaną obsługę oraz pewne niedoskonałości).

ORACLE

Approvals Management

[Ustawianie typu transakcji](#) [Powrót do portalu](#)

Attributes Conditions Actions Groups **Rules** Test Admin

Wszystkie typy transakcji

Wybór typu transakcji

Typ transakcji: Payables Invoice Approval

Kontynuacja

[Atrybuty](#) | [Warunki](#) | [Zatwierdzenia](#) | [Grupy](#) | **[Reguły](#)** | [Test](#) | [Administracja](#)
[Portal](#) | [Ustawianie typu transakcji](#)

Interakcja z użytkownikami:

1. Interakcja z użytkownikami jest realizowana za pomocą formularzy aplikacji lub za pomocą email, z tym, że email nie obsługuje-z oczywistych względów – list wartości. Mail zapewnia jedynie wybór jednej z wielu decyzji oraz wprowadzanie komentarzy. W praktyce użytkownicy kluczowi używają formularzy aplikacji, a użytkownicy których zadaniem jest tylko akceptacja lub odrzucenie dokumentów – poczty elektronicznej.
2. Workflow pobiera adresy email z poziomu pracownika, a jeżeli pracownik nie został zdefiniowany – z poziomu użytkownika:

Autoryzacja	Aplikacja	Opis	Grupa zabezpieczeń	Do
Twórca aplikacji	Biblioteka Obiektów		Standardov	07-12-2006
RBPL Powiadomienia	Biblioteka Obiektów		Standardov	08-12-2006
Administrator systemu	Administracja System		Standardov	07-12-2006
RBPL KG Superużytkownik	Księga Główna		Standardov	07-12-2006
RBPL Discoverer - Superu	BIS - Aplikacje		Standardov	27-12-2006

Budowa własnego procesu workflow:

1. Standardowy proces workflow jest uruchamiany tylko dla dokumentów zbilansowanych (tj. kwota z nagłówka = sumie kwot z linii). W praktyce zdarza się, że proces workflow musi zostać uruchomiony po wprowadzeniu samego nagłówka. W takim przypadku należy:
 - a. Wyłączyć standardowy workflow w opcjach zobowiązań
 - b. Dodać personalizację, która zmieni status faktury na „wymagane”
 - c. Dodać do menu polecenie uruchamiające nowy workflow

Standardowe zlecenie inicjujące workflow faktur: *Proces workflow zatwierdzania faktury*

Można zastosować również odwrotne rozwiązanie – poniżej stosowny opis (fragment mail)

W załączeniu tekst personalizacji dla formularza APXINWKB.

Jeśli chodzi o ustawienie wymagalności zatwierdzenia, to personalizacja oczywiście nie działa dla dokumentów tworzonych przez interfejs. Generalnie, są dwie opcje dla zmiany statusu:
 1. zastosowana w PZU - tj. WŁĄCZENIE opcji wymagalności zatwierdzenia w opcjach Zobowiązań, i ustawianie wartości 'NOT REQUIRED' dla kategorii dokumentów, które zatwierdzenia NIE WYMAGAJĄ.
 2. zastosowana w RB - tj. WYŁĄCZENIE opcji wymagalności zatwierdzenia w opcjach Zobowiązań, i ustawianie wartości 'REQUIRED' dla kategorii dokumentów, które zatwierdzenia WYMAGAJĄ

Uwaga: dla opcji 1, w oknie czynności pojawia się opcja 'Uruchomienie zatwierdzania' - standardowo dzieje się to inaczej (poprzez business event) niż 'moją' metodą w pakiecie XX_AP_APPROVAL_PKG, gdzie bezpośrednio tworzę i uruchamiam proces. Jeśli zdecydujecie się domyślnie włączyć wymagalność zatwierdzenia, ale workflow wywoływać ręcznie, należy ukryć opcję w oknie Czynności.

 1. Opis: APINV. Czynności menu
 Zdarzenie: WHEN-NEW-FORM-INSTANCE

1. Menu: SPECIAL31: Uruchomienie procesu akceptacji dokumentu:
 :INV_SUM_FOLDER

1. Opis: APINV. Ustawienie wymagalności zatwierdzenia
 Zdarzenie: WHEN-VALIDATE-RECORD
 Obiekt: INV_SUM_FOLDER
 Warunek: :INV_SUM_FOLDER.INVOICE_ID is null and
 :INV_SUM_FOLDER.WFAPPROVAL_STATUS = 'NOT REQUIRED' and
 Xx_Ap_Approval_pkg.doc_approval_required(:INV_SUM_FOLDER.invoice_type_lookup_code, :INV_SUM_FOLDER.doc_category_code) = 1

1. Właściwość: Pozycja: INV_SUM_FOLDER.WFAPPROVAL_STATUS: VALUE: REQUIRED
 2. Właściwość: Pozycja: INV_SUM_FOLDER.WFAPPROVAL_STATUS_DSP: VALUE: Wymagane
 3. Wbudowane: DO_KEY: COMMIT_FORM

2. Opis: APINV. Weryfikacja warunków dla uruchomienia procesu akceptacji
 Zdarzenie: SPECIAL31
 Warunek:
 XX_AP_APPROVAL_PKG.VERIFY_APPROVAL(NVL(:INV_SUM_FOLDER.invoice_id,0)) is not null

1. Komunikat: Błąd:
 =XX_AP_APPROVAL_PKG.VERIFY_APPROVAL(NVL(:INV_SUM_FOLDER.invoice_id,0))

3. Opis: APINV. Uruchomienie procesu akceptacji
 Zdarzenie: SPECIAL31
 Warunek:
 XX_AP_APPROVAL_PKG.VERIFY_APPROVAL(NVL(:INV_SUM_FOLDER.invoice_id,0)) is null

1. Właściwość: Pozycja: INV_SUM_FOLDER.WFAPPROVAL_STATUS: VALUE: INITIATED
 2. Właściwość: Pozycja: INV_SUM_FOLDER.WFAPPROVAL_STATUS_DSP: VALUE: W toku
 3. Właściwość: Pozycja: INV_SUM_FOLDER.APPROVAL_ITERATION: VALUE: =TO_CHAR(TO_NUMBER(NVL(:INV_SUM_FOLDER.APPROVAL_ITERATION, 0))+1)
 4. Wbudowane: DO_KEY: COMMIT_FORM
 5. Wbudowane: FORMS_DDL: ='begin
 xx_ap_approval_pkg.init_approval('||:inv_sum_folder.invoice_id||'); end;'
 6. Komunikat: Wyświetlanie: ='Uruchomiono proces zatwierdzania dla dokumentu '||:inv_sum_folder.invoice_num

Wskazówki techniczne

1. statusy zatwierdzania dla faktury i status zatwierdzania w tabeli czynności (ap_invoices_all.wfapproval_status i ap_inv_aprvl_hist_all.response)

w przypadku ap_invoices_all.wfapproval_status korzystaj tylko ze standardowych wartości - `select * from fnd_lookup_values where enabled_flag =`

