

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12月14日 水曜日

アプリケーション・インポート後の自動化を自動的にアクティブにする

アプリケーションに作成されている自動化は、アプリケーションをインポートした時点では無効になっています。インポート後にいきなりアクティブになり、知らないうちに処理が実行されても困るので、これは妥当な仕様でしょう。

Oracle APEX 22.1のマニュアルでは以下のセクションに説明があります。

アプリケーション・ビルダー・ユーザズ・ガイド

19.5.2 自動化のインポートとエクスポートについて

とはいえ、インポート後に操作しなくてもアクティブになって欲しい自動化の処理もあるようです。少し実装方法について考えてみました。

これも自動化を使って実装します。

Oracle APEXの自動化はAPEXの標準ビュー `APEX_APPL_AUTOMATIONS` より、登録されている自動化やそのステータスを確認できます。

以下のSELECT文で無効になっている自動化を見つけることができます。**静的ID**(列 `static_id`) が **auto-enabled-** で始まるという条件を付けることにより、インポート後に自動的にアクティブにしたい自動化を選択しています。

```
select automation_id, application_id, static_id from apex_appl_automations
where static_id like 'auto-enable-%'
and polling_status_code = 'DISABLED'
```

enable-candidate-automation.sql hosted with ❤ by GitHub

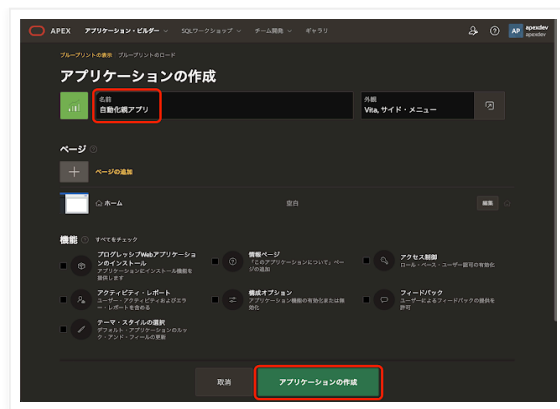
[view raw](#)

自動化をアクティブにするためにプロシージャ `APEX_AUTOMATION.ENABLE` が使えます。このプロシージャの引数は自動化が作成されている **アプリケーションID** と自動化の **静的ID** です。そのため、上記のSELECT文で検索された値を引数として `APEX_AUTOMATION.ENABLE` を呼び出すことで、自動化をアクティブにできそうです。

APEXアプリケーションを作成し、上記の処理を実装してみます。

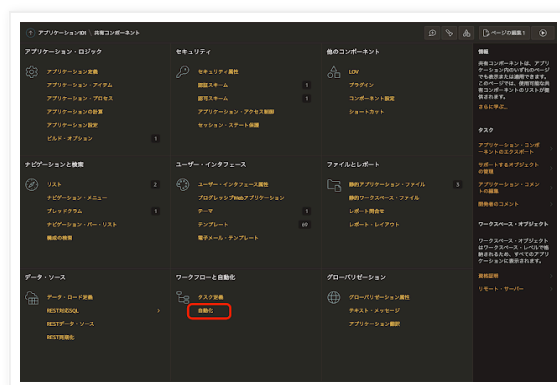
アプリケーション作成ウィザードを起動し、空のアプリケーションを作成します。**名前を自動化親アプリ**とします。

アプリケーションの作成をクリックします。

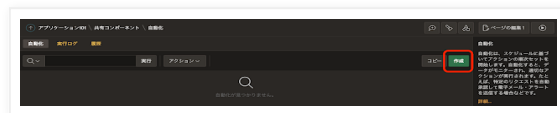


アプリケーションが作成されたら、自動化の作成を行います。

共有コンポーネントの自動化を開きます。

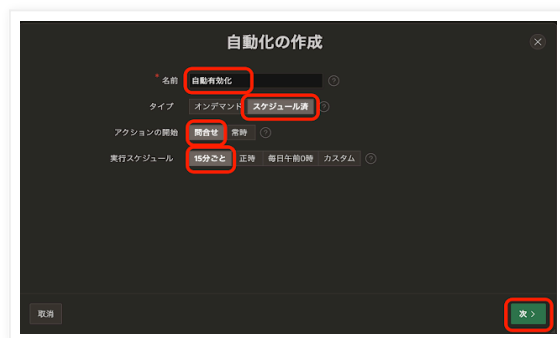


作成済みの自動化が一覧されます。作成をクリックします。



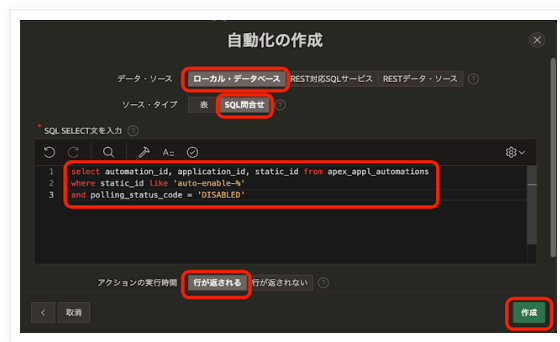
名前は自動有効化とします。タイプはスケジュール済（翻訳がいまひとつです。定期実行または繰り返し実行のことです）、アクションの開始に問合せを選択します。実行スケジュールは、この中で一番短い15分ごとを選択します。

