

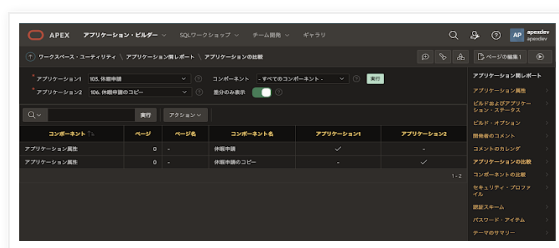
日々是Oracle APEX

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。

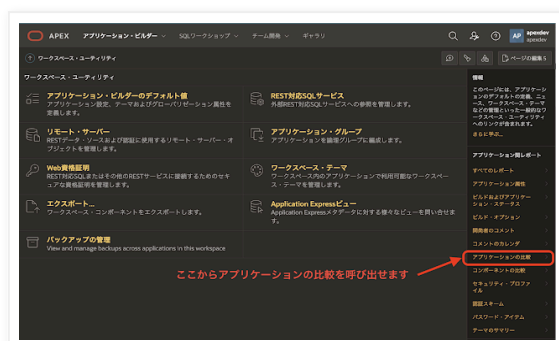
2022年1月14日 金曜日

APEXアプリケーションを比較している実装を確認する

Oracle APEXのアプリケーション・ビルダーは、**アプリケーションの比較**という機能を提供しています。



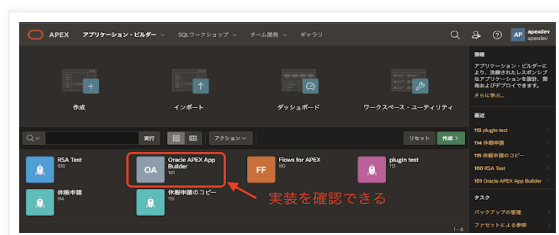
この機能は**ワークスペース・ユーティリティ**を開いた画面の、右ペインのリストから呼び出すことができます。見つけにくい所にあり、存在を忘れがちな機能ではあります。



一体どういう実装でアプリケーションを比較しているのかを確認してみました。

以下に確認手順を記載します。

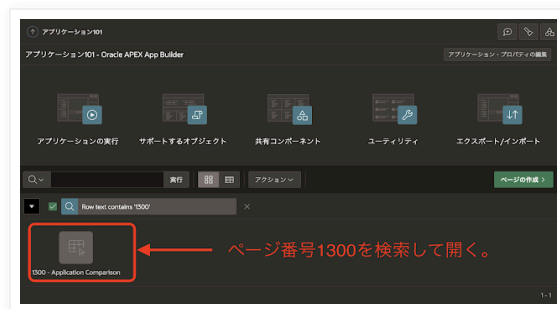
最初に[こちらの記事](#)にある手順にそって、アプリケーション・ビルダーの実装となる**f4000.sql**をワークスペースにインポートします。



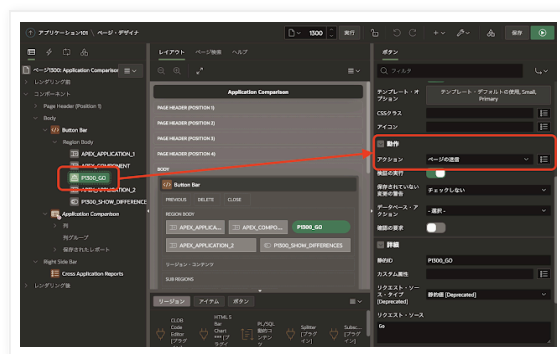
アプリケーションの比較をするページの**ページ番号**を確認します。APEX 21.2では**1300**でした。



アプリケーション・ビルダーのページ番号1300を検索し、ページ・デザイナーで開きます。

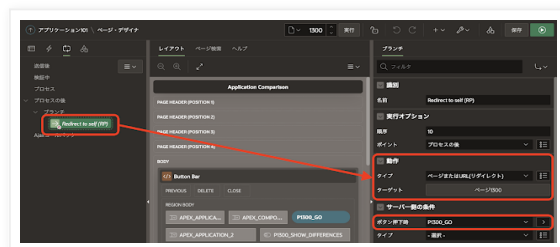


実行ボタンはP1300_GO、動作のアクションがページの送信であることが分かります。そのため、ボタンの処理はプロセスとして実装されています。



プロセス・ビューを開き、ボタンP1300_GOが押されたときに実行されるプロセスを確認します。サーバー側の条件のボタン押下時が、P1300_GOとなっているプロセスRedirect to self (RP)が見つかります。動作のタイプとして、ページまたはURL(リダイレクト)が定義されています。

