



One weakness of Oracle error handling in previous versions was tracing where the error first began, since Oracle 10g Developers have has two new processes in the DBMS_UTILITY package.

These are *DBMS_UTILITY.FORMAT_ERROR_STACK* and *DBMS_UTILITY.FORMAT_ERROR_BACKTRACE*



This function will retrieve the current error stack and can be viewed as a SQLERRM replacement, and given that it has a limit of 2000 characters as opposed to 512 with SQLERRM it has obvious advantages.

```
declare
  l_number number(2);
begin
  l_number := 99 + 1;
exception
  when others then
    dbms_output.put_line(DBMS_UTILITY.FORMAT_ERROR_STACK);
end;
ORA-06502: PL/SQL: numeric or value error: number precision too large
```



This function will output the history of the errors as they are re-traced through the various calling processes, this literally shows at what line each process failed until it reached the appropriate exception handler. In this example the previous ORA-06502 is simulated five levels down the procedures to show the results of using the function

```
create or replace procedure one is  
begin  
    two;  
exception  
    when others then  
        dbms_output.put_line(dbms_utility.format_error_stack);  
        dbms_output.put_line(dbms_utility.format_error_backtrace);  
end;
```

Error Handling – Using DBMS_UTILITY.FORMAT_BACKTRACE



```
create or replace procedure two is  
begin three; end;
```

```
create or replace procedure three is  
begin four; end;
```

```
create or replace procedure four is  
begin five; end;
```

```
create or replace procedure five is  
  l_number number(2);  
begin  
  l_number := 99 + 1;  
end;
```



The result DBMS_UTILITY.FORMAT_BACKTRACE is as follows

...

SQL> execute one

ORA-06502: PL/SQL: numeric or value error: number precision too large

ORA-06512: at "SQL1000_USER.FIVE", line 4

ORA-06512: at "SQL1000_USER.FOUR", line 3

ORA-06512: at "SQL1000_USER.THREE", line 3

ORA-06512: at "SQL1000_USER.TWO", line 3

ORA-06512:

at "SQL1000_USER.ONE", line 3