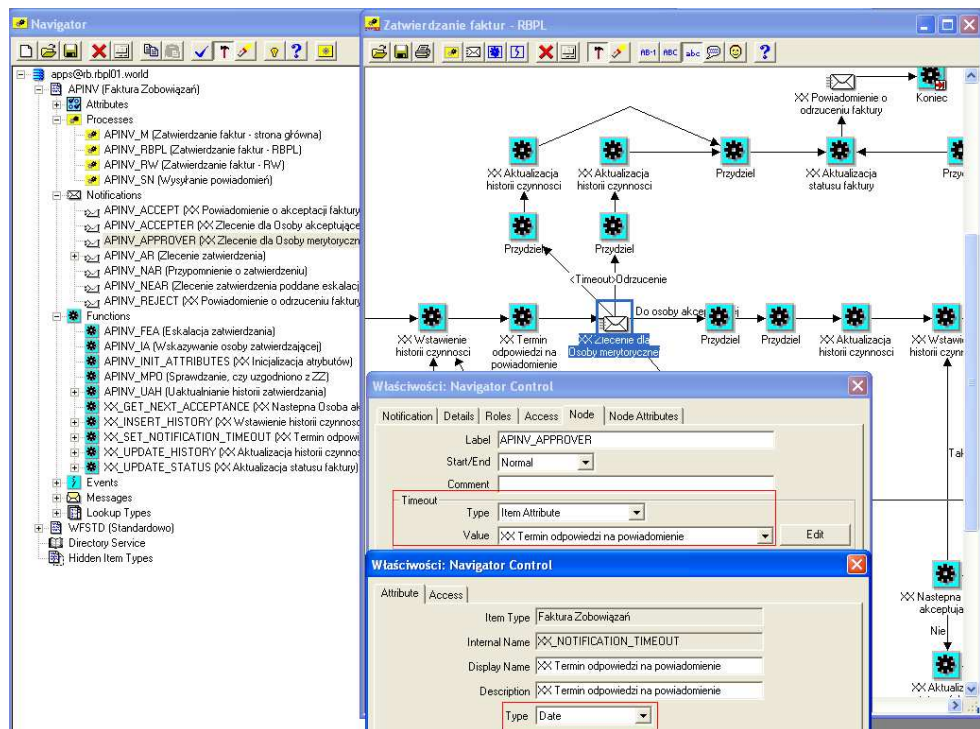
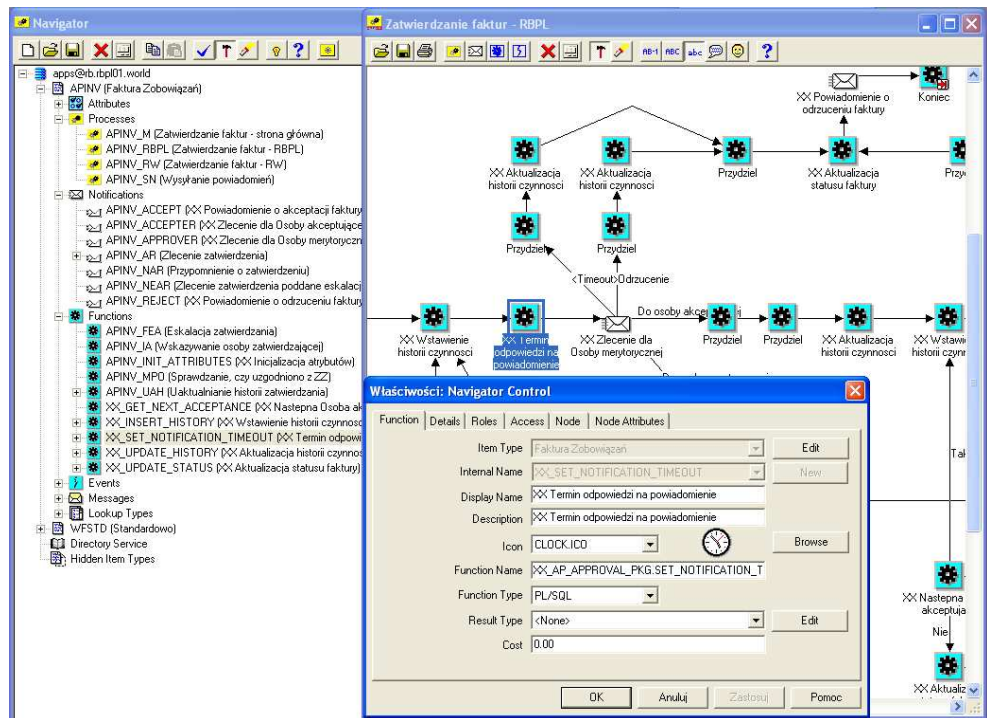
'y' and language = userenv('lang') and lookup_type = 'ap_wfapproval_status'. kolumna wfapproval_status jest sprawdzana przez systemowe procesy i nie może zawierać innych, nietypowych wartości.

w przypadku ap_inv_aprvl_hist_all.response możesz dopisywać nowe wartości - `select * from wf_lookups_tl where lookup_type = 'xx_approval_status' and language = 'pl'`.

2. Kolumny w tabeli ap_invoices_all związane z workflow:

Kolumna	Opis
WFAPPROVAL_STATUS	Status of invoice in the Invoice Approval Workflow process Lista wartości dla tej kolumny jest zdefiniowana za pomocą lookupa o nazwie AP_WFAPPROVAL_STATUS (zob. select * from ap_lookup_codes where lookup_type = 'AP_WFAPPROVAL_STATUS')
REQUESTER_ID	Zlecający. Requester of invoice is used by the Invoice Approval Workflow process to generate the list of approvers
VALIDATION_REQUEST_ID	Identifier for the Invoice Validation concurrent request that is currently processing the invoice.
APPROVAL_ITERATION	Count of times invoice has passed through the Invoice Approval Workflow process

3. Process workflow można uruchomić za pomocą polecenia Wf_Engine.CreateProcess (uruchamia od razu proces workflow) lub za pomocą pakietu ap_approval_pkg (wstawia proces workflow do kolejki – możliwe kilkuminutowe opóźnienie). Zaleca się metodę 1 – zob. XX_AP_APPROVAL_PKG.
4. Pocztę dla workflow konfiguruje administrator Aplikacji (Administrator systemu -> Notification mailer)
5. Standardowy mechanizm, wyzwalający zdarzenie *timeout* o typie *Relative time* nie uwzględnia, że soboty i niedziele są dniami wolnymi od pracy. Problem ten rozwiązuje się poprzez wyliczanie terminu przez własną funkcję i przekazanie terminu przez zmienną:



6. Opis funkcji (klocków) z których zbudowano workflow:

- **Init** – inicjuje wartości atrybutów globalnych

- **Termin odpowiedzi na powiadomienie** – wylicza parametr *timeout* dla powiadomienia (ignoruje soboty i niedziele). Występuje zawsze przed uruchomieniem powiadomienia. Przekazanie parametru przez zmienną globalną *XX Termin odpowiedzi na powiadomienie* (*XX_NOTIFICATION_TIMEOUT*)

