



from the prying eyes of competitors. It is less obvious that one should use it as a general mechanism to hide code from other developers in your organization.

The problem with obfuscating your code is that no one can read the source code in the database, even the owner of the code, even those who need to debug the code to fix problems.

If you would like to compile your program into the database so that the owner of the program can read and work with code, but others (who are allowed to run the program) cannot see the implementation, you should define your programs in packages.

If you implement your functionality in schema-level (a.k.a, stand-alone) procedures and functions, then granting execute authority also allows programmers connected to other schema to see the source code of those program units. This fact is demonstrated below:

```
SQL> CONNECT scott/tiger
Connected.
SQL> CREATE OR REPLACE PROCEDURE my_procedure
 2 IS
 3 BEGIN
 4     DBMS_OUTPUT.put_line ( 'my_procedure' );
 5 END my_procedure;
 6 /

Procedure created.

SQL> GRANT EXECUTE ON my_procedure TO hr
 2 /

Grant succeeded.

SQL> CONNECT hr/hr
Connected.
SQL> SELECT     TYPE, text
 2             FROM all_source
 3             WHERE owner = 'SCOTT' AND NAME = 'MY_PROCEDURE'
 4             ORDER BY TYPE, line
 5 /

TYPE                TEXT
-----
PROCEDURE           PROCEDURE my_procedure
PROCEDURE           IS
PROCEDURE           BEGIN
PROCEDURE           DBMS_OUTPUT.put_line ( 'my_procedure' );
PROCEDURE           END my_procedure;
```

Suppose that you instead implement your procedures and functions within a package. You then can grant execute authority on that package to other schemas. Users connected to those schemas will be able to only see the package specifications, and not the bodies. This fact is demonstrated below:

```
SQL> CONNECT scott/tiger
Connected.
SQL> CREATE OR REPLACE PACKAGE my_package
 2 IS
 3     PROCEDURE hidden_implementation;
 4 END my_package;
 5 /

Package created.

SQL> CREATE OR REPLACE PACKAGE BODY my_package
 2 IS
 3     PROCEDURE hidden_implementation
 4     IS
 5     BEGIN
 6         DBMS_OUTPUT.put_line ( 'hidden_implementation' );
 7     END hidden_implementation;
 8 END my_package;
 9 /

Package body created.

SQL> GRANT EXECUTE ON my_package TO hr
 2 /

Grant succeeded.

SQL> CONNECT hr/hr
Connected.
SQL> SELECT     TYPE, text
 2             FROM all_source
 3             WHERE owner = 'SCOTT' AND NAME = 'MY_PACKAGE'
 4             ORDER BY TYPE, line
 5 /

TYPE                TEXT
-----
PACKAGE             PACKAGE my_package
```

```
PACKAGE          IS  
PACKAGE          PROCEDURE hidden_implementation;  
PACKAGE          END my_package;
```

Notice that none of the lines from the package body are visible through ALL\_SOURCE.

[Back to the Table of Contents](#)

---

[Printer View](#)