

日日是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。

2021年3月2日火曜日

エディションベース再定義をAPEXで使用する - プロシージャ編

11gの頃から提供されている機能にエディション・ベースの再定義(Edition-based Redefinition)というのがあります。難しそうなので、今まで触ったことさえなかったのですが、ちょっと試してみることにしました。

食わず嫌いもよくないですし、アプリケーションのアップグレードを無停止で行うには、この機能を活用する必要があります。

この機能については初心者なので、最初は以下のマニュアルに記載のある例をOracle APEXのアプリケーションで実践してみます。

データベース開発者ガイド Release 19

27 エディションベースの再定義の使用

アプリケーションをアップグレードするためのEBRの使用

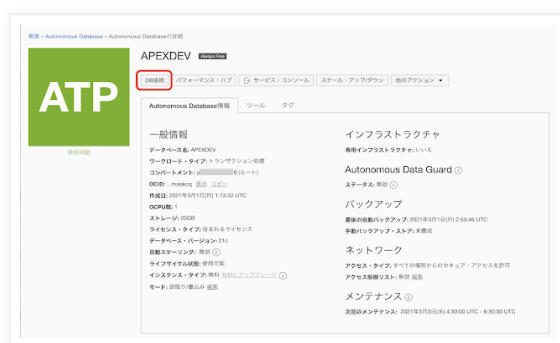
エディションのみを使用するEBRの手順

確認には、Always FreeのAutonomous Transaction Processingを使用しました。またエディションを指定した状態でプロシージャを作成するために、コマンドラインのインターフェースとしてSQLclを使用しました。

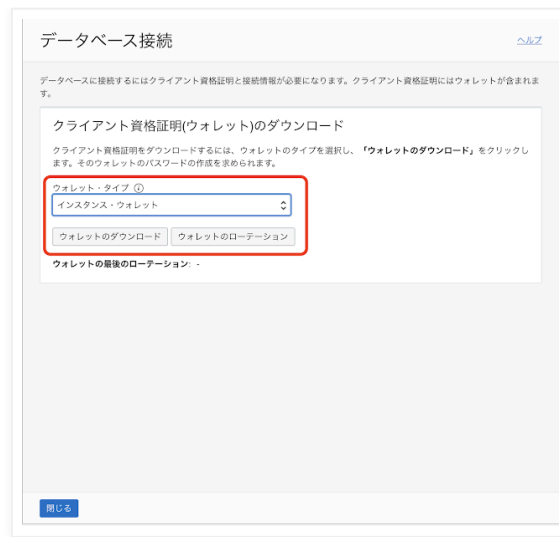
APEXのワークスペース・ユーザー(スキーマ)をAPEXDEVとして作成しています。Autonomous Databaseですので、管理者ユーザーはADMINです。

0. 準備

Autonomous Databaseのコンソールからウォレットをダウンロードします。**DB接続**をクリックします。



ウォレットをダウンロードします。ウォレット・タイプはインスタンス・ウォレットとリージョナル・ウォレットのどちらでも構いません。



Wallet_インスタンス名.zipというファイルがダウンロードされます。

今回はSQLclを使います。SQダウンロードは[こちら](#)から行います。

ダウンロードしたウォレットとSQLclを使ってAutonomous Databaseに接続するには、以下のcloudconfigオプションでウォレットを指定してSQLclを呼び出します。

```
% sql -cloudconfig Wallet_インスタンス名.zip ユーザー名@サービス名
```

1. ワークスペース・スキーマをエディション対応にする。

ユーザーADMINで接続し、スキーマAPEXDEVをエディション対応にします。

```
SQL> alter user apexdev enable editions;
```

User APEXDEVが変更されました。

```
SQL>
```

2. プロシージャhelloを作成する。

注意：少なくとも一回はワークスペースにログインしてください。ワークスペース作成直後ではSQLclから接続できませんでした。また、APEXの管理画面からワークスペースと同時にデータベース・スキーマを作成する際に、パスワードの二重入力は求められず、また、伏字になるため、意図した通りにパスワードが入力されているか確認する方法がありません。ですので、ワークスペースにログインすることで、パスワードが意図した通り設定されていることを、あらかじめ確認しておきます。

ユーザーAPEXDEV(APEXのワークスペース・ユーザー)で接続し、プロシージャhelloを作成します。APEXで動作させる前提で、dbms_output.put_lineではなく、http.pで出力するプロシージャとして登録します。

