

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。

2020年6月5日 金曜日

Oracle APEXアプリケーションのグローバル化(4) - アプリケーション自体の対応

Oracle APEXアプリケーションをグローバル化する手順の説明です。 [こちらの記事](#)からの続きになります。

多言語の情報を保持するスキーマ

メニューの情報は表TKY_MENUSに保持されています。この内容は以下になります。

ID	MENU_NAME	VOLUME	PRICE
1	プレミアム牛めし	ミニ盛	490
2	ごろごろ創業ビーフカレー	並盛	790
3	ビビン丼	大盛	600

今の時点では、ログイン時にどの言語を選択しても、レポートには上記が表示されます。まずはじめに、MENU_NAMEについて翻訳したデータを別表に保持し、選択した言語で表示が切り替わるようにアプリケーションを改変します。

MENU_ITEMを各国語で保存するための表TKY_MENUS_TLを以下のように定義します。

```
create table TKY_MENUS_TL
```

```
(
  ID                number primary key,
  MENU_ID           number not null,
  LOCAL_MENU_NAME   varchar2(80) not null,
  LANGUAGE          varchar2(3) not null
);
```

```
alter table TKY_MENUS_TL add constraint TKY_MENUS_TL_FK foreign key ("MENU_ID") references TKY_MENUS ("ID") enable;
```

この表にメニュー名(MENU_ITEM)の翻訳済みの名前を入力します。

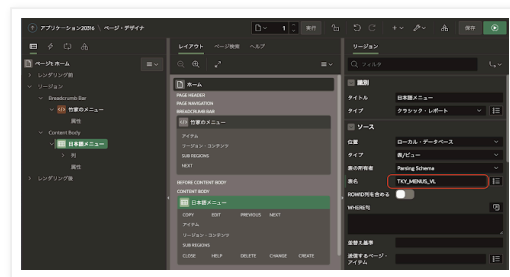
```
begin
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(1, 1, 'プレミアム牛めし', 'JA');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(2, 1, '프리미엄 소고기덮밥', 'KO');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(3, 1, 'Premium Gyumeshi', 'US');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(4, 1, '高級牛肉飯', 'ZHS');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(5, 2, 'ごろごろ創業ビーフカレー', 'JA');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(6, 2, '소고기 듬뿍 타케야원조 소고기 카레', 'KO');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(7, 2, 'Takeya Tender Boil Beef Curry', 'US');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(8, 2, '香炖牛肉竹屋原始咖喱套餐', 'ZHS');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(9, 3, 'ビビン丼', 'JA');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(10, 3, '김치소갈비덮밥', 'KO');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(11, 3, 'BBQ Beef & Kimchi Bowl', 'US');
  insert into tky_menus_tl(id, menu_id, local_menu_name, language) values(12, 3, '泡菜五花牛肉盖饭', 'ZHS');
  commit;
end;
```

LANGUAGEに指定しているJA, KO, US, ZHSといった値ですが、こちらの説明は実装を終えた後に行います。

そして、現在の言語の選択状況にしたがった翻訳結果を表示するビューを以下のように定義します。このビューの要点はSYS_CONTEXT('USERENV','LANG')の使用です。

```
CREATE OR REPLACE FORCE VIEW "TKY_MENU_VL" ("ID", "MENU_NAME", "VOLUME", "PRICE")
AS
select o.id, l.local_menu_name as menu_name, o.volume, o.price
from tky_menus o join tky_menus_tl l on o.id = l.menu_id
where l.language = SYS_CONTEXT('USERENV','LANG');
```

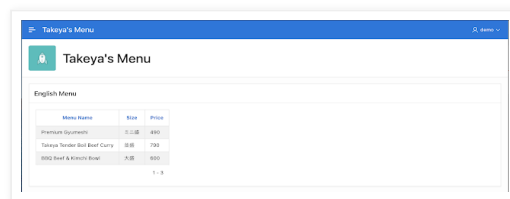
作成したビューTKY_MENU_VLを使って、作成済みのアプリケーションのレポート定義を置き換えます。TKY_MENUであった部分をTKY_MENU_VLに変更します。



