

Created by Ahsan Arif

WHEN OTHERS Clause

The WHEN OTHERS clause is used to trap all remaining exceptions that have not been handled by your Named System Exceptions and Named Programmer-Defined Exceptions.

The syntax for the WHEN OTHERS clause in a procedure is:

```
CREATE [OR REPLACE] PROCEDURE procedure_name
  [ (parameter [,parameter]) ]
IS
  [declaration_section]
BEGIN
  executable_section

EXCEPTION
  WHEN exception_name1 THEN
    [statements]

  WHEN exception_name2 THEN
    [statements]

  WHEN exception_name_n THEN
    [statements]

  WHEN OTHERS THEN
    [statements]

END [procedure_name];
```

The syntax for the WHEN OTHERS clause in a function is:

```
CREATE [OR REPLACE] FUNCTION function_name
  [ (parameter [,parameter]) ]
  RETURN return_datatype
IS | AS
  [declaration_section]
BEGIN
  executable_section
```

```

EXCEPTION
  WHEN exception_name1 THEN
    [statements]

  WHEN exception_name2 THEN
    [statements]

  WHEN exception_name_n THEN
    [statements]

  WHEN OTHERS THEN
    [statements]

END [function_name];

```

Here is an example of a procedure that uses a **WHEN OTHERS** clause:

```

CREATE OR REPLACE PROCEDURE add_new_order
  (order_id_in IN NUMBER, sales_in IN NUMBER)
IS
  no_sales EXCEPTION;

BEGIN
  IF sales_in = 0 THEN
    RAISE no_sales;

  ELSE
    INSERT INTO orders (order_id, total_sales )
      VALUES ( order_id_in, sales_in );
  END IF;

  EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
      raise_application_error (-20001,'You have tried to insert a duplicate order_id.');
```

```

    WHEN no_sales THEN
      raise_application_error (-20001,'You must have sales in order to submit the order.');
```

```

    WHEN OTHERS THEN
      raise_application_error (-20002,'An error has occurred inserting an order.');
```

```

END;

```

In this example, if an exception is encountered that is not a **DUP_VAL_ON_INDEX** or a **no_sales**, it will be trapped by the **WHEN OTHERS** clause.

Frequently Asked Questions

Question: Is there any way to get the ORA error number (and/or description) for the errors that will fall into OTHERS?

Something like:

```
WHEN OTHERS THEN
'Error number ' & Err.Number& ' has happen.'
```

Answer: Yes, you can use SQLCODE function to retrieve the error number and SQLERRM function to retrieve the error message.

For example, you could raise the error as follows:

```
EXCEPTION
  WHEN OTHERS THEN
    raise_application_error(-20001,'An error was encountered - '||SQLCODE||' -ERROR-
    '||SQLERRM);
END;
```

Or you could log the error to a table as follows:

```
EXCEPTION
  WHEN OTHERS THEN
    err_code := SQLCODE;
    err_msg := substr(SQLERRM, 1, 200);

    INSERT INTO audit_table (error_number, error_message)
    VALUES (err_code, err_msg);
END;
```