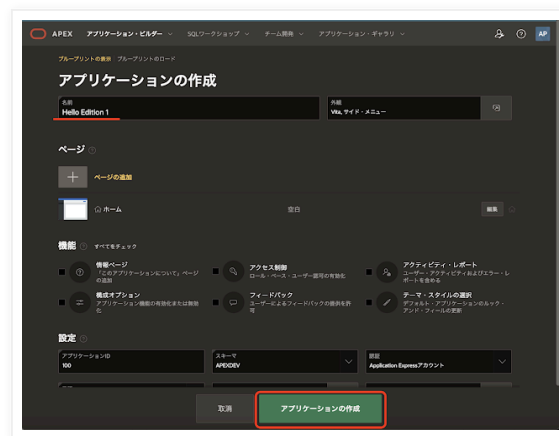
```
SQL> create or replace procedure hello is
2   begin
3       http.p('Hello, edition 1.');
```

Procedure HELL0がコンパイルされました

SQL>

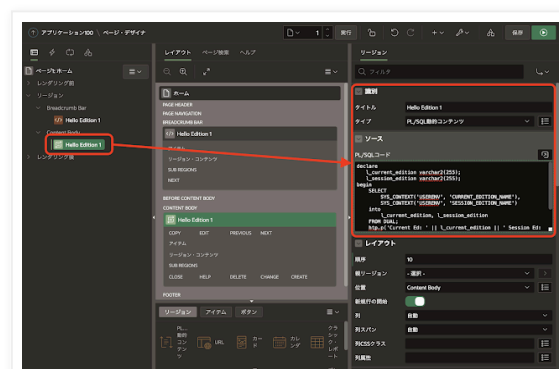
3. 作成したhelloを呼び出すAPEXアプリケーションを作成する。

名前をHello Edition 1として、空のアプリケーションを作成します。

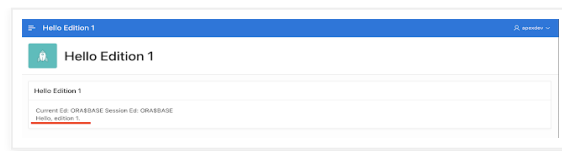


アプリケーションが作成できたら、ホーム・ページにタイプがPL/SQL動的コンテンツのリージョンを作成します。タイトルはHello Edition 1、ソースのPL/SQLコードとして、ページ処理中のエディションを確認するためのコードも含めた、以下のコードを記述します。

```
declare
  l_current_edition varchar2(255);
  l_session_edition varchar2(255);
begin
  SELECT
    SYS_CONTEXT('USERENV', 'CURRENT_EDITION_NAME'),
    SYS_CONTEXT('USERENV', 'SESSION_EDITION_NAME')
  into
    l_current_edition, l_session_edition
  FROM DUAL;
  http.p('Current Ed: ' || l_current_edition || ' Session Ed: ' || l_session_edition);
  http.nl;
  hello;
end;
```



アプリケーションを実行し、プロシージャhelloによって印刷された文字列を確認します。



Hello, edition 1.と表示されていることが確認できます。また、このデータベース・セッションのエディションがORA\$BASEで、このリージョンに記述されているPL/SQLコードがエディションORA\$BASEで評価されていることも確認できます。

4. エディションE2を作成する。

ユーザーADMINで接続し、エディションE2を作成します。

```
SQL> create edition e2 as child of ora$base;
```

Edition E2は作成されました。

```
SQL>
```

作成したエディションE2をユーザーAPEXDEVで利用可能にします。

```
SQL> grant use on edition e2 to apexdev;
```

Grantが正常に実行されました。

```
SQL>
```

5. エディションE2でのプロシージャhelloを作成する。

ユーザーAPEXDEVで接続し、プロシージャhelloを更新します。最初にセッションにエディションE2を設定します。その後、プロシージャhelloを作り直します。

```
SQL> alter session set edition = e2;
```

Sessionが変更されました。

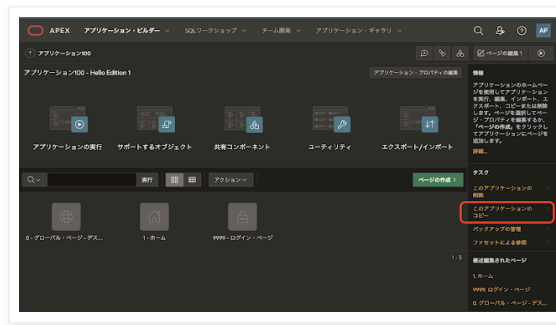
```
SQL> create or replace procedure hello is
  2   begin
  3       http.p('Hello, edition 2. ');
  4   end;
  5   /
```