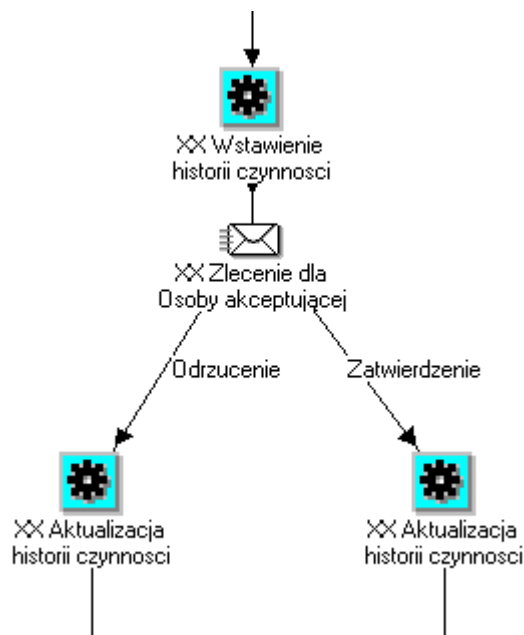
- **Aktualizacja statusu faktury** – aktualizuje status przetwarzania workflow faktury (WFWORKFLOW_STATUS). Uruchamiaj na początku (żeby było widać, że faktura jest przetwarzana) i na końcu procesu (aby nadać jej odpowiedni status).

- **Wstawianie historii czynności** – wstawia rekord do tabeli ap_inv_aprvl_hist_all (Narzędzia – przegląd historii zatwierdzania)

- **Aktualizacja historii czynności** – aktualizuje rekord w tabeli ap_inv_aprvl_hist_all

Używaj klocków wstawienie/aktualizacja na początku i końcu workflow oraz podczas zatwierdzeń.

Wstawienie wykonuje się przed powiadomieniem (tak, aby inni użytkownicy widzieli, że faktura aktualnie jest w powiadomieniu u danej osoby), a aktualizację wykonuje się po powiadomieniu- zob. rysunek poniżej.



Przykładowe wdrożenia:

- PZU, projektant: Jacek Nosowski, wykonawca: Jacek Nosowski, Michał Kalisz. Proces zatwierdzania zakłada funkcję lidera, umieszczonego centralnie w procesie. Lider decyduje o tym, kto powinien dokument opisać merytorycznie, a następnie zaakceptować, czy czym opis oraz akceptacja może być wykonywana przez kilka osób – w zależności od decyzji lidera. Lider był zapisany w polu zatwierdzający. Workflow jest zintegrowany z systemem Klienta przechowującym zeskanowane wersje dokumentów. Repozytorium jest uzupełniane o dodatkowe dokumenty w trakcie procesu workflow.

Zabezpieczenie przed wielokrotnym uruchomieniem workflow dla dokumentu gdy dokument jest w trakcie przetwarzania workflow; ponowne uruchomienie odrzuconego workflow

Standardowo istnieje zabezpieczenie uniemożliwiające uruchomienie workflow o już istniejącym identyfikatorze (itemKey). itemKey = id dokumentu.

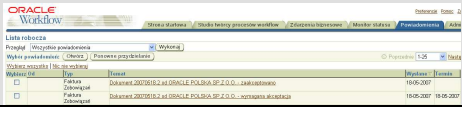
Nie zależnie od powyższego po uruchomieniu workflow ustaw status dokumentu wskazujący, że workflow został uruchomiony. Na końcu workflow zmieniaj ten status.

Czasami jednak workflow dla danego dokumentu musi być uruchamiany wielokrotnie, np. gdy dokument zostanie niesłusznie odrzucony.

Aby umożliwić ponowne uruchomienie workflow itemKey składa się w praktyce z id dokumentu oraz nr wersji uruchomienia.

itemKey = <id dokumentu>_<wersja>. (np. invoice_id i approval_iteration w tabeli ap_invoices_all)

FAQ

Pytanie	Odpowiedź
Jak wygląda interfejs użytkownika końcowego ?	Zob. R:Administrator procesów typu workflow M:Workflow-Administrator / powiadomienia 
Czy do powiadomienia można załączać załączniki (np. dokument pdf wygenerowany przez zlecenie współbieżne)	Tak
Czy po uruchomieniu workflow za pomocą menu trzeba odświeżyć formularz, żeby zobaczyć zmianę w polu status zatwierdzania ? A może zmieniasz status personalizacją?	Personalizacją
Gdy zostanie uruchomiony workflow, to po kliknięciu w pole Zatwierdz. na formularzu faktury pojawia się komunikat o błędzie : Niewłaściwy argument procedury INV_SUM_UTILITIES3.call_item_handler: FIELD_NAME='INV_SUM_FOLDER.WFAPPROVAL_STATUS_DSP'. FRM-40735: Wyzwalacz WHEN-NEW-ITEM-INSTANCE zgłoszony nieobsługiwany wyitek ORA-20002.	Zablokować pole
Jak powinien się zachować system gdy workflow zostanie anulowany?	Nie zakłada się możliwości anulowania procesu workflow. Proces można odrzucić i a następnie uruchomić ponownie
Co zrobić żeby workflow nie wysłał zaległych maili po skonfigurowaniu mailera ?	Nie wyrzuci, bo zaległe maile są oznaczone jako ERROR (wf_notifications.mail_status) W razie co można wykonać: update wf_notifications set mail_status = NULL where mail_status = 'MAIL'
Każdy użytkownik może, korzystając z autoryzacji preferencje ustawić sposób przyjmowania powiadomien. Ale jak ustawić hurtem sposób odbierania powiadomien mailem ?	Skrypt aktualizuje preferencje w tabelach WF_LOCAL_ROLES i FND_USER_PREFERENCES. Skrypt uruchamia się w dwóch trybach - odpowiednio kasując komentarz (1) i komentarz (2). (1) - aktualizacja preferencji użytkownika

	(2) - insert preferencji użytkownika właściwie, to należałoby to robić jednocześnie, na zasadzie albo-albo. Jeśli chodzi o kod preferencji dla wysyłania powiadomień, 'QUERY' oznacza powiadomienia tylko w Aplikacji, a 'MAILTEXT' - powiadomienia generowanie w trybie tekstowym, a 'MAILHTML', powiadomienia generowanie w HTML. ----- <pre> declare begin for xx in (select * from wf_users w where orig_system = 'PER' -- 1 and w.name in (select user_name from FND_USER_PREFERENCES where PREFERENCE_NAME = 'MAILTYPE' and user_name like w.name and PREFERENCE_VALUE <> 'QUERY') -- 2 and not exists (select null from FND_USER_PREFERENCES where PREFERENCE_NAME = 'MAILTYPE' and user_name like w.name)) LOOP -- 1 update FND_USER_PREFERENCES set PREFERENCE_VALUE = 'QUERY' where user_name = xx.name and PREFERENCE_NAME = 'MAILTYPE'; -- 2 insert into FND_USER_PREFERENCES values (xx.name, 'WF', 'MAILTYPE', 'QUERY'); END LOOP; end; ----- </pre>
W historii zatwierdzeń niektóre statusy pojawia³¹ się z b³ędym opisem, np. ACCEPTANCE REQUIRED, WORKFLOW LAUNCHED -> oczekuj¹ce. Ale np. MANUALLY APPROVED, REJECTED pojawiaj¹ się prawidł³owo.	1. AP_INVOICES_ALL.WFAPPROVAL_STATUS : używaj wyłącznie systemowych statusów (select * from fnd_lookup_values where enabled_flag = 'Y' and language = userenv('LANG') and lookup_type = 'AP_WFAPPROVAL_STATUS' and language = 'PL') 2. ap_inv_aprvl_hist_all.response : możesz używać własnych statusów.