元のアプリケーションが変更されたので、再度パブリッシュを行います。



変更されたレポートを確認します。例えばサインインの際に言語セレクトで英語を選択すると、レポートは以下のように表示されます。



メニュー名についてはビュー定義にしたがって、英語で表示されていることが確認できます。同様に中国語、韓国語についても、それぞれの言語で表示されます。

ビューで使用しているファンクションSYS_CONTEXTですが、これ自体はOracle APEXの機能ではなく、Oracle Databaseによって提供されています。SYS_CONTEXTは、データベースのセッションに設定されている属性を返すファンクションです。引数としてUSERENV、LANGを指定することで、セッションに設定されているOracle Databaseの言語を返します。Oracle Database自体のグローバル化セッション・サポートの詳細については[マニュアルの記載](#)に譲ります。SYS_CONTEXTについての記載は[こちら](#)です。

Oracle APEXが受け付けたHTTPのリクエストは必ずデータベースで処理されます。ですので、処理を行うためにデータベースのセッションが取得されます。一般には、新規に作成されるよりは、すでにあるセッションが再利用されますが、データベースでの処理が始まってすぐに、データベースのセッションにOracle APEXのアプリケーションが認識している言語を反映させます。この処理については、デバッグ・ログから確認することができます。

日本語の場合は次のコマンドが実行されています。



英語の場合は次のコマンドが実行されています。



中国語(簡体字)の場合は次のコマンドが実行されています。

```
alter session set nls_language = '中国語' nls_territory='中国';
```

韓国語の場合は次のコマンドが実行されています。

```
alter session set nls_language = '韓国語' nls_territory='韓国';
```

SYS_CONTEXT('USERENV','LANG')は、このようにALTER文を発行することによってセッションに設定されているNLS_LANGUAGEとNLS_TERRITORYの短縮形を返します。

```
alter session set nls_language = '言語' nls_territory='地域';
```

短縮形の情報はマニュアル、グローバルイゼーション・サポート・ガイドのA. ロケール・データ、A1.言語に**対応表**が乗っています。対応表の言語の略称は小文字ですが、SYS_CONTEXT('USERENV','LANG')の値は大文字になります。

Oracle APEXはアプリケーションのグローバルイゼーションに関して、このような仕組みを持っています。この仕組みを前提として、翻訳したメニュー名を保持する表TKY_MENU_TLのLANGUAGE列にJA, US, ZHS, KOなどのOracle Databaseがサポートしている言語の短縮名を設定しており、それを参照するビューTKY_MENU_VLではSYS_CONTEXT('USERENV','LANG')で返される言語に一致する行を選択することで、メニュー名をアプリケーションのサインイン時に設定した言語で表示させています。

動的翻訳

お気づきのように、アプリケーションの翻訳は完全ではありません。サイズの表記が日本語のままです。この翻訳には、Oracle APEXが提供している動的翻訳を使ってみます。

共有コンポーネントのアプリケーションの翻訳より、**動的翻訳**を開きます。



動的翻訳の準備として、プライマリ言語の文字列と、それを翻訳した文字列を対にして、動的翻訳リポジトリに登録しておきます。そして、APEX_LANG.LANGファンクションをプライマリ言語の文字列を引数にして呼び出し、翻訳済みの文字列を返り値として得ます。

動的翻訳を開くと一覧画面になります。**作成**をクリックします。



動的翻訳の編集画面になります。**翻訳元テキスト**はつねにアプリケーションのプライマリ言語(この場合は日本語)での文字列です。**言語**の選択は、翻訳先となる言語で、**翻訳先テキスト**は翻訳元テキストに対応した翻訳済みの文字列です。