Procedure HELLOがコンパイルされました

```
SQL>
```

6. エディションE2で動作するアプリケーションを作成する。

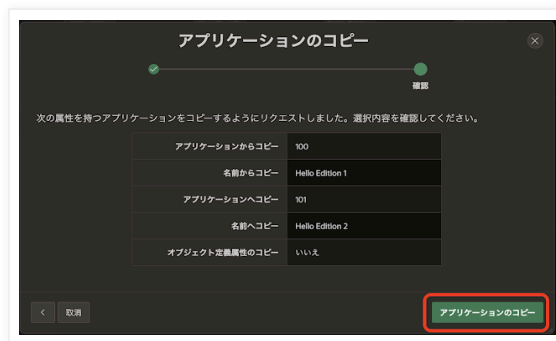
先ほど作成したアプリケーションのコピーします。



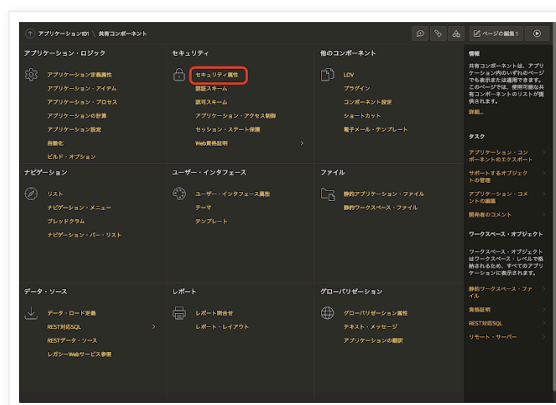
新規アプリケーション名はHello Editin 2とします。



アプリケーションのコピーを実行します。



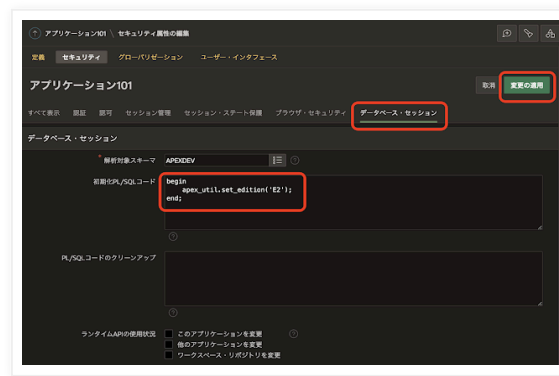
アプリケーションがコピーされたら、共有コンポーネントのセキュリティ属性を開きます。



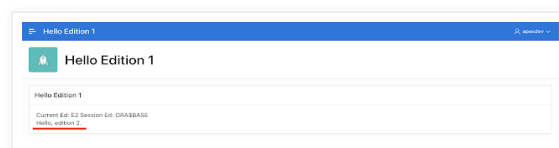
データベース・セッションの初期化PL/SQLコードとして、**APEX_UTIL.SET_EDITION**を呼び出し、エディションE2を設定します。

```
begin
  apex_util.set_edition('E2');
end;
```

変更の適用を行います。

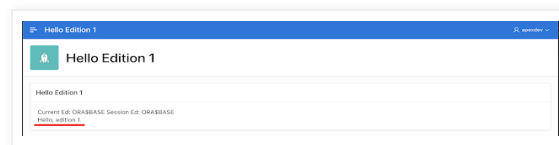


アプリケーションを実行し、結果を確認します。



エディションE2として定義したプロシージャhelloの結果、**Hello, edition 2.**が印刷されています。

コピー元のアプリケーションを実行し、表示を確認します。



紐づいているエディションに従って、呼び出されるプロシージャが異なっていることが確認できます。

この状態で、旧バージョンのアプリケーションと新バージョンのアプリケーションが同時に動作しています。この後のアプリケーションの切り替え作業は、Oracle APEXの機能を活用して実施することになります。[以前に書いた記事](#)が参考になります。

7. データベースにデフォルトで適用されるエディションを設定する。

ユーザーADMINで接続し、デフォルトで使われるエディションを設定します。

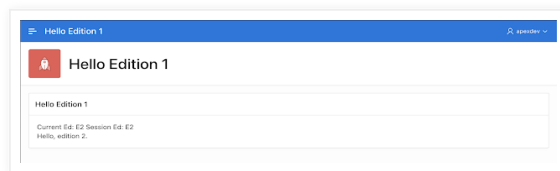
```
SQL> alter database default edition = e2;
```

Databaseが変更されました。

```
SQL>
```

データベースを再起動します。

再度、一番最初に作成したアプリケーション(APEX_UTIL.SET_EDITIONを行っていないアプリケーション)を実行します。



セッション・エディション、カレント・エディションともに**E2**になり、プロシージャhelloの出力結果も**Hello, edition 2.**になっていることが確認できます。