リダイレクト先のページ番号は1300なので、ページの再表示になります。

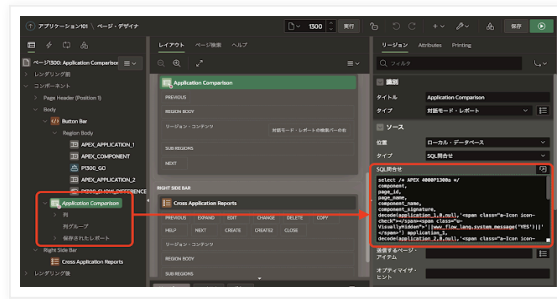


レンダリング・ビューを確認するとApplication Comparisonという対話モード・レポートのリージョンが確認できます。以下のサーバー側の条件が設定されていて、比較するアプリケーションが設定されているときのみ、レポートが表示されるようになっています。

```

nvl(v('REQUEST'),'X') != 'NORUN' and nvl(:APEX_APPLICATION_1,'0') != '0' and
nvl(:APEX_APPLICATION_2,'0') != '0'

```



レポートのソースとして、以下のSQL問合せが使用されています。

```
select /* APEX 4000P1300a */
component,
page_id,
page_name,
component_name,
component_signature,
decode(application_1,0,null,'<span class="a-Icon icon-check"></span><span class="u-VisuallyHidden"></span>') a_1,
decode(application_2,0,null,'<span class="a-Icon icon-check"></span><span class="u-VisuallyHidden"></span>') a_2
from (
select page_id,
page_name,
component,
component_name,
component_signature,
application_1,
application_2
from apex_component_signatures) q
where
(nvl(:P1300_SHOW_DIFFERENCES,'Y')='N' or
(nvl(:P1300_SHOW_DIFFERENCES,'Y')='Y' and application_1 != application_2))
```

apex_app_comparison.sql hosted with ❤ by GitHub

[view raw](#)

ビューAPEX_COMPONENT_SIGNATURESの検索が実質的な処理であることが分かります。

アプリケーション・ビルダーは、APEXがインストールされているスキーマをデフォルトの解析スキーマとします。APEX_COMPONENT_SIGNATURESにはスキーマが指定されていないので（今回はAPEX 21.2で確認しているため、）APEX_210200がAPEX_COMPONENT_SIGNATURESの所有者です。

名前がAPEX_で始まっているので、これがビューであるのは間違いありませんが、一応、APEX_COMPONENT_SIGNATURESのオブジェクト・タイプを確認します。

DBA権限を持ったユーザーで実行します。

```
SQL> select object_name, object_type from dba_objects where owner = 'APEX_210200'
and object_name = 'APEX_COMPONENT_SIGNATURES';
```

OBJECT_NAME	OBJECT_TYPE
-----	-----
APEX_COMPONENT_SIGNATURES	VIEW

```
SQL>
```

ビューの定義を確認します。

```
SQL> set long 30000
SQL> set pages 1000 lines 1000 trims on trimo on
SQL> col text format a10000
SQL> set heading off
SQL> select text from dba_views where owner = 'APEX_210200' and view_name =
'APEX_COMPONENT_SIGNATURES';

select page_id,
       page_name,
       component,
       component_name,
       component_signature,
       sum(decode(application_id,x.a1,1,0)) as application_1,
       sum(decode(application_id,x.a2,1,0)) as application_2
  from
    (select nv('APEX_APPLICATION_1') a1, nv('APEX_APPLICATION_2') a2,
v('APEX_COMPONENT') c from sys.dual) x,
    (
      select 'PAGE' component,
             page_id,
             page_name,
             lpad(page_id,5,'00000')||'. '||page_title component_name,
             component_signature,
             application_id
        from APEX_APPLICATION_PAGES,
             (select nv('APEX_APPLICATION_1') a1, nv('APEX_APPLICATION_2') a2,
v('APEX_COMPONENT') c from sys.dual) x
       where application_id in (x.a1,x.a2)
             and (nvl(x.c,'0')='0' or x.c='PAGE')
      union all
      select 'PAGE.VALIDATION' component,
             page_id,
             page_name,
             VALIDATION_NAME component_name,
             component_signature,
             application_id
        from APEX_APPLICATION_PAGE_VAL,
             (select nv('APEX_APPLICATION_1') a1, nv('APEX_APPLICATION_2') a2,
v('APEX_COMPONENT') c from sys.dual) x
       where application_id in (x.a1,x.a2)
             and (nvl(x.c,'0')='0' or x.c='PAGE.VALIDATION')
      union all
      select 'PAGE.BRANCH' component,
             page_id,
             page_name,
             nvl(branch_name, substr(BRANCH_ACTION,1,40)||length(branch_action))
component_name,
             component_signature,
             application_id
        from APEX_APPLICATION_PAGE_BRANCHES,
             (select nv('APEX_APPLICATION_1') a1, nv('APEX_APPLICATION_2') a2,
v('APEX_COMPONENT') c from sys.dual) x
       where application_id in (x.a1,x.a2)
```

