

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。

2021年4月6日 火曜日

アプリケーションのグローバル変数(または定数)の扱い

最近、PL/SQL SDKを使ってオブジェクト・ストレージを操作するアプリケーションを作りました。そのアプリケーションで、アプリケーション全体で使用する値として、コンパートメントID、ネームスペース、リージョンおよびクリデンシャル名をアプリケーション・アイテムとして定義しました。

そこで、アプリケーション・アイテムとして定義するのが良い方法だったのか、少々考えるところがありました。複数のページ間で値を共有するために、いくつかの方法を使うことができます。以下の3つの方法です。

1. アプリケーション置換文字列
2. アプリケーション・アイテム
3. アプリケーション設定

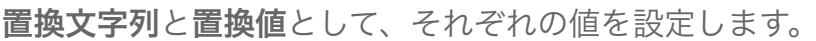
オブジェクト・ストレージを操作するアプリケーションにて定義したアプリケーション・アイテム、**G_COMPARTMENT_ID**、**G_NAMESPACE_NAME**、**G_REGION**、**G_CREDENTIAL_NAME**をそれぞれの方法で定義しなおして、それぞれできること、できないことを紹介します。

アプリケーション置換文字列

アプリケーションをアプリケーション・ビルダーで開いたトップページから、**アプリケーション・プロパティの編集**をクリックして開きます。



または、共有コンポーネントのアプリケーション・ロジックに含まれるアプリケーション定義属性を開きます。



置換文字列、例えば**&G_COMPARTMENT_ID**.やバインド変数:**G_COMPARTMENT_ID**といった指定や**vファンクション**は、アプリケーション・アイテムやページ・アイテムと同じく可能です。

アプリケーション置換文字列の特別な機能として、アプリケーションをインポートするときに値の設定ができます。

サポートするオブジェクトを開きます。



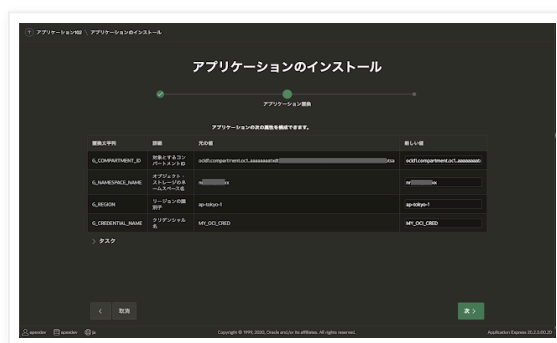
インストールのアプリケーション置換文字列を開きます。



プロンプトにチェックを入れ、プロンプト・テキストを入力します。チェックを入れた置換文字列は、アプリケーションのインポート時にプロンプト・テキストと共に表示され、値の入力が可能になります。この設定を行った後に、アプリケーションをエクスポートします。



アプリケーションのインポート時に表示される画面が以下になります。



詳細としてプロンプト・テキストが表示され、新しい値を元の値から置き換えることにより、置換文字列をインポート時に変更することができます。

アプリケーション置換文字列でできないこと

アプリケーション内で値の変更ができません。つまり、

```
:G_COMPARTMENT_ID := '新しいOCID';
apex_util.set_session_state('G_COMPARTMENT_ID','新しいOCID');
```

といった記述を行った場合、G_COMPARTMENT_IDというアイテムが見つからない、というエラーが発生します。

アプリケーション・アイテム

共有コンポーネントのアプリケーション・ロジックに含まれるアプリケーション・アイテムを開きます。



登録済みのアプリケーション・アイテムが一覧されます。ここから作成や変更を実施します。

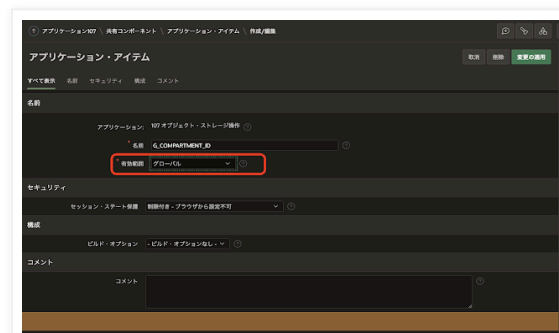


アプリケーション・アイテムでできること

アプリケーションのグローバル変数として、ページを跨った値の設定や参照を行うことができます。置換文字列、例えば&G_COMPARTMENT_ID.、バインド変数:G_COMPARTMENT_ID、vファンクションで扱うことができ、APEX_UTIL.SET_SESSION_STATEプロシージャによっても操作できます。