この状態で、エディションORA\$BASEは不要なエディションになります。本来、エディションのリタイアと削除が可能なのですが、ORA\$BASEについては以下のユーザーまたはロールに割り当てられているUSE権限を管理者ユーザーADMINではREVOKEできないため、削除はできません。

```
SQL> select grantee, privilege
       2  from dba_tab_privs
       3  where table_name = 'ORA$BASE' and type = 'EDITION';
```

GRANTEE	PRIVILEGE
PDB_DBA	USE
SYSTEM	USE
ADMIN	USE
DATAPUMP_CLOUD_EXP	USE
DATAPUMP_CLOUD_IMP	USE
GGADMIN	USE

6行が選択されました。

```
SQL>
```

REVOKEしようすると、ORA-1927が発生します。

```
SQL> revoke use on edition ora$base from pdb_dba;
```

次のコマンドの開始中にエラーが発生しました : 行 1 -

```
revoke use on edition ora$base from pdb_dba
```

エラー・レポート -

```
ORA-01927: cannot REVOKE privileges you did not grant
01927. 00000 - "cannot REVOKE privileges you did not grant"
*Cause:      You can only revoke privileges you granted.
*Action:     Don't revoke these privileges.
SQL>
```

Autonomous DatabaseではデフォルトのエディションであるORA\$BASEは削除できない、という前提でエディションを作りましょう。

8. 作成したオブジェクトの確認方法など。

列EDITION_NAMEを参照すると、どのエディションのオブジェクトが使われているかを確認できます。

```
SQL> select owner, object_type, object_name, edition_name from
all_objects where object_name = 'HELLO';
```

OWNER	OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
APEXDEV	PROCEDURE	HELLO	E2

```
SQL>
```

エディションごとのプロシージャの作成状況は、末尾に*_AEがついているビューを参照します。

```
SQL> select owner, object_type, object_name, edition_name from
all_objects_ae where object_name = 'HELLO';
```

OWNER	OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
APEXDEV	PROCEDURE	HELLO	ORA\$BASE
APEXDEV	PROCEDURE	HELLO	E2

```
SQL>
```

簡単にプロシージャのライフサイクルを確認してみます。

最初に準備として、デフォルトのエディションをORA\$BASEに戻し、データベースを再起動しておきます。

ユーザーAPEXDEVで接続します。接続した状態での、セッション・エディションを確認します。

```
SQL> show edition
EDITION
-----
ORA$BASE
SQL>
```

Oracle APEXでは、アプリケーションに埋め込まれたSQLはDBMS_SQLパッケージを使って実行されます。その際にAPEX_UTIL.SET_EDITIONで設定されたエディションがカレント・エディションとして適用されます。APEX_UTIL.SET_EDITIONにてカレント・エディションが設定されていない場合にのみ、セッション・エディションの元でSQLが評価されます。

SQLclなどの場合はセッション・エディションの元でSQLが評価されます。

プロシージャGREETINGを作成します。Hello, edition 1.を印刷します。

```
SQL> create procedure greeting is
2   begin
3       dbms_output.put_line('Hello, edition 1.');
```

実行します。

```
SQL> set serveroutput on
SQL> exec greeting;
```


Hello, edition 1.

PL/SQLプロシージャが正常に完了しました。

SQL>

Hello, edition 1.が印刷されます。USER_OBJECTSビューを確認します。

```
SQL> select object_type, object_name, edition_name from user_objects
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
PROCEDURE	GREETING	ORA\$BASE

SQL>

作成したプロシージャGREETINGは、エディションORA\$BASEのプロシージャを参照しています。

セッション・エディションをE2にしてプロシージャGREETINGを実行します。

```
SQL> alter session set edition = e2;
```

Sessionが変更されました。

```
SQL> exec greeting;
Hello, edition 1.
```

PL/SQLプロシージャが正常に完了しました。

SQL>

Hello, edition 1. が印刷されます。

先ほどと同様にプロシージャGREETINGのEDITION_NAMEを確認します。

```
SQL> select object_type, object_name, edition_name from user_objects
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
PROCEDURE	GREETING	ORA\$BASE

SQL>

セッション・エディションはE2ですが、プロシージャGREETINGのエディションはORA\$BASEです。プロシージャに対する操作をしていなければ、親となるエディションのプロシージャが継承されます。

エディションE2でプロシージャGREETINGを変更します。

```
SQL> create or replace procedure greeting is
2 begin
```

```
3      dbms_output.put_line('Hello, edition 2.');
```

```
4  end;
```

```
5  /
```