次へ進みます。



データ・ソースはローカル・データベース、ソース・タイプとしてSQL問合せを選択し、SQL SELECT文を入力に、アクティブに変更する自動化を検索するSELECT文を記述します。アクション

の**実行時間**（これも翻訳が今ひとつです。アクションが実行されるタイミングです。）として**行が返される**を選択します。



以上で**作成**をクリックします。

自動化が作成されます。

最初に**アクションの実行**のセクションを変更します。

アクションの実行時間は**行が返される**のままです。主キー列として**AUTOMATION_ID(Number)**を選択します。設定ミスなどがあり自動化が大量に実行されることを防ぐため、**処理する最大行数に1**、**アクションのエラーの処理**に**自動化の無効化**を選択します。動作の確認ができれば、より緩やかな設定に変更します。



アクションにデフォルトで作成されている**新規アクション**を編集します。

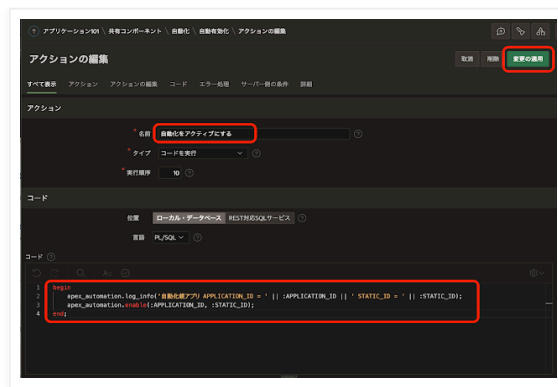


アクションの名前を**自動化をアクティブにする**に変更し、コードからは**APEX_AUTOMATION.ENABLE**を呼び出します。

```
begin
    apex_automation.log_info('自動化親アプリ APPLICATION_ID = ' || :APPLICATION_ID || ' STATIC_ID = ' || :STATIC_ID);
    apex_automation.enable(:APPLICATION_ID, :STATIC_ID);
end;
```

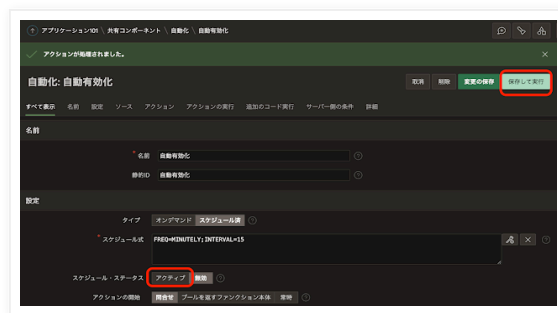
apex_automation_enable.sql hosted with ❤ by GitHub

[view raw](#)



以上で自動化の作成は完了です。

スケジュール・ステータスをアクティブにし、保存して実行します。

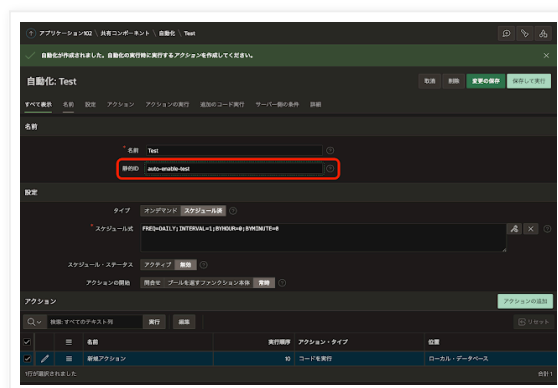


実行ログは、自動化の実行ログより確認できます。



動作確認のために別にAPEXアプリケーションを作成し、自動化を作成します。静的IDはauto-enable-testとします。スケジュール・ステータスは無効にします。

特にAlways Freeのインスタンスを使用している場合ですが、OCPUに限りがあるので無駄に自動化が動かないよう、一日に一度など少ない実行間隔にしておきます。



アプリケーション・ビルダーには複数のアプリケーションの自動化の状態を一度に確認する画面は無いみたいなので、以下のSELECT文により動作を確認します。

```
select
```

```
application_name
,name
,static_id
,polling_status_code
,polling_last_run_timestamp at time zone 'Asia/Tokyo'
,polling_next_run_timestamp at time zone 'Asia/Tokyo'
from apex_appl_automations;
```

automation_status_confirm.sql hosted with ❤ by GitHub

[view raw](#)

POLLING_STATUS_CODEがDISABLEDだった自動化がACTIVEに変わっていたら、期待通りに動作していると言えます。

自動化以外にもないですが、アプリケーションのエクスポートを以下に置きました。
<https://github.com/ujnak/apexapps/blob/master/exports/enable-automation-by-automation.zip>

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: 18:18

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.