czy użytkownicy, którzy bed ¹ korzystali tylko z powiadomien mailowych rowniez musza miec zalożone konto w oracle applications?	
czy kiedykolwiek zdarza sie używaæ kilka interakcji workflow (bo przeciez workflowow nie anulujemy z definicji) ?	Tak, gdy dokument zostanie odrzucony
Ustawi ³ em na powiadomieniu timeout - jedna minuta. Sprawdzi ³ em po 5-u minutach stan workflow - timeout nie zadzia ³ - workflow nadal czekal od odpowiedz z powiadomienia. Czy jeszcze coś trzeba ustawić ?	Tak – uruchomić cykliczne uruchamianie zlecenia: workflow-proces w tle.
proedura get_invoice_header. W ktorym miejscu jest ustawiany parametr document_id ? (tzn. skad notyfikacja, ktora uruchamia procedure get_invoice_header wie, ze chodzi o invoice_id ?)	W init. Numer jest po znaku /
Zastępstwa: Sk ¹ d system wie, że użytkownik jest nieobecny i trzeba go zast ¹ pić?	
Czy jest standardowy mechanizm do sprawdzania zastępstw ? (funkcja daj_zastępcę)	To się dzieje samo.
Dlaczego na serwerze PZU by ³ y tylko dwa foldery : discard i accept (chyba). Co z pozosta ³ ymi odpowiedziami użytkowników ?	Akcept – w poprawnym formacie Discard- w złym formacie
Co to jest frame target ? (atrybut o typie dokument)	

Pytania:

1. Testowe głosowanie: WFSTD_DEFAULT_VOTE (Domyślny komunikat głosowania "Tak/Nie") – dlaczego message nie ma atrybutu tak/nie o typie respond ?
2. Czy proces „testowe głosowania” można uruchomić ?

jak się tworzy rolę zawierającą dwóch użytkowników ?

```

declare
    vrole_name          varchar2(200);
    vrole_display_name  varchar2(200);
begin
    vrole_name          := null;
    vrole_display_name := null;
    wf_directory.CreateAdHocRole(
        role_name => vrole_name
    ,role_display_name => vrole_display_name
    ,role_users => 'MSZYMCZAK MZAJAKALA'
    );
    commit;
    raise_application_error(-20000, vrole_name);
end;

begin
    wf_directory.RemoveUsersFromAdHocRole('REQUESTORS', 'MSZYMCZAK');
end;

begin
    wf_directory.AddUsersToAdHocRole('REQUESTORS', 'MSZYMCZAK');
end;

select * from wf_roles where name like '%HOC%'
select * from WF_LOCAL_USER_ROLES where role_name = '-WF_ADHOC-11'

```

Notification mailer

Jak skonfigurować workflow tak aby wysyłał maile

W celu skonfigurowania i uruchomienia usługi "Workflow Notification Mailer" wymagany jest dostęp do poczty elektronicznej. Usługa musi być dostępna z poziomu Aplikacji.

Konto pocztowe dla usługi "Workflow Notification Mailer" może być założone:

1/ na obecnie funkcjonującym serwerze pocztowym lub

2/ na nowym, dedykowanym serwerze pocztowym

Rekomendujemy pierwsze rozwiązanie.

Aby wykonać konfigurację usługi "Workflow Notification Mailer" prosimy o udostępnienie dostępu do usługi poczty elektronicznej i przesłanie następujących parametrów:

Nazwa węzła programu obsługi poczty = ?WFMAIL

Protokół danych przychodzących = IMAP

Nazwa serwera danych przychodzących = ?

Nazwa użytkownika = ?

Hasło = ?

Folder "Skrzynka odbiorcza" = INBOX

Protokół danych wychodzących = SMTP

Nazwa serwera danych wychodzących = ?

Dane wychodzące - limit oczekiwania na połączenie = 120

Włączone wychodzące połączenie SSL = ?

Folder "Przetworzone" = PROCESS

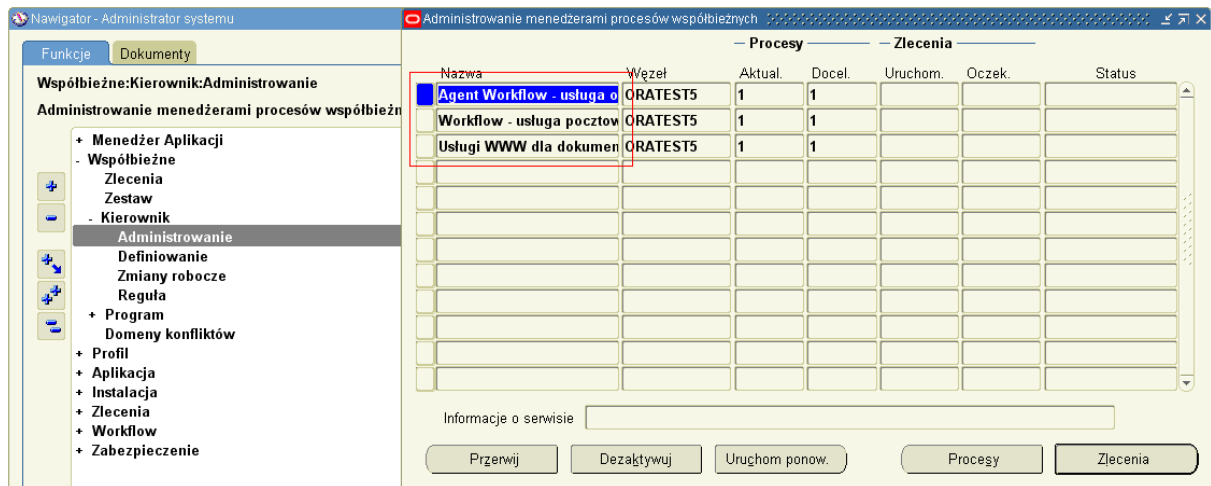
Folder "Odrzucone" = DISCARD

Diagnostyka notification mailer

1. Sprawdź status mailera – powinien być uruchomiony
2. Sprawdź komunikaty o błędach w powiadomieniach
 - a. `select name , text_value from wf_notification_attributes where name like 'MAIL%'`
3. Sprawdź log mailera
4. Sprawdź powiadomienia wysłane do użytkownika SYSADMIN

Po wprowadzeniu zmian w pakietach notification mailer na trwałe zachowuje wewnętrznie błędną informację „istniejący stan pakietów został odrzucony” (ORA-04061: existing state of has been invalidated) – wysyłane jest powiadomienie do użytkownika SYSADMIN. Z tego powodu, po wprowadzeniu zmian w pakietach zatrzymaj i uruchom usługi mailera - rysunek poniżej. Filtr %Workflow%. Przyciski *przerwij*, a następnie uruchom *ponow/aktywuj*.

