

日々はOracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2022年1月31日月曜日

プラグインのコンポーネント設定を参照する

Flows for APEXを使って経費精算のアプリケーションを作っている際に、SELECT文を以下のように書いていました。

```
select
  expe.expe_id
, expe.expe_justification
, expe.expe_amount
, expe.expe_status
, expe.expe_submitted_on
, expe.expe_invoice_dd
, expe.expe_purpose
, inst.prcs_id
, tibx.sbfl_id
from expe_expenses expe
  join flow_instances_vw inst on expe.expe_id = inst.prcs_business_ref
 left join flow_task_inbox_vw tibx on expe.expe_id = tibx.sbfl_business_ref
where inst.dgrm_name = '経費精算'
      and (tibx.sbfl_dgrm_name = '経費精算' or tibx.sbfl_dgrm_name is null)
      and expe.expe_submitted_by = :APP_USER
```

SELECT文にフロー・モデル名が直書きされています。フロー・モデル名をバインド変数に置き換える方法について考えてみます。フロー・モデル名を**G_FLOW_NAME**というバインド変数に置き換え、結果として以下のSELECT文に置き換えます。

```
select
  expe.expe_id
, expe.expe_justification
, expe.expe_amount
, expe.expe_status
, expe.expe_submitted_on
, expe.expe_invoice_dd
, expe.expe_purpose
, inst.prcs_id
, tibx.sbfl_id
from expe_expenses expe
  join flow_instances_vw inst on expe.expe_id = inst.prcs_business_ref
 left join flow_task_inbox_vw tibx on expe.expe_id = tibx.sbfl_business_ref
where inst.dgrm_name = :G_FLOW_NAME
      and (tibx.sbfl_dgrm_name = :G_FLOW_NAME or tibx.sbfl_dgrm_name is null)
      and expe.expe_submitted_by = :APP_USER
```

経費精算のアプリケーションが使用するフロー・モデルは、プラグインの**コンポーネント設定のGlobal Flow**として定義されています。



バインド変数として使用するG_FLOW_NAMEは、アプリケーション・アイテムとして定義します。

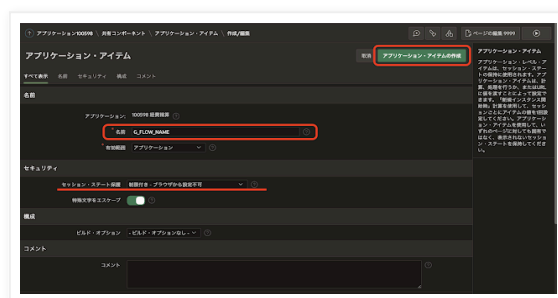
共有コンポーネントのアプリケーション・アイテムを開きます。



作成をクリックし、アプリケーション・アイテムG_FLOW_NAMEの作成を始めます。



名前として、G_FLOW_NAMEを入力します。セキュリティのセッション・ステート保護はデフォルトの制限付き - ブラウザから設定不可のままにします。一番、厳しい制限になります。以上の設定で、アプリケーション・アイテムの作成をクリックします。



アプリケーション・アイテムG_FLOW_NAMEが作成されます。

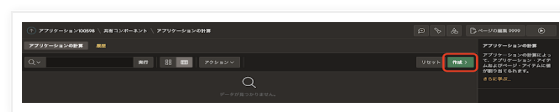


作成したアプリケーション・アイテムG_FLOW_NAMEに、フロー・モデル名を設定します。アプリケーション内でG_FLOW_NAMEを参照するときはいつでも、コンポーネント設定のGlobal Flowの値を返すようにします。

共有コンポーネントのアプリケーションの計算を開きます。



作成をクリックします。



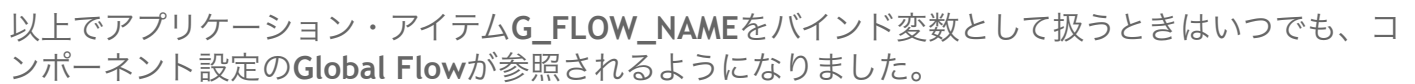
計算アイテムとしてG_FLOW_NAMEを選択します。頻度の計算ポイントとして新規インスタンス(新規セッション)開始時を選択します。

この頻度であれば、コンポーネント設定のGlobal Flowが変更されると、経費精算のアプリケーションは一旦サインアウトしてセッションを終了したのち、サインインをやり直して新規セッションを開始して初めて、変更されたGlobal FlowがG_FLOW_NAMEに反映されます。実際にはGlobal Flowの変更はほとんど行わないので、一番頻度が少なく（つまり実行回数が少ない）データベースに負荷がかからない設定を選択します。

