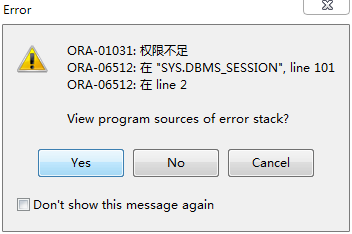
DBMS\_SESSION.SET\_CONTEXT抛异常

在调用存储过程时失败，经过调试发现，在调用pkg\_config包中的接口时发生异常，调试进入该接口中时，发现异常语句为DBMS\_SESSION.SET\_CONTEXT(user,str\_key,str\_value)，其中str\_key的值为配置项的名称，str\_value为配置项的值value，但是不知道user是什么（user为一个关键字），于是执行了以下语句查看user的值为当前用户名oracle：

|  |
| --- |
| begin  dbms\_output.put\_line(user);  end; |

初步判断为当前用户下没有执行该语句（DBMS\_SESSION.SET\_CONTEXT(user,str\_key,str\_value)）的权限，于是将该语句拷贝到sql window中进行执行：

|  |
| --- |
| begin  DBMS\_SESSION.SET\_CONTEXT(user,’1’,’2’);  end; |

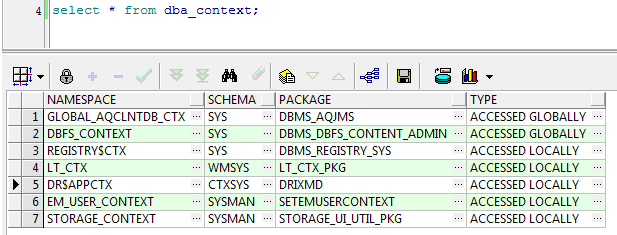


果然报了权限不足的错误，于是上网查找解决方法，在sys用户下执行下列语句授权：

|  |
| --- |
| grant create any context to oracle;  grant create any evaluation context to oracle;  grant alter any evaluation context to oracle;  grant drop any context to oracle;  grant drop any evaluation context to oracle;  grant execute any evaluation context to oracle;  grant create evaluation context to oracle; |

经过一番授权，再次调用，发现还是报没有权限：

最终在网上找到解决方法，说是没有该pkg（pkg\_config）的context，于是查看dba\_context表：



确实没有pkg\_config的上下文配置，于是创建该pkg在当前用户下的pkg配置：

|  |
| --- |
| create or replace context oracle using oracle.pkg\_config; ----locally局部context  或者  create or replace context oracle using oracle.pkg\_config accessed globally;---- globally全局context |

其中第一个oracle为context名，第二个oracle为用户名，pkg\_config为包名。

DBMS\_SESSION.SET\_CONTEXT(user,str\_key,str\_value)中的入参user关键字（这里被使用作为context名）的值即为namespace，用户名即为schema

在添加了该context后调用pkg\_config包的存储过程后，问题解决,并没有再抛出异常；

|  |
| --- |
| declare  str\_var\_value varchar2(5);  begin  pkg\_config.setappcontext('1','2');  select sys\_context(user,'1') into str\_var\_value from dual;  dbms\_output.put\_line(str\_var\_value);  end; |

但是当直接调用dbms\_session.set\_context时依然报无权限：

|  |
| --- |
| declare  str\_var\_value varchar2(5);  begin  dbms\_session.set\_context(user,'1','2');  select sys\_context(user,'1') into str\_var\_value from dual;  dbms\_output.put\_line(str\_var\_value);  end; |

于是查看当前dba\_context表下没有sys用户dbms\_session包的context

于是添加该context

|  |
| --- |
| create or replace context oracle using sys.dbms\_session; |

再次直接调用，可以成功

|  |
| --- |
| declare  str\_var\_value varchar2(5);  begin  dbms\_session.set\_context(user,'1','2');  select sys\_context(user,'1') into str\_var\_value from dual;  dbms\_output.put\_line(str\_var\_value);  end; |

在设置好了该上下文后抛异常的问题解决。

总结：

在有权限的情况下，相应用户的包必须有相应的context才能正确执行。

如在用户oracle下：

1. oracle用户有创建和执行context等的权限。
2. dba\_context表中必须有sys用户的dbms\_session的context（与调用set\_context方法时传入的context名称对应），这时自定义的包pkg\_config也可以调用。注意context名称。

这样既在自己的包中调用dbms\_session.set\_context，也可以直接调用dbms\_session.set\_context。

1. 但是如果有context但是context的schema为当前用户，包为pkg\_config,而没有schema为sys,包为dbms\_session的context，则只能在pkg\_config中调用dbms\_session，而不能直接调用dbms\_session。

延伸：

userenv在我们查看当前用户的session的字符集的时候是经常用到的，例如：

select userenv('language') from dual;

sys\_context的用户和这个userenv类似，是一个保持了和session有关的session级别的上下文。 这个上下文是一个session里的都可以访问到的地方，所以如果我们于需要在这里放入一些session级别的自己的信息，就可以使用这个sys\_context的上下文了。就像在java中的线程变量ThreadLocal,可以把键值对放在线程变量中，在其他的地方通过键来获取相应的值。