

PLSQL の共通オブジェクト定義を、以下のようにします。

1 . ARG_ARRAY

```
CREATE OR REPLACE TYPE ARG_ARRAY AS VARRAY(30) OF VARCHAR2(100);
/
```

引数の受け渡しは、配列タイプで受け渡す。
 すべて文字列タイプとし、必要であれば PL / SQL 内で NUMBER 型等に変換する。
 引数の最大数は、30個とし、最大文字数は、100桁とする。
 引数の順番は、queryタグのargs 属性で設定する。
 もちろん、受け取り側は、その順番で受け取る必要がある。

2 . ERR_MSG_ARRAY

```
CREATE OR REPLACE TYPE ERR_MSG AS OBJECT
( NO NUMBER, /* 行番号 */
  KEKKA OUT NUMBER , /* 結果 0 :正常 1 :警告 2 :異常 */
  ID VARCHAR2(10), /* エラーメッセージID */
  MSG1 VARCHAR2(100), /* メッセージの引数 1 or ORACLEメッセージ */
  MSG2 VARCHAR2(20), /* メッセージの引数 2 */
  .....
  MSG5 VARCHAR2(20) ); /* メッセージの引数 5 */
/
```

```
CREATE OR REPLACE TYPE ERR_MSG_ARRAY AS VARRAY(100) OF ERR_MSG;
/
```

エラーメッセージは、オブジェクトタイプで宣言し、最大100件まで登録することができる。
 それ以上のエラーが出た場合は表示を打ち切る事とする。
 エラーメッセージに引数を5個まで登録できる。

3 . SY_SARG_ARRAY

```
CREATE OR REPLACE TYPE SY_SARG AS OBJECT
( NO NUMBER, /* 行番号 */
  CDKH VARCHAR2(1) ); /* 改廃コード A :追加 C :変更 D :削除 */
/
```

```
CREATE OR REPLACE TYPE SY_SARG_ARRAY AS VARRAY(100) OF SY_SARG;
/
```

プロシジャーに渡すシステム制御情報は、行番号と改廃コードとする。
 エラーメッセージや処理の振り分けは、これらの情報を用いてPL/SQL 側で
 記述する。
 これらも、100件までの情報を渡すが、それ以上の登録が画面から必要な場合は、
 別途専用の PL/SQL と、登録用Java プログラムが必要になる。
 それ以上の情報の一括登録は、画面情報をインプットとするのではなく、
 別に、要求テーブルや、ファイルなどを利用してDB登録する事を推奨する。

4 . PROCEDURE SET_ERRMSG

```
CREATE OR REPLACE PROCEDURE SET_ERRMSG
(P_ERRMSG IN OUT ERR_MSG_ARRAY,
 P_NO IN NUMBER := NULL,
 P_KEKKA IN NUMBER := NULL,
 P_ID IN VARCHAR2 := NULL,
 P_MSG1 IN VARCHAR2 := NULL,
 P_MSG2 IN VARCHAR2 := NULL,
 P_MSG3 IN VARCHAR2 := NULL,
 P_MSG4 IN VARCHAR2 := NULL,
 P_MSG5 IN VARCHAR2 := NULL) IS
```

```
TOO_MANY_ERRORS EXCEPTION; -- エラー件数最大数オーバーエラー
V_ERRMSG ERR_MSG; -- エラーメッセージオブジェクト
BEGIN
```

エラーメッセージ配列を設定する場合の初期化などを行うプロシージャ