サイズの表記は以下ですので、これを動的翻訳として登録します。各言語3行、合計で9行文の登録になります。

日本語	英語	中国語(簡体字)	韓国語
ミニ盛	Mini	小碗	미니사이즈
並盛	Regular	平碗	보통사이즈
大盛	Large	大碗	곱빼기

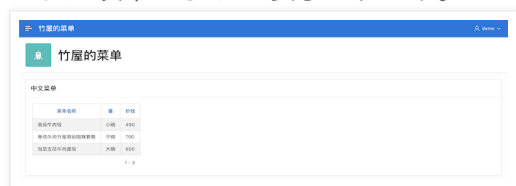
すべて登録すると以下のようになります。



では、動的翻訳をアプリケーションに適用します。先ほど作成したビューの定義を修正します。変更点は volume をそのまま渡していたところを、APEX_LANG.LANG(volume)として、動的翻訳を適用させている点です。

```
CREATE OR REPLACE FORCE VIEW "TKY_MENUS_VL" ("ID", "MENU_NAME", "VOLUME", "PRICE") AS
  select o.id, l.local_menu_name as menu_name, apex_lang.lang(o.volume) as volume, o.price
  from tky_menus o join tky_menus_tl l on o.id = l.menu_id
  where l.language = SYS_CONTEXT('USERENV','LANG');
```

変更されたレポートを確認します。例えばサインインの際に言語セレクトで**中文（簡体）**を選択すると、レポートは以下のように表示されます。



サイズの表記が中国語になっていることがわかります。英語、韓国語も同様に言語の選択に応じた表記に変わります。

APEX_LANG.LANGファンクションに関するマニュアルの記載は[こちら](#)です。

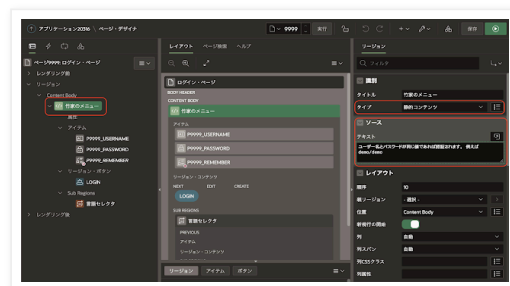
これでレポートについては、多言語対応が完了しました。

多言語に対応したメッセージ

このアプリケーションはデモ用途で、ユーザー名とパスワードに同じ値を使用するだけでユーザー認証が成功します。それをサインインのダイアログにメッセージとして埋め込んでみます。以下のようなメッセージです。



一番簡単な方法は、リージョンのソースに直接、メッセージを書き込むことです。

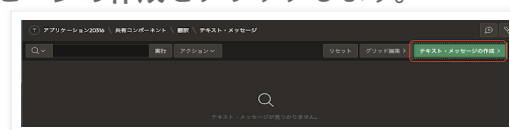


日本語だけの対応であれば、これで十分です。Oracle APEXのアプリケーションの多くは、このように直接メッセージを書き込んでいると思います。

このようなケースを多言語に対応させる方法として、共有コンポーネントのアプリケーションの翻訳の中にテキスト・メッセージというリポジトリが提供されています。



テキスト・メッセージを開くと、登録されているメッセージが一覧されます。新規にテキスト・メッセージを登録するには、テキスト・メッセージの作成をクリックします。



テキスト・メッセージの登録には、APEX_LANG.MESSAGEファンクションがテキスト・メッセージを呼び出す際の引数となる名前、メッセージの言語、その言語でのメッセージとして表示されるテキストを指定します。JavaScriptで使用にチェックを入れると、JavaScript APIのapex.langコールでも、ここで指定したメッセ

ージが利用できるようになります。これらを指定した後、**テキスト・メッセージの作成**(または、**作成後、さらに作成**)をクリックします。



テキスト・メッセージの名前をT_SIGNIN_MESSAGEとして、以下、4 言語分のテキスト・メッセージを登録します。

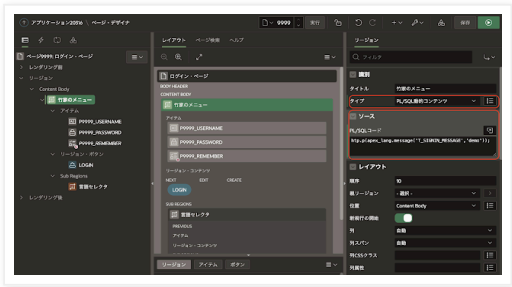
言語	テキスト・メッセージ
ja	ユーザー名とパスワードが同じ値であれば認証されます。 例えば %0/%0
en	If the user name and password are the same, you will be authenticated. For example, %0/%0
zh-cn	如果用户名和密码相同，则将通过身份验证。 例如 %0/%0
ko	사용자 이름과 암호가 같은 값이면 인증됩니다. 예 %0/%0