Nie pomogło: zrestartuj serwer Apache.



Rapid Install Release 11i

Rapid Install Setup

[Home](#)

Logon

[Apps Logon Links](#)

Documentation

[11i Online help](#)

Server Administration

[Machine Admin Scripts](#)

Logon Options for Oracle Applications 11i

Logon Information for Instance ITD2

Logon to E-Business Suite Home

[E-Business Home Page](#)

Logon to Oracle Applications Manager

[Oracle Applications Manager](#)



System Configuration

Hosts
AutoConfig
License Manager

Application Services

Generic Services
Request Processing Managers
Transaction Managers

Workflow

Home
Work Item Metrics
Agent Activity
Background Engines
Notification Mailer
Service Components
Purge

Concurrent Requests

Submit New
Pending
Running
Completed (Last Hour)

Service Fulfillment Manager

Service Fulfillment Manager

Others

Applications Manager Log

Last Updated : 25-05-2007 13:56:12

Filter Type (Internal) WF_MAILER Go

Select a Service Component and ...						
Edit View Log View Details View Event History Delete						
Wybierz	Name	Status	Type	Startup Mode	Container Type	Container
☒	Workflow Notification Mailer	✓ Działające	Workflow - obsługa poczty	Automatyczne	Aplikacje Oracle - system zarządzania usługami GSM	Workflow - usługa pocztowa

WSKAZÓWKA GSM - Generic Service Management

Add to Support Cart Test Mailer Set Override Address View Event History View Log Edit

Definition

ID	10006	Name	Workflow Notification Mailer
Startup Mode	Automatyczne	Container Type	Aplikacje Oracle - system zarządzania usługami GSM
Inbound Agent	Workflow Notification In Queue	Outbound Agent	Workflow Notification Out Queue
Correlation ID			

Status Infor

Details

Container	Workflow - usługa pocztowa	Maksymalna liczba błędów	10000
Liczba wątków danych przychodzących	1	Liczba wątków danych wychodzących	1
Poziom dziennika	Błąd	Limit oczekiwania na odczyt programu przetwarzania	10
Minimalny czas uśpienia pętli programu przetwarzania	5	Maksymalny czas uśpienia pętli programu przetwarzania	60
Czas uśpienia pętli po wystąpieniu błędu programu przetwarzania	60	Limit oczekiwania na zamknięcie podczas odczytu programu przetwarzania	Y

Email Servers

Gdzie się ustawia język dla powiadomień

Autoryzacja: powiadomienia

Preferencje ogólne

Języki

Język bieżącej sesji	Polski	▼	i
Domyślny język aplikacji	Angielski (USA)	▼	i

Dostępność

Funkcje dostępności	Brak	▼	i
---------------------	------	---	---

Regionalne

Państwo	Polska	▼	
Format daty	dd-MMM-yyyy (23-kwi-2008)	▼	
Strefa czasowa		▼	
Format liczb	10.000,00	▼	
Waluta		▼	
Kodowanie znaków dla klienta	Europa środkowa (Windows)	▼	i

Zmiana hasła

Alias	Elżbieta Czarnocka
Stare hasło	
Nowe hasło	
Powtórzenie hasła	

Jak przełączyć sposób obierania maili przez użytkowników na mail

1. Każdy użytkownik może samodzielnie ustawić sposób odbierania powiadomień, korzystając z autoryzacji Preferencje.

Przydomek	Maciej Szymczak
Język	Polski
Klient - kodowanie znaków	Europa środkowa (Windows)
Format daty	31-12-2000
Format liczb	10,000.00
Terytorium	Stany Zjednoczone
Proszę wysłać mi powiadomienia drogą elektroniczną	Poczta w formacie HTML z załącznikami
Główny węzeł dokumentu	E-Business Suite
Stare hasło	
Hasło	
Powtórzenie hasła	

2. Sposób 1 może okazać się zbyt czasochłonny lub niemożliwy do wykonania. Wówczas wykonaj:

```
-- sets for all the users notification_type = 'MAILHTML' (Poczta w formacie HTML w
załącznikami)
declare
  vpreference_value fnd_user_preferences.preference_value%type := 'MAILHTML';
begin
  --update wrong preferences
  for rec in ( select * from wf_users w
              where orig_system = 'PER'
                --and name = 'MSZYMCZAK'
                and status = 'ACTIVE'
                and w.name in (select user_name from fnd_user_preferences where
preference_name = 'MAILTYPE' and user_name like w.name and preference_value <>
vpreference_value)
              ) loop
    update wf_local_roles set notification_preference = vpreference_value where name = rec.name
    and orig_system = 'PER';
    update fnd_user_preferences set preference_value = vpreference_value where user_name =
rec.name and preference_name = 'MAILTYPE';
  end loop;
  --insert preferences
  for rec in ( select * from wf_users w
              where orig_system = 'PER'
                --and name = 'MSZYMCZAK'
                and status = 'ACTIVE'
                and not exists (select 1 from fnd_user_preferences where preference_name =
'MAILTYPE' and user_name like w.name)
              ) loop
    update wf_local_roles set notification_preference = vpreference_value where name = rec.name
    and orig_system = 'PER';
    insert into fnd_user_preferences (user_name, module_name, preference_name, preference_value)
    values (rec.name, 'WF', 'MAILTYPE', vpreference_value);
  end loop;
  commit;
end;
--test:
--select * from wf_users where name = '...' and orig_system = 'PER'
--select * from fnd_user_preferences where user_name = '...'
```

Tabele workflow

Zob. workflow.sql

Zob. też ERD

Interakcja z użytkownikami

Interakcja z użytkownikami może być realizowana za pomocą:

- natywnego mechanizmu powiadomień
- lub za pomocą klienta poczty elektronicznej
- lub za pomocą poczty elektronicznej przez przeglądarkę np. Outlook Web Access uruchamiany za pomocą przeglądarki Internet explorer

Cechy wymienionych sposobów dostępu:

Sposób dostępu	Cechy
Natywny mechanizm powiadomień	<i>Metoda dostępu:</i> strona WWW Oracle Applications <i>Obsługa list wartości:</i> Tak <i>Obsługa decyzji jedna z wielu:</i> Tak (przez wybranie przycisku) <i>Obsługa komentarzy:</i> Tak <i>Wymagalność pól:</i> tak
Klient poczty elektronicznej	<i>Metoda dostępu:</i> uruchomienie klienta poczty <i>Obsługa list wartości:</i> Nie <i>Obsługa decyzji jedna z wielu:</i> Tak (przez wybranie łącza, które wygeneruje odpowiedź zwrotną) <i>Obsługa komentarzy:</i> Tak (w odpowiedzi zwrotnej) <i>Wymagalność pól:</i> tak (powtórny mail)
Poczta elektroniczna – przeglądarka	<i>Metoda dostępu:</i> przeglądarka <i>Obsługa list wartości:</i> Nie <i>Obsługa decyzji jedna z wielu:</i> Tak, ale występują poważne utrudnienia. Po naciśnięciu łącza wyświetla się zawartość wiadomości odpowiedzi w kliencie poczty elektronicznej. Odpowiedź trzeba przygotować ręcznie poprzez skopiowanie zawartości maila z powrotem do przeglądarki. <i>Obsługa komentarzy:</i> Tak (w odpowiedzi zwrotnej) <i>Wymagalność pól:</i> tak (powtórny mail)