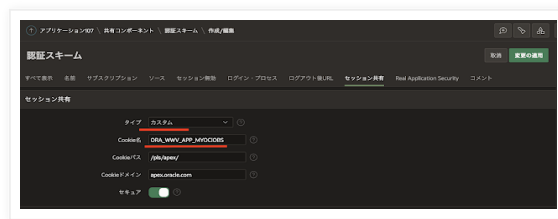
アプリケーション・アイテムであれば、異なるアプリケーション間でも値の共有が可能です。

アプリケーション・アイテムの設定に含まれる有効範囲をグローバルに変更します。



アプリケーション・アイテムを共有するすべてのアプリケーションに同名のアプリケーション・アイテムを定義し、その有効範囲をグローバルに設定します。

続いてアプリケーション間でセッションを共有するため、**認証スキームのセッション共有にてカスタムのクッキーを設定**します。



これもアプリケーション・アイテムを共有する全てのアプリケーションで、同じクッキーを設定します。このように設定することで、複数のアプリケーションでアプリケーション・アイテムを共有することができます。

なお、セッション共有のタイプに**ワークスペース共有**を選択した場合、保存を行うとその設定は、**Cookie名が&WORKSPACE_COOKIE.**である**カスタム**設定になります。**CookieパスやCookieドメイン**の設定なども**ブランク**になり、**セキュアもOFF**であるため、ワークスペース共有はそのまま使用しない方がよいでしょう。

アプリケーション・アイテムでできないこと

アプリケーション置換文字列や、次に説明するアプリケーション定義のように、アプリケーションのインストール時に値を設定することはできません。アプリケーションのエクスポートに値を含むこともできません。

アプリケーション設定

共有コンポーネントのアプリケーション・ロジックに含まれるアプリケーション設定を開きます。



アプリケーション設定の一覧が表示されます。



アプリケーション設定は、以下の設定を含みます。



名前と値の設定以外に、必ず値が設定されていることを保証する**必須の値**のフラグ、設定可能な値をカンマ区切りで指定する**有効な値**、**アップグレード時に値を維持**、の設定があります。

アプリケーション設定でできること

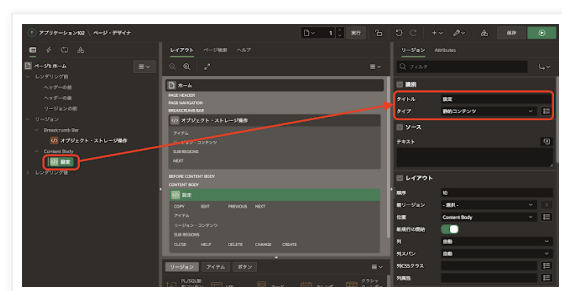
アプリケーション設定は値のプレースホルダーではないので、置換文字列やバインド変数、vファンクションでは使えません。APEX_APP_SETTING.GET_VALUE、およびSET_VALUEプロシージャを呼び出すことで、値の参照と設定を行います。

例えば、アプリケーション設定をリージョンに適用するには、**アプリケーションの計算の計算タイプ**を式に変更し、**計算**に`apex_app_setting.get_value('REGION')`を指定します。



アプリケーション設定を扱う実装を、以下に紹介します。アプリケーション設定として、COMPARTMENT_ID、NAMESPACE_NAME、REGION、CREDENTIAL_NAMEが登録済みとします。

オブジェクト・ストレージを操作するアプリケーションのホーム・ページをページ・デザイナーにて開きます。Content Bodyに**名前を設定**とし、**タイプが静的コンテンツ**のリージョンを新規に作成します。

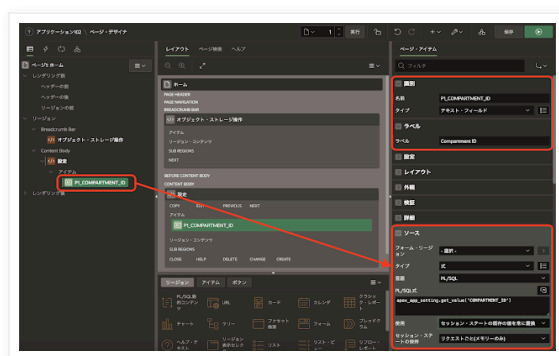


アプリケーション設定の値を保持するページ・アイテムを作成します。最初にコンパートメントIDを作成します。リージョン設定上でコンテキスト・メニューを表示させ、**ページ・アイテムの作成**を実行します。識別の**名前**をP1_COMPARTMENT_IDとします。**タイプ**は**テキスト・フィールド**、**ラベル**を**Compartment ID**とします。

