

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。

2023年5月8日月曜日

オーバーロードされたプロシージャをAPI呼び出しで使用する

Oracle APEX 22.2よりプロセス・タイプとしてAPIの呼び出し(Invoke API)が追加されています。PL/SQLのサブプログラム（プロシージャおよびファンクション）をAPEXのプロセスとして呼び出すことができます。

PL/SQLでは引数の名前や数、データ型が異なるサブプログラムを同じサブプログラム名で作成できます。一般にはオーバーロードと呼ばれます。PL/SQL言語リファレンスでは、以下のように説明されています。

8 PL/SQLサブプログラム

8.9 オーバーロードされたサブプログラム

https://docs.oracle.com/cd/F19136_01/lnpls/plsql-subprograms.html#GUID-47D5A50E-7AAF-4C80-A06A-37593EA2526A

オーバーロードされているプロシージャまたはファンクションをAPIの呼び出しに指定する方法を紹介します。

今回はOracle APEX 23.1に追加されたパッケージAPEX_BACKGROUND_PROCESSに含まれるABORTプロシージャを呼び出してみます。

このプロシージャをプロセスに設定すると、以下のように警告が表示されます。

プロセス → オーバーロード確認 → 設定 → プロシージャまたはファンクション

プロシージャまたはファンクションTESTはオーバーロードされています。パッケージ仕様部で最初に出現するパラメータを使用しています。



プロシージャAPEX_BACKGROUND_PROCESS.ABORTのパラメータとしてp_application_idとp_process_idが認識されています。

パッケージAPEX_BACKGROUND_PROCESSの定義を確認してみます。

一般にOracle APEXに含まれるPL/SQLパッケージはシノニムとして定義されています。今回の例ではAPEX_BACKGROUND_PROCESSがシノニムになります。このシノニムより、実体となるパッケージを求めます。ビューALL_SYNONYMSを検索します。

```
select * from all_synonyms where synonym_name = 'APEX_BACKGROUND_PROCESS';
```

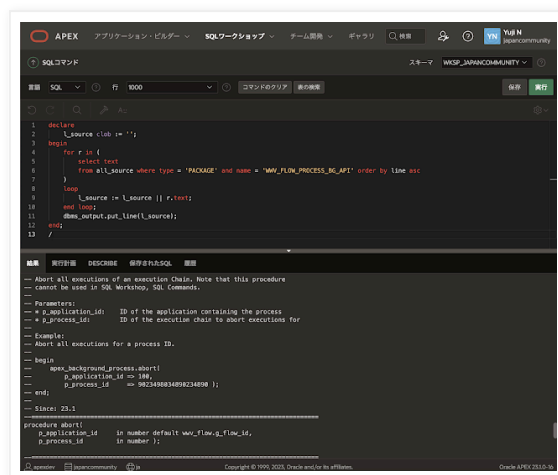
report					
OWNER	SYNONYM NAME	TABLE OWNER	TABLE NAME	DB LINK	ORIGIN COL ID
PUBLIC	APEX_BACKGROUND_PROCESS	APEX_230100	WWV_FLOW_PROCESS_BG_API		0

このパッケージが**WWV_FLOW_PROCESS_BG_API**として実装されていることがわかります。このパッケージの定義部のソースを出力します。

```
declare
    l_source clob := '';
begin
    for r in (
        select text
        from all_source where type = 'PACKAGE' and name = 'WWV_FLOW_PROCESS_BG_API' order by li
    )
    loop
        l_source := l_source || r.text;
    end loop;
    dbms_output.put_line(l_source);
end;
```

print-package-def.sql hosted with ❤ by GitHub

[view raw](#)



プロシージャABORTに関しては、以下の2通りの呼び出し方法が定義されていることが確認できます。

ひとつは**p_process_id**を引数とします。

```
=====
-- Abort all executions of an execution Chain. Note that this procedure
-- cannot be used in SQL Workshop, SQL Commands.
--
-- Parameters:
-- * p_application_id:    ID of the application containing the process
-- * p_process_id:       ID of the execution chain to abort executions for
--
-- Example:
-- Abort all executions for a process ID.
```

```
--
-- begin
--     apex_background_process.abort(
--         p_application_id => 100,
--         p_process_id      => 9023498034890234890 );
-- end;
--
-- Since: 23.1
=====
procedure abort(
    p_application_id    in number default wwv_flow.g_flow_id,
    p_process_id        in number );
```

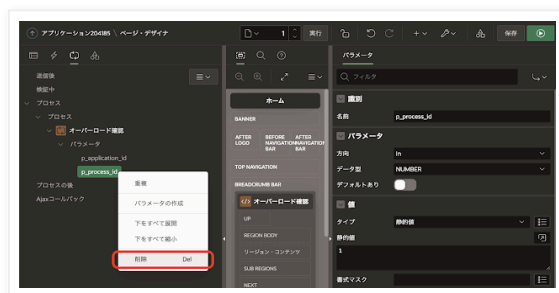
もうひとつはp_execution_idを引数とします。

```
=====
-- Abort a specific execution of an execution Chain. Note that this procedure
-- cannot be used in SQL Workshop, SQL Commands.
--
-- Parameters:
-- * p_application_id:    ID of the application containing the process
-- * p_execution_id:     ID of the execution to abort
--
-- Example:
-- Abort background execution ID 4711
--
-- begin
--     apex_background_process.abort(
--         p_application_id => 100,
--         p_execution_id   => 4711 );
-- end;
--
-- Since: 23.1
=====
procedure abort(
    p_application_id    in number default wwv_flow.g_flow_id,
    p_execution_id      in number );
```

プロセス・タイプAPIの呼び出しでは最初に出現するパラメータが選択されるため、p_process_idが選択されます。

APEX 22.2の時点では、APIの呼び出しよりシグネチャが異なるサブプログラムを呼び出すには、パラメータの削除と作成を手作業で行なう必要があります。

パラメータp_process_idの代わりにp_execution_idを引数に与えるためには、パラメータp_process_idを削除します。



続いてパラメータの作成を実行します。

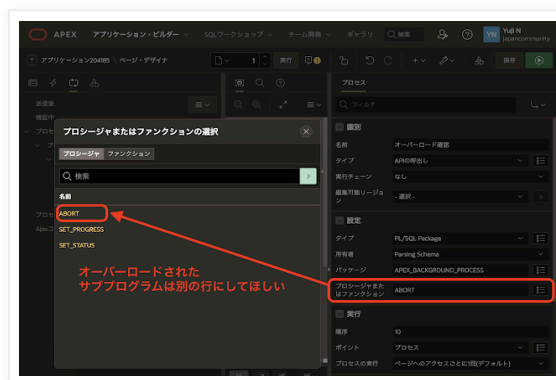


識別の名前やパラメータの方向、データ型、デフォルトありなども、手作業で設定します。今回の例では、名前はp_execution_id、方向はIn、データ型はNUMBER、デフォルトありはOFFになります。



オーバーロードされたサブプログラムの呼び出し方法は以上になります。

プロセス・タイプのAPIの呼び出しでは、マニュアルなどを参照せずに引数の指定をできるのが利点のひとつです。APIの呼び出しの設定のプロシージャまたはファンクションからサブプログラムを選択する際に、オーバーロードされたサブプログラムは別の行として選択できるようになって欲しいと思います。



完

Yuji N. 時刻: 11:01

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.