Popychanie workflow

Popchanie workflow wykonuje zlecenie

Workflow-proces w tle

Workflow trzeba popychać:

- gdy wywołana funkcję ze statusem „DEFFER”
- w przypadku timeoutów

Jak z poziomu kodu popchnąć proces workflow (zatrzymany z powodu błędu) lub zasymulować odpowiedź na powiadomienie

Opisane poniżej API jest wykorzystywane np. przez panel administratora procesów workflow

```
Wf_Engine.HandleError(item_type, item_key, activity, comflag, result);
```

- * item_type, item_key
standardowe identyfikatory typu i instancji procesu workflow
- * activity
identyfikator procedury, której ma dotyczyć wymuszenie workflow (**WAŻNE !!: Wartość w pola Label na zakładce Node**)

przykładowo, activity bieżącego powiadomienia można sprawdzić tak:

```
select activity_label , notification_id
from WF_ITEM_ACTIVITY_STATUSES_v
where item_type = :item_type
and item_key = :item_key
and activity_end_date is null
and activity_status_code = 'NOTIFIED'
and activity_type_code = 'NOTICE'
```

- * comflag
 - SKIP - omiń klocek
 - RETRY - ponownie wykonaj klocek
 - RESET - nie wiem
- * result
powinno odpowiadać kodowi odpowiedzi, spodziewanej na wyjściu procedury/powiadomienia

Uwaga:

- w sporadycznych (i do końca nie wyjaśnionych) okolicznościach funkcja handleError nie zamyka powiadomienia, nie zwracając komunikatu o błędzie. Wówczas wykonaj wf_notification.cancel.

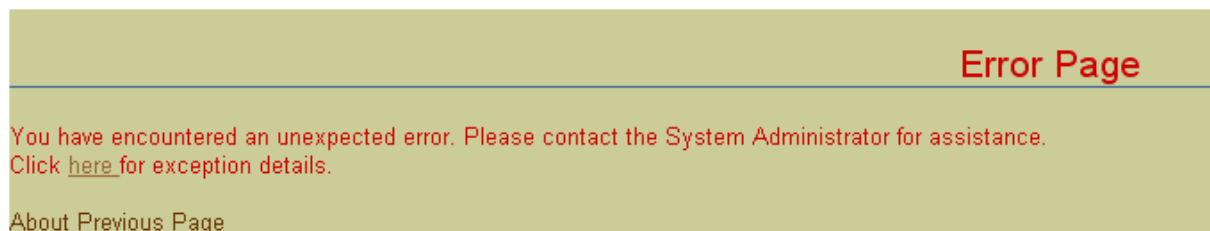
Przykład „gaszący” powiadomienia związane z błędem obsługi eventa (prawdopodobnie wyzwalanego zmianą nazwy kontrahenta):

```
begin
  for x in (
    select wias.item_type, wias.item_key, wpa.activity_name,
    notification_id
    from wf_item_activity_statuses wias, wf_process_activities wpa
    where wias.process_activity = wpa.instance_id --
    and wias.activity_status in ('NOTIFIED')
    and wpa.activity_name like 'EVENT_ERROR_NTF' and wias.item_type =
    'WFERROR')
  loop
    wf_notification.cancel(x.notification_id, 'jakiś komentarz')
    wf_engine.handleerror(x.item_type, x.item_key, 'EVENT_ERROR_NTF'
    ,wf_engine.eng_skip, 'ABORT');
  end loop;
end;
```

inne przykłady: polpharma / pakiet xxapworkflow_push_notifs

ERROR PAGE

Ustaw profile “%Diagnostyka%” na Tak. Wówczas dostaniesz bardziej szczegółowy komunikat o błędzie:



Dziennik zdarzeń / log

```
wf_log_pkg.string(6, 'Dummy', 'Child Process Created and current approver is ' || l_user);
```

Co to są zdarzenia / events

Zdarzenia służą do integracji z innymi systemami.

Zdarzenie posiada

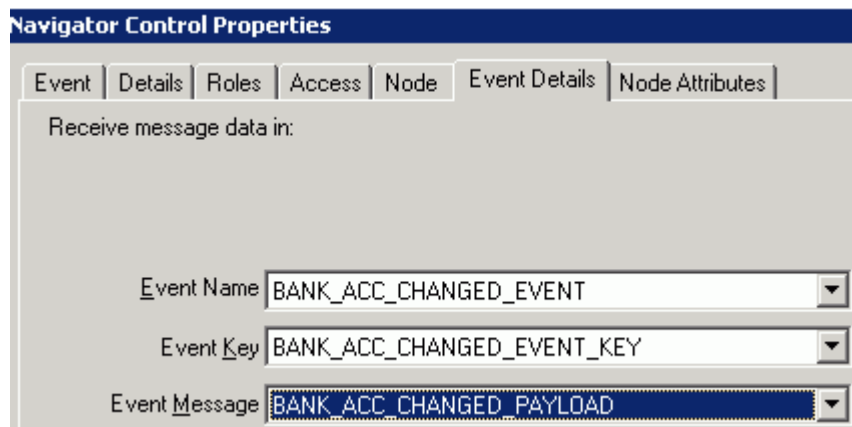
- ➡ (obligo) nazwę, klucz zdarzenia
- ➡ (opcjonalnie) zestaw parametrów, skojarzony plik XML.

Obsługa zdarzeń jest realizowana za pomocą advanced queuing, z każdym zdarzeniem związana jest nazwa systemowej kolejki (z reguły jest to wf_deferred)

Aby zobaczyć listę zdarzeń w systemie ich wartości użyj tego:
workflow.sql

Zob. też workflow.sql, workflow.ppt

Na poziomie eventa wskazuje się atrybuty workflow, do których przepisać event_key, event_name i event_Data (XML)



W celu poprawnego przetwarzania zdarzeń musi być aktywna usługa „Workflow Agent Listener%”

Uruchomienie zdarzenia

```
wf_event.raise(eventname,itemkey);  
np.  
eventname= 'oracle.apps.ap.event.invoice.approval'  
itemkey=TO_CHAR(l_invoice_id)||'_'||TO_CHAR(r_inv.approval_iteration)
```


Pamiętaj, że żeby zdarzenie mogło zostać przetworzone, należy je zarejestrować oraz określić subskrypcje.