計算の計算タイプにファンクション本体を選択します。言語はPL/SQLとして、以下のコードを計算に記述します。

```
declare
  l_pos integer;
  l_flow_name APEX_APPL_PLUGIN_SETTINGS.ATTRIBUTE_01%type;
begin
  select attribute_01 into l_flow_name
  from APEX_APPL_PLUGIN_SETTINGS
  where application_id = :APP_ID
    and plugin_code = 'PLUGIN_COM.FLOWS4APEX.MANAGE_INSTANCE.PROCESS';
  l_pos := instr(l_flow_name, ',');
  if l_pos > 0 then
    -- カンマがあれば、それに続くバージョンを取り除く。
    l_flow_name := substr(l_flow_name, 1, (l_pos - 1));
  end if;
  return l_flow_name;
exception
  when no_data_found then
    /*
     * Global Flowが設定されていないので、アプリケーション・アイテムは設定しない。
     */
    return null;
end;
```

以上の設定を行い、**計算の作成**を実行します。



経費精算

経費精算

セッションを開く

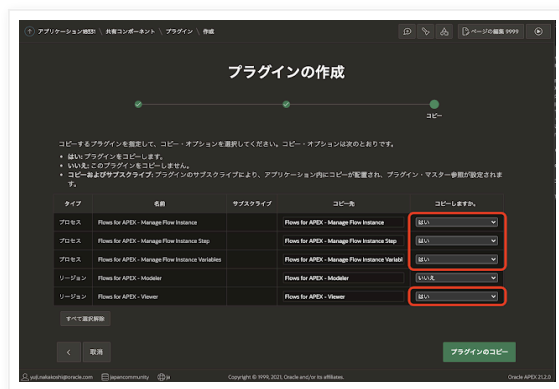
[illegible]

バインド変数の設定は以上になります。

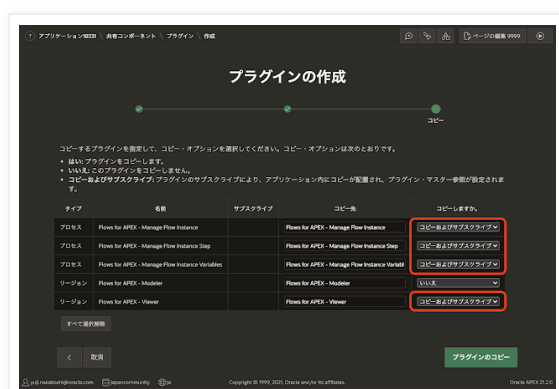
今回の作業中で気がついた点を追記しておきます。

プラグインを他のアプリケーションからコピーして作成する場合、**コピーしますか？**の指定で、**はい**（コピーのみ）と**コピーおよびサブスクライブ**を選ぶことができます。

コピーしますか？には**はい**を選択した画面です。

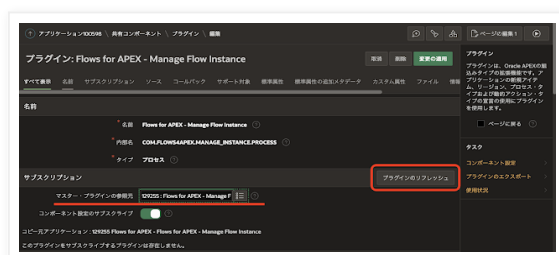


コピーおよびサブスクライブを選択した画面です。



コピーおよびサブスクライブを選択した場合、単にプラグインがコピーされるだけではなく、コピー元のプラグインに紐づけられた状態になります。

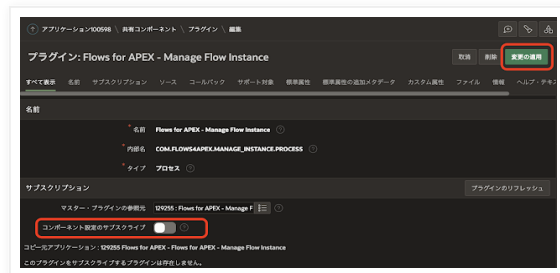
プラグインの設定画面の**サブスクリプションのマスター・プラグインの参照元**として、コピーと元のプラグインが設定されます。コピー元のプラグインに変更があった場合、**プラグインのリフレッシュ**を実行することにより、その変更をコピーとして作成されている、このプラグインに反映することができます。



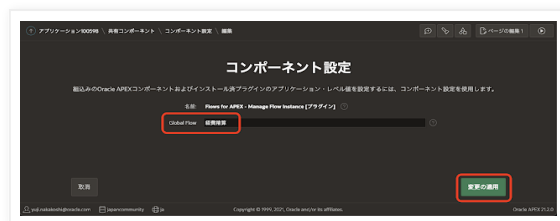
コピーおよびサブスクライブを指定してプラグインを作成した場合、デフォルトでは**コンポーネント設定**の画面で**変更の適用**のボタンが表示されません。



サブスクライブしているプラグインで個別にコンポーネント設定を行うには、**コンポーネント設定のサブスクライブをOFF**にします。



コンポーネント設定のサブスクライブをOFFにすると、サブスクライブしているプラグインごとにコンポーネント設定を行うことができます。



今回の記事は以上です。

Oracle APEXのアプリケーション作成の参考になれば幸いです。

完

Yuji N. 時刻: 13:29

共有

<

ホーム

>

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。こちらの記事につきましては、免責事項の参照をお願いいたします。

詳細プロフィールを表示