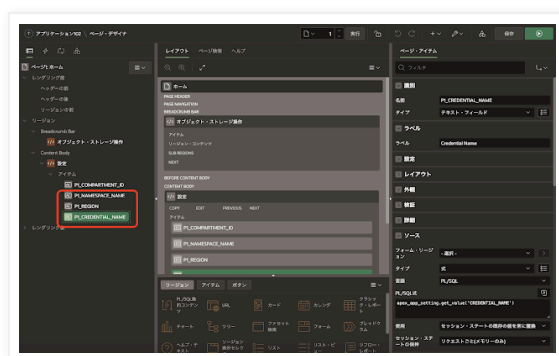
ソースのフォーム・リージョンはフォームではないので無指定(- 選択 -のまま)、タイプを式、言語はPL/SQLを選択し、式に以下を指定します。アプリケーション設定COMPARTMENT_IDの値を、ページ・アイテムのソースとします。

```
apex_app_setting.get_value('COMPARTMENT_ID')
```

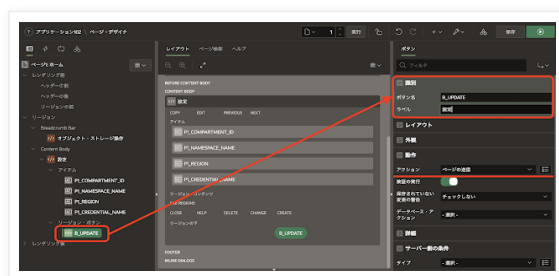
使用にはセッション・ステートの既存の値を常に置換を選択し、ページが呼び出される度に式が評価され、その値がページ・アイテムの値になるようにします。セッション・ステートの保持はリクエストごと(メモリーのみ)を選択します。



同様にページ・アイテムP1_NAMESPACE_NAME、P1_REGIONおよびP1_CREDENTIAL_NAMEを作成します。同じ設定なので、P1_COMPARTMENT_IDを重複させて、名前やラベル、式を変更すると速いでしょう。



値を確定するボタンを作成します。ボタンの作成を実行し、ボタン名をB_UPDATE、ラベルを設定とします。



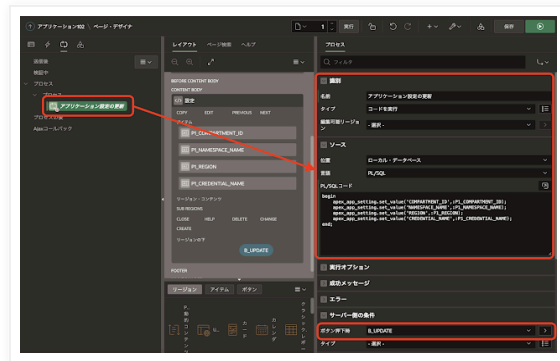
左ペインでプロセス・ビューを開き、実際にアプリケーション設定の更新を行うプロセスの作成をします。プロセス上でコンテキスト・メニューを表示させ、プロセスの作成を実行します。作成したプロセスの名前をアプリケーション設定の更新とし、タイプにコードの実行を選択します。ソースのPL/SQLコードとして以下を記述します。

begin

```
apex_app_setting.set_value('COMPARTMENT_ID',:P1_COMPARTMENT_ID);  
apex_app_setting.set_value('NAMESPACE_NAME',:P1_NAMESPACE_NAME);  
apex_app_setting.set_value('REGION',:P1_REGION);  
apex_app_setting.set_value('CREDENTIAL_NAME',:P1_CREDENTIAL_NAME);
```

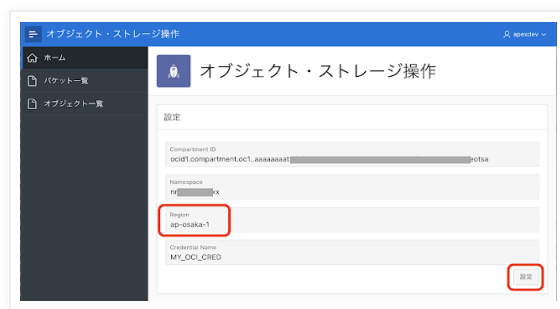
end;

サーバー側の条件として、ボタン押下時にB_UPDATEを選択します。



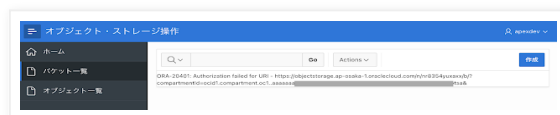
以上でアプリケーションの改変は完了です。アプリケーションを実行して確認します。

Regionをap-tokyo-1からap-osaka-1へ変更し、設定をクリックします。これでアプリケーション設定のREGIONは変更されます。



アプリケーション・アイテムの計算が評価されるのは認証後なので、一旦サインアウトし、再度サインインします。再度サインインした後でも、Regionが上の画面と同じく、変更したap-osaka-1を維持していることが確認できます。