Uruchomienie zdarzenia z plikiem XML / czytanie XML

źródło: <http://oracle.anilpassi.com/workflows-events-with-xml-payload-part-2.html>

```
create or replace trigger xxap_bank_accounts_bru1
BEFORE UPDATE
ON ap.ap_bank_accounts_all
FOR EACH ROW
DECLARE
    l_event_key number;
    l_event_data clob;
    l_event_name varchar2(250);
    l_text varchar2(2000);
    l_message varchar2(10);
BEGIN
    SELECT ap_payment_event_s.NEXTVAL INTO l_event_key FROM dual;
    l_event_name := 'xx.ap.bank.accountupdate';
    l_message := wf_event.test(l_event_name);
    dbms_lob.createtemporary(l_event_data
                           ,FALSE
                           ,dbms_lob.CALL);
    l_text := '<?xml version ="1.0" encoding ="ASCII"?>';
    dbms_lob.writeappend(l_event_data
                       ,length(l_text)
                       ,l_text);
    l_text := '<ap_bank_accounts>';
    dbms_lob.writeappend(l_event_data
                       ,length(l_text)
                       ,l_text);

    -----
    l_text := '<bank_branch_id>';
    l_text := l_text || fnd_number.number_to_canonical(:new.bank_branch_id);
    l_text := l_text || '</bank_branch_id>';
    dbms_lob.writeappend(l_event_data
                       ,length(l_text)
                       ,l_text);

    -----
    l_text := '<bank_account_number>';
    l_text := l_text || :new.bank_account_num;
    l_text := l_text || '</bank_account_number>';
    dbms_lob.writeappend(l_event_data
                       ,length(l_text)
                       ,l_text);

    -----
    l_text := '<changed_by_user_id>';
    l_text := l_text ||
               fnd_number.number_to_canonical(:new.last_updated_by);
    l_text := l_text || '</changed_by_user_id>';
    dbms_lob.writeappend(l_event_data
                       ,length(l_text)
                       ,l_text);

    -----
    l_text := '</ap_bank_accounts>';
    dbms_lob.writeappend(l_event_data
```

```

        ,length(l_text)
        ,l_text);

-- raise the event with the event with Bank Account Payload
wf_event.RAISE(p_event_name => l_event_name
               ,p_event_key  => l_event_key
               ,p_event_data => l_event_data);

END;
```

W workflow zawartość pliku możesz odczytać w następujący sposób

```

CREATE OR REPLACE PROCEDURE xx_get_bank_account_number(itemtype IN VARCHAR2
               ,itemkey  IN VARCHAR2
               ,actid    IN NUMBER
               ,funcmode IN VARCHAR2
               ,resultout OUT NOCOPY VARCHAR2) IS

    bank_event_document CLOB;
    event               wf_event_t;
    xmldoc              xmldom.domdocument;
    parser              xmlparser.parser;
    v_bank_account_number VARCHAR2(500);
BEGIN
    IF (funcmode = 'RUN')
    THEN
        --
        event := wf_engine.getitemattrevent(itemtype => itemtype
                                           ,itemkey  => itemkey
                                           ,NAME    => 'BANK_ACC_CHANGED_PAYLOAD');

        bank_event_document := event.geteventdata();
        v_bank_account_number := irc_xml_util.valueof(bank_event_document
                                                    ,'/ap_bank_accounts/bank_account_number');

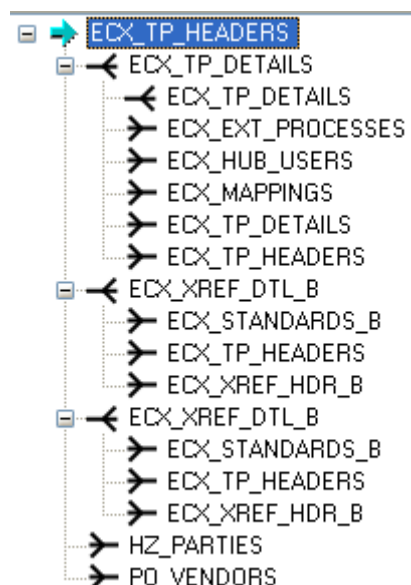
        wf_engine.setitemattrtext(itemtype => itemtype
                                ,itemkey  => itemkey
                                ,aname    => 'BANK_ACCT_NUMBER_FROM_PAYLOAD'
                                ,avalue  => v_bank_account_number);

    END IF;
END xx_get_bank_account_number;
```

Inne informacje

Zob. pakiet [hr_previous_employment_be1](#)

Zob. [ECX_STANDARD](#).GETXMLTP, użyte w workflow „Bramka XML-standard”



Zastępstwa / reguły czasu wolnego

Czynność	Zwykły użytkownik	Administrator procesów typu workflow
Dostęp do powiadomień innych użytkowników (Dostęp do skrzynki X dla Y)	Dostęp do skrzynki X dla Y musi zdefiniować X. Informacje: Zob. Dostęp do listy roboczej	n/a
Reguły czasu wolnego	Tak. Zob. Reguły czasu wolnego Informacje: Powiadomienia są automatycznie przekierowywane do określonych użytkowników	n/a
Przekierowywanie własnych powiadomień do innych użytkowników	Tak Informacje: Reguła czasu wolnego musi zostać zdefiniowana przed wysłaniem powiadomienia. Regułę może zdefiniować sam user lub administrator. Jeżeli nie została zdefiniowana, wówczas trzeba wykonać „ręcznie” przekierowanie. działa, tylko gdy nie ma #HIDE_REASSIGN	Regułę może zdefiniować sam user lub administrator
Przeglądanie i akceptowanie powiadomień wszystkich użytkowników	Nie	Tak Informacje: Administrator procesów typu workflow ->Administracja Jest zachowana informacja, że faktycznie czynność wykonał sysadmin
Przekierowywanie wszystkich powiadomień do innych użytkowników	Nie	Tak Informacje: , tylko gdy nie ma #HIDE_REASSIGN

Oba typy zastępstw są dostępne z poziomu okna powiadomień.

Wybór powiadomien:

☑ WSKAZÓWKA [Reguły czasu wolnego](#) - przekierowywanie lub automatyczne odpowiadanie na powiadomienia.

☑ WSKAZÓWKA [Dostęp do listy roboczej](#) - określanie, którzy użytkownicy mogą przeglądać powiadomienia i wykonywać związane z nimi czynności.

+ jeżeli message nie ma atrybutu **#HIDE_REASSIGN** to na powiadomieniu jest dostępny przycisk „przydziel ponownie”, który pozwala na przekierowanie powiadomienia do innej osoby. Jak sprawdzić, kto faktycznie wykonał czynność : Person_id – adresta powiadomienia. WF_NOTIFICATIONS.performer – kto faktycznie wykonał

Procesy podrzędne

Jeżeli chcesz utworzyć kilka równoległych procesów podrzędnych, to przeczytaj ten rozdział.