[中略]

```
        from apex_appl_plugins,
            (select nv('APEX_APPLICATION_1') a1, nv('APEX_APPLICATION_2') a2,
v('APEX_COMPONENT') c from sys.dual) x
        where application_id in (x.a1, x.a2)
            and (nvl(x.c,'0')='0' or x.c='PLUGIN')
        union all
        select 'TEXT.MESSAGE' component,
            0 as page_id,
            null as page_name,
            translatable_message as component_name,
            component_signature,
            application_id
        from apex_application_translations,
            (select nv('APEX_APPLICATION_1') a1, nv('APEX_APPLICATION_2') a2,
v('APEX_COMPONENT') c from sys.dual) x
        where application_id in (x.a1, x.a2)
            and (nvl(x.c,'0')='0' or x.c='TEXT.MESSAGE')
    ) d
group by page_id, page_name, component, component_signature, component_name
```

SQL>

ビューの定義をファイルに落として内容を確認すると、それぞれのコンポーネントごとに列
COMPONENT_SIGNATURESを比較してunion allしていることが分かります。

```
$ grep select view.sql | grep component
select 'PAGE' component,
select 'PAGE.VALIDATION' component,
select 'PAGE.BRANCH' component,
select 'PAGE.COMPUTATION' component,
select 'PAGE.PROCESS' component,
select 'PAGE.REGION' component,
select 'PAGE.ITEM' component,
select 'PAGE.BUTTON' component,
select 'AUTHENTICATION' component,
select 'LOV.ENTRY' component,
select 'LOV' component,
select 'APPLICATION.PROCESS' component,
select 'APPLICATION.COMPUTATION' component,
select 'APPLICATION.SETTING' component,
select 'NAVBAR' component,
select 'TAB' component,
select 'PARENT.TAB' component,
select 'BREADCRUMB' component,
select 'LIST' component,
select 'LIST.ENTRY' component,
select 'TREE' component,
select 'SHORTCUT' component,
select 'APPLICATION.ITEM' component,
select 'BUILD.OPTION' component,
select 'BREADCRUMB.ENTRY' component,
select 'APPLICATION.ATTRIBUTES' component,
select 'PAGE.REGION' component,
select 'REPORT.COLUMN' component,
select 'REPORT.COLUMN' component,
select 'AUTHORIZATION' component,
select 'ACL.ROLE' component,
select 'THEME' component,
select 'TEMPLATE.CALENDAR' component,
select 'TEMPLATE.BUTTON' component,
```

```
select 'TEMPLATE.POPUP' component,  
select 'TEMPLATE.REPORT' component,  
select 'TEMPLATE.BREADCRUMB' component,  
select 'TEMPLATE.LIST' component,  
select 'TEMPLATE.LABEL' component,  
select 'TEMPLATE.REGION' component,  
select 'TEMPLATE.PAGE' component,  
select 'WEB.SERVICE' component,  
select 'DATA.LOADING' component,  
select 'PLUGIN' component,  
select 'TEXT.MESSAGE' component,  
$
```

ビューAPEX_COMPONENT_SIGNATURES自体もですが、ビュー定義にWWW_FLOW_STEPSといった保護されているオブジェクトが含まれているため、一般のユーザーからはビューの検索やこのSQLの実行はできません。オンプレの環境であれば権限を与えればアクセスできますが、Autonomous DatabaseのAPEXの場合は不可です。

実装を確認できるだけでも、同様の機能を検討しているときには、有益な情報になるでしょう。

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: 12:44

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](#)

Powered by Blogger.