Procedure GREETINGがコンパイルされました

SQL>

変更したプロシージャGREETINGを実行します。

```
SQL> exec greeting;
```

```
Hello, edition 2.
```

PL/SQLプロシージャが正常に完了しました。

SQL>

Hello, edition 2.が印刷されます。先ほどと同様にUSER_OBJECTSのEDITION_NAMEを確認します。

```
SQL> select object_type, object_name, edition_name from user_objects
```

```
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
PROCEDURE	GREETING	E2

SQL>

EDITION_NAMEはE2で、エディションE2の元で改変したプロシージャGREETINGが参照されていることがわかります。

セッション・エディションをORA\$BASEに戻して、プロシージャGREETINGを実行してみます。

```
SQL> alter session set edition = ora$base;
```

Sessionが変更されました。

```
SQL> exec greeting;
```

```
Hello, edition 1.
```

PL/SQLプロシージャが正常に完了しました。

SQL>

Hello, edition 1.が印刷されます。USER_OBJECTSビューも確認してみます。

```
SQL> select object_type, object_name, edition_name from user_objects
```

```
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
-------------	-------------	--------------

PROCEDURE	GREETING	ORA\$BASE
-----------	----------	-----------

SQL>

セッション・エディションORA\$BASEからは、ORA\$BASEで登録されているプロシージャGREETINGが参照されていることが分かります。

セッション・エディションをE2に変更し、プロシージャGREETINGを削除します。

```
SQL> alter session set edition = e2;
```

Sessionが変更されました。

```
SQL> drop procedure greeting;
```

Procedure GREETINGが削除されました。

SQL>

プロシージャGREETINGを実行します。プロシージャが存在しないため、エラーが発生します。親となるエディションのプロシージャは呼ばれません。

```
SQL> exec greeting;
```

次のコマンドの開始中にエラーが発生しました : 行 1 -

```
BEGIN greeting; END;
```

エラー・レポート -

```
ORA-06550: line 1, column 7:
PLS-00201: identifier 'GREETING' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
06550. 00000 - "line %s, column %s:\n%s"
*Cause:      Usually a PL/SQL compilation error.
*Action:
SQL>
```

USER_OBJECTSビューを見ても、エントリは無くなっています。

```
SQL> select object_type, object_name, edition_name from user_objects
where object_name = 'GREETING';
```

行が選択されていません

SQL>

USER_OBJECTS_AEビューを確認します。エディションE2でのプロシージャGREETINGは存在しないのではなく、削除されている(NON-EXISTENT)という扱いになっています。

```
SQL> select object_type, object_name, edition_name from user_objects_ae
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
-------------	-------------	--------------

PROCEDURE	GREETING	ORA\$BASE
NON-EXISTENT	GREETING	E2

SQL>

そのため、親となるエディションのプロシージャが呼び出されることはありません。

作成されたプロシージャGREETINGをすべてのエディションから削除しても、プロシージャが無くなったことにはならず、すべてのエディションでNON-EXISTENTとなります。

```
SQL> alter session set edition = ora$base;
```

Sessionが変更されました。

```
SQL> drop procedure greeting;
```

Procedure GREETINGが削除されました。

```
SQL> select object_type, object_name, edition_name from user_objects_ae
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
NON-EXISTENT	GREETING	ORA\$BASE
NON-EXISTENT	GREETING	E2

SQL>

このあと、エディションORA\$BASEにて再度プロシージャGREETINGを作成します。

```
SQL> create or replace procedure greeting is
2  begin
3      dbms_output.put_line('Hi, edition 1. ');
4  end;
5  /
```

Procedure GREETINGがコンパイルされました

SQL>

プロシージャの作成状況を確認します。

```
SQL> select object_type, object_name, edition_name from user_objects_ae
where object_name = 'GREETING';
```

OBJECT_TYPE	OBJECT_NAME	EDITION_NAME
PROCEDURE	GREETING	ORA\$BASE
NON-EXISTENT	GREETING	E2

SQL>

一見、親となるエディションでプロシージャGREETINGを作成したので、子となるエディションE2でもプロシージャGREETINGを呼び出せそうですが、以前に削除している情報が残っているため、新たに作成したプロシージャGREETINGは子エディションには継承されません。

再度、継承を有効にする方法は見つかりませんでした。その場合は新たにエディションを作る必要があります。

一番簡単なプロシージャのエディションベースの再定義については以上です。次はエディショニング・ビューに取り組もうと考えています。

完

Yuji N. 時刻: 14:30

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.