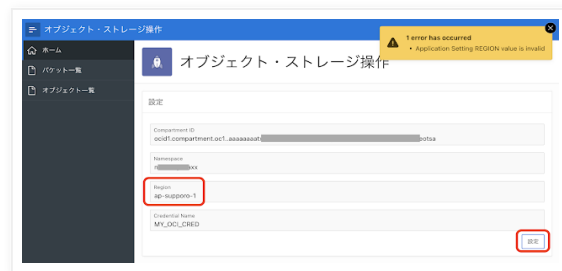
私は大阪リージョンをサブスクライブしていないので、バケット一覧を表示しようとするエラーになりました。



アプリケーション設定では指定可能な値を制限できます。アプリケーション設定を開き、有効な値にap-osaka-1,ap-tokyo-1,us-ashburn-1,us-phoenix-1を設定することで、利用可能なリージョンを4つに制限できます。そして、アプリケーション設定のREGIONの値がap-osaka-1に変更されています。

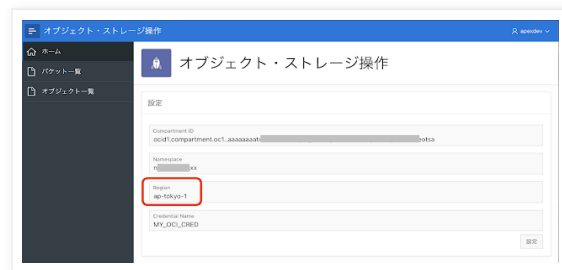


アプリケーション設定に有効な値を設定した後、画面よりRegionとしてap-sapporo-1(このようなリージョンはありません)を設定してみます。Application Setting REGION value is invalidというエラーが発生します。

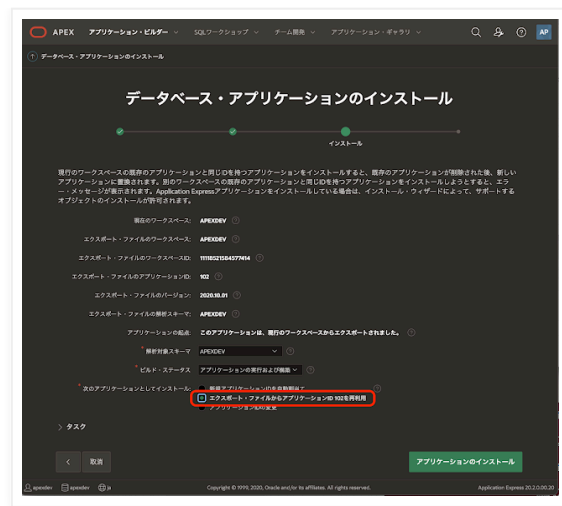


この状態でアプリケーションをエクスポートします。アプリケーション設定のREGIONはap-osaka-1のままエクスポートされます。アップグレード時に値を維持はONになっています。

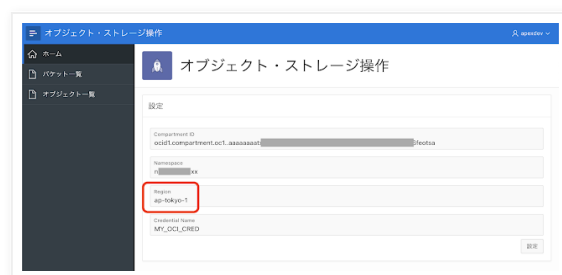
アプリケーションに戻り、Regionをap-tokyo-1に戻します。



この状態で、先程エクスポートしたアプリケーションをインポートします。データベース・アプリケーションのインストールにて、次のアプリケーションとしてインストールにエクスポート・ファイルからアプリケーションID nnn を再利用を選択して、アプリケーションのインストールを実行します。つまり、既存のアプリケーションを置換（アップグレード）します。



アプリケーションを実行すると、**Regionがap-tokyo-1として維持されている**ことが確認できます。



アプリケーション設定の**アップグレード時に値を維持がON**の場合、すでにアプリケーションに設定されている値が維持されます。OFFの場合は、エクスポートに含まれる値が設定されます。今回の例だと、ap-osaka-1になります。

アプリケーション設定でできないこと

アプリケーション設定は値のプレースホルダーではないので、置換文字列やバインド変数、vファンクションでは使えません。

以上で今回の説明は終了です。

アプリケーション置換文字列として各種設定を行うアプリケーションのエクスポートを以下に配置しました。

<https://github.com/ujnak/apexapps/blob/master/exports/obs-app-string.sql>

アプリケーション設定を使うエクスポートはこちらです。

<https://github.com/ujnak/apexapps/blob/master/exports/obs-app-setting.sql>

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: 18:11

共有

[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.