結果として以下のように登録されます。



登録されたテキスト・メッセージを使って、サインインのダイアログに表示するメッセージを多言語に対応させます。ページ・デザイナーにてサインインのページを開き、リージョンの**タイプ**を**PL/SQL動的コンテンツ**に変更し、**ソース**の**PL/SQLコード**に以下を指定します。

```
http.p(apex_lang.message('T_SIGNIN_MESSAGE','demo'));
```



アプリケーションの変更をすべての翻訳済みアプリケーションに反映させるため、パブリッシュを実行します。

アプリケーションを実行して動作を確認します。サインイン画面の言語セレクトで**한국어(韓国語)**を選択すると、メッセージがハングルで表記されていることが確認できます。英語、中国語も同様です。

APEX_LANG.MESSAGEファンクションの詳細については、マニュアルの[こちら](#)に説明があります。

JavaScriptで使用する文字列の翻訳

今回のサンプル・アプリケーションには含まれていませんが、データの編集を行うフォームをウィザードで作成し、そのフォーム上の削除ボタンをクリックすると、以下のポップアップが表示されます。

これはボタンにたいして、以下のJavaScriptを実行することで実現しています。

```
javascript:apex.confirm(htmlldb_delete_message,'DELETE');
```

「削除操作を実行しますか?」というメッセージは、そのままでは多言語に対応しません。このメッセージを多言語に対応させるには、先ほどのテキスト・メッセージのリポジトリとJavaScript APIの `apex.lang.formatMessage` を使用します。

まず、先ほどと同様にテキスト・メッセージを登録します。名前は `T_CONFIRM_DELETE` とします。**JavaScriptで使用をON**にします。



各言語で対応するテキスト・メッセージを登録します。

言語	テキスト・メッセージ
ja	この削除操作を実行しますか。
en	Would you like to perform this delete action?
zh-cn	是否要执行此删除操作?
ko	이 삭제 작업을 수행하겠습니까?

全て登録すると、以下のようになります。

名前	言語	テキスト	JavaScriptの値	メッセージID
T_CONFIRM_DELETE	ja-jp	この削除操作を実行しますか。	11	11
T_CONFIRM_DELETE	en	Would you like to perform this delete action?	12	12
T_CONFIRM_DELETE	zh-cn	是否要执行此删除操作?	13	13
T_CONFIRM_DELETE	ko	이 삭제 작업을 수행하겠습니까?	14	14

テキスト・メッセージの準備ができたので、先ほどのJavaScriptの記述を以下に変更します。

```
javascript:apex.confirm(apex.lang.getMessage('T_CONFIRM_DELETE'),'DELETE');
```

アプリケーションの変更があったので、シードとパブリッシュを再実行します。このようにアプリケーションを設定することで、ダイアログの表示が切り替わります。

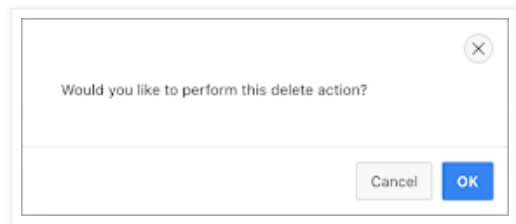
JavaScript APIを使った翻訳を確認する場合、アプリケーション・ビルダーからアプリケーションを起動せず、直接、ブラウザからアプリケーションにアクセスしてください。アプリケーション・ビルダーからアクセスすると、アプリケーション・ビルダーの言語で翻訳されます。つまり開発を日本語で行っている場合、そこからアプリケーションを起動すると、アプリケーション側でセッションの言語を切り替えても、つねに開発画面の言語、この場合は日本語になります。

以下、多言語に対応したダイアログの表示です。

簡体中国語では以下になります。



英語では以下になります。



韓国語では以下になります。



Oracle APEXが提供するJavaScript APIのマニュアルの記載は[こちら](#)になります。新しい版については、まだ日本語の翻訳が準備できていません。翻訳に関するAPIでは、

`apex.lang.getMessage`
`apex.lang.formatMessage`
`apex.lang.formatMessageNoEscape`

などが使用できます。

Oracle APEXのアプリケーションで使用できる、文字列の翻訳に関する機能の紹介は以上になります。

[続く](#)

Yuji N. 時刻: 19:50

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.