

PL/SQL static code analysis: "DBMS_UTILITY.FORMAT_ERROR_STACK" and "FORMAT_ERROR_BACKTRACE" should be used together

1-2 minutes

Since Oracle 10g, `DBMS_UTILITY.FORMAT_ERROR_BACKTRACE` is available to get an exception's stack trace, i.e. files and lines that lead up to the exception. When combined with `DBMS_UTILITY.FORMAT_ERROR_STACK`, which contains the exception error code and message, developers are able quickly identify defects.

This rule verifies that whenever either is used in an exception handler, the other is used as well.

Noncompliant Code Example

```
BEGIN
  RAISE_APPLICATION_ERROR(-20000, 'This is an error
example');
EXCEPTION
  WHEN OTHERS THEN -- Noncompliant; only
FORMAT_ERROR_STACK is used

  DBMS_OUTPUT.PUT(DBMS_UTILITY.FORMAT_ERROR_STACK);
  -- "ORA-20000: This is an error example"
  DBMS_OUTPUT.PUT_LINE("");
END;
```

/

Compliant Solution

```
BEGIN
```

```
  RAISE_APPLICATION_ERROR(-20000, 'This is an error  
example');
```

```
EXCEPTION
```

```
  WHEN OTHERS THEN
```

```
    DBMS_OUTPUT.PUT(DBMS_UTILITY.FORMAT_ERROR_STACK);
```

```
    -- "ORA-20000: This is an error example"
```

```
    DBMS_OUTPUT.PUT(DBMS_UTILITY.FORMAT_ERROR_BACKTRACE);
```

```
    -- "ORA-06512: at line 2"
```

```
      DBMS_OUTPUT.PUT_LINE("");
```

```
END;
```

/