1. Uruchom procesy podrzędne za pomocą kodu pl-sql

Jak to zrobić: zob. Jak uruchomić / zatrzymać workflow z poziomu kodu. Pamiętaj o ustawienie procesu podrzędnego - `WF_ENGINE.SETITEMPARENT`

Właściwości: Navigator Control

Function | Details | Roles | Access | Node | Node Attributes

Item Type: >>> Obsługa umów sprzedaży [Edit]

Internal Name: US_CREATE_OPINION [New]

Display Name: Tworzenie procesu podrzednego - opiniowanie

Description:

Icon: START ICON [Browse]

Function Name: US_AGREEMENT_APPROVAL_2.CREATE_OPIN

Function Type: PL/SQL

Result Type: <None> [Edit]

Cost: 0.00

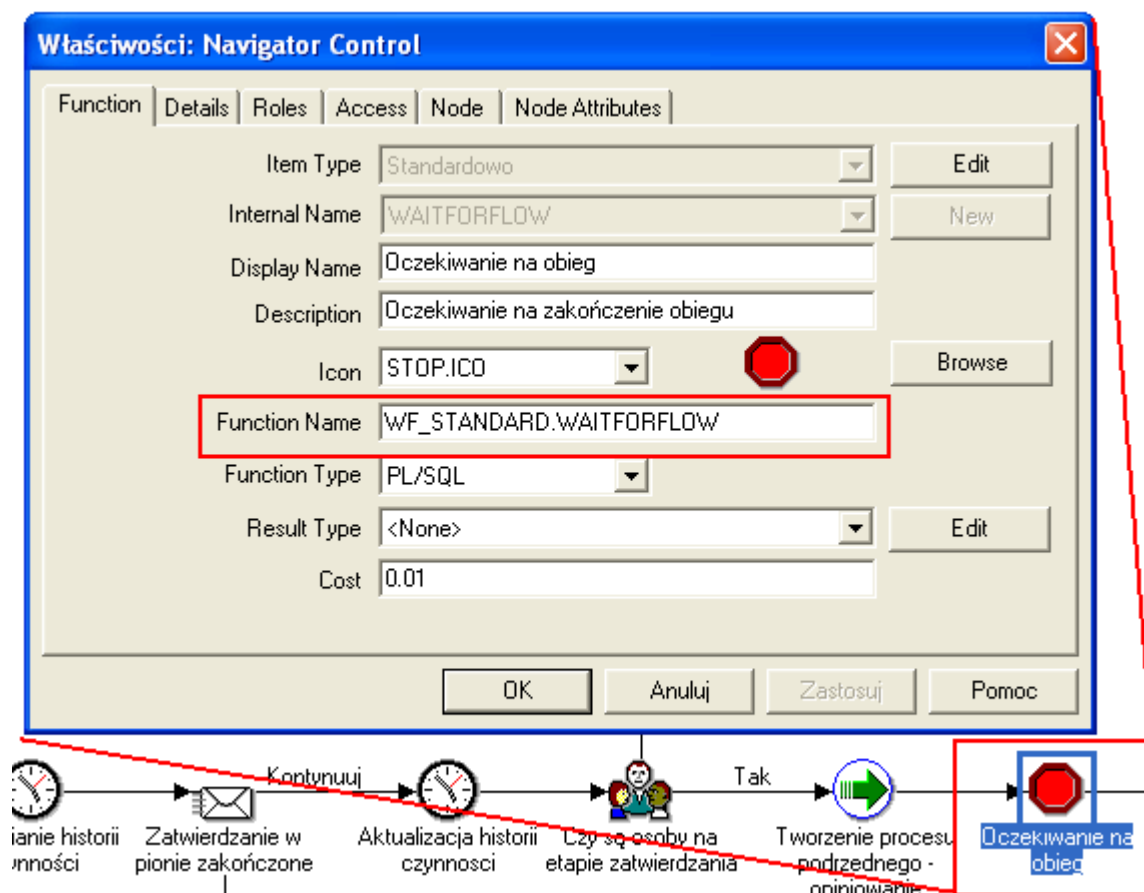
[OK] [Anuluj] [Zastosuj] [Pomoc]

Workflow Diagram:

zwrot umowy dw/osoby przygotowujacej → dW/stawianie historii czynności → Zatwierdzanie w pionie zakończone → Kontynuuj → Aktualizacja historii czynności → Czy są osoby na etapie zatwierdzania → Tak → Tworzenie procesu podrzednego - opiniowanie

Przerwij

2. Dodaj klocek wstrzymujący proces główny do czasu zakończenia podprocesów



To jeszcze nie wszystko:

1/ W procesie podrzędnym trzeba dodać „popychacz”, który zainicjuje sprawdzenie w poziomie procesu nadrzędnego, czy już można kontynuować.
 2/ trzeba wzajemnie połączyć popychacz i klocek czekający (label – atrybut). Nie jestem pewny, czy to jest niezbędne – do sprawdzenia.

3. Określ we właściwościach procesu nadrzędnego (klocek WAITFORFLOW) nazwę popychacza i że czekamy na procesy podrzędne

Właściwości: Navigator Control

Function | Details | Roles | Access | Node | **Node Attributes**

Attribute

Name

Name	Value Type	Value	Type	Description
Etykieta operacji ...	Constant	CONTINUEFLOW	Text	Etykieta operacji
Obieg kontynuujący	Constant	Podrzedne	Lookup	Lokalizacja operacji

OK Anuluj Zastosuj Pomoc

Właściwości: Navigator Control

Function | Details | Roles | Access | **Node** | Node Attributes

Label OCZEKIWANIE_NA_KONSULTACJE

Start/End Normal

Comment oczekiwanie na konsultacje merytoryczne

Timeout

Type No Timeout

Performer

Type Constant

Value <None> Edit

OK Anuluj Zastosuj Pomoc

3. Dodaj popychacz w procesie podrzędnym

Właściwości: Navigator Control

Function | Details | Roles | Access | Node | Node Attributes

Item Type: Standardowo [Edit]

Internal Name: CONTINUEFLOW [New]

Display Name: Kontynuacja obiegu

Description: Kontynuacja obiegu

Icon: START.ICO [Browse]

Function Name: WF_STANDARD.CONTINUEFLOW

Function Type: PL/SQL

Result Type: <None> [Edit]

Cost: 0.01

OK Anuluj Zastosuj Pomoc

Diagram flow: lizacja uchwała → Przydziel → Konsultacja merytoryczna → Zatwierdzenie → Aktualizacja historii czynności → Kontynuacja obiegu

Właściwości: Navigator Control

Function | Details | Roles | Access | Node | Node Attributes

Label: CONTINUEFLOW

Start/End: Normal

Comment:

Timeout Type: No Timeout

Performer Type: Constant

Value: <None> [Edit]

OK Anuluj Zastosuj Pomoc

Właściwości: Navigator Control

Function | Details | Roles | Access | Node | Node Attributes

Attribute

Name: Etykieta operacji czekającej

Type: Constant

Value: OCZEKIWANIE_NA_KONSULTACJE

Name	Value Type	Value	Type	Description
Etykieta operacji ...	Constant	OCZEKIWANIE...	Text	Etykieta opera
Obieg czekający	Constant	Nadrzędne	Lookup	Lokalizacja op

OK Anuluj Zastosuj Pomoc

Przykład: Moduł MUS, projekt PPL

Zagadnienia otwarte

przejrzyj przykłady workflow (c:\oracle\OraHome1\wf\data\US)