oracle 存储过程 utl_file.put_line 将结果输出到文件

参考 ： https://zhidao.baidu.com/question/406381788.html
https://www.2cto.com/database/201411/349979.html
http://blog.sina.com.cn/s/blog_a169ccab0102vpwi.html
http://blog.csdn.net/liqfyiyi/article/details/7043942

如果是在oracle ebs中可以执行fnd_file.put_line(fnd_file.output,'String') 和dbms_output.put_line('String')(一次输出不能超过255个字符);
如果是Form,Report可以用text_io
如果使用oracle DB可以使用dbms_output.put_line和utl_file.put_line（注意它的输出path是固定的，在安装数据库时候已经设定，可以从table中查询到）

法一 ：
eg ：
1.创建路径
要操作文件，就需要有对应的路径，而oracle中使用路径需要用到它的一个结构：directories（路径、地址），具体用法如下：
创建需要操作的路径(ORACLE库所在的服务器)，注意这里的路径文件夹一定要存在，没有的话手工先创建，不然很多地方使用会出异常。路径中最好不要有中文，部分地方使用的时候会出无法找到路径的异常。
eg ：
create or replace directory BLOB_FILE_DIR as '/home/oracle/export';--linux、unix系统路径(mac最底层核心是unix)，BLOB_FILE_DIR是创建的路径名
create or replace directory BLOB_FILE_DIR as 'D:\test';--windows系统路径
2.授权
将该目录和UTL_FILE包授权给所需用户，注意授权一定要做，不然会无法使用。
eg ：
grant read,write on directory BLOB_FILE_DIR to testUesr;--路径授权，添加对路径读、写权限
grant execute on utl_file to testUesr;--utl_file包授权，添加执行权限
3.写出文件
使用utl_file写出文件，通过查询库中内容，写出到指定服务器路径下，总体过程如下：
（1）通过UTL_FILE.FOPEN方法找到对应路径，创建文件，并且给出写入规则。
（2）通过UTL_FILE.PUT_LINE方法向文件中写入内容（UTL_FILE.PUT_LINE写入VARCHAR2类型数据，UTL_FILE.PUT_RAW方法是写入RAW类型的数据，一般来说RAW容量更大，用的更加广泛），这里由于ORACLE有长度限制，一般采用循环方式分批写入。
（3）写入完成后，通过UTL_FILE.FCLOSE方法关闭文件，结束写出。
我们看一下具体操作，这里要创建一个存储过程GET_TEST_BLOB来演示该功能，具体看里面的注释。
eg ：

```
CREATE OR REPLACE PROCEDURE GET_TEST_BLOB(I_ID VARCHAR2) IS
  L_FILE     UTL_FILE.FILE_TYPE;
  L_BUFFER   VARCHAR2(4000);--VARCHAR2最长4000，所以超过的话应该使用循环的方式或者用RAW,最长到32676
  L_FILENAME VARCHAR2(300);
BEGIN
  SELECT F.C_TEXT INTO L_BUFFER FROM TEST_BLOB F WHERE F.C_ID = I_ID;--随意建一个表，包含<span style="font-family:Arial, Helvetica, sans-serif;">C_TEXT、C_NAME、C_ID字段即可</span>
  SELECT F.C_NAME INTO L_FILENAME FROM TEST_BLOB F WHERE F.C_ID = I_ID;
  L_FILE := UTL_FILE.FOPEN('BLOB_FILE_DIR', L_FILENAME, 'w');
  --第三个参数为打开模式，包括'r', 'w', 'a' 'rb', 'wb', 'ab'六种
  --'r'：读文件，一定要保证有该文件，不然会报UTL_FILE.INVALID_PATH异常
  --'w'：写文件，没有该文件的话会自动添加；有的话会覆盖
  --'a'：追加文件，一定要保证有该文件，在已有文件内容后追加内容
  --带有'b'后缀的为使用byte模式，BLOB与VARCHAR2不一样，BLOB打开时一定要用带有'b'后缀的模式
  DBMS_OUTPUT.PUT_LINE('===OPEN OK===' || L_FILENAME || '===' ||
                       LENGTH(L_BUFFER) || '===' || L_BUFFER);
  UTL_FILE.PUT_LINE(L_FILE, L_BUFFER);--写入文件
  DBMS_OUTPUT.PUT_LINE('===EXPORT OK===');
  UTL_FILE.FCLOSE(L_FILE);
```

```
EXCEPTION
  WHEN UTL_FILE.INVALID_PATH THEN--无效的路径
    DBMS_OUTPUT.PUT_LINE('===INVALID_PATH===' || I_ID);
    RAISE;
  WHEN UTL_FILE.INVALID_MODE THEN--无效的打开模式
    DBMS_OUTPUT.PUT_LINE('===INVALID_MODE===' || I_ID);
    RAISE;
  WHEN UTL_FILE.INVALID_OPERATION THEN--无效的操作，文件打开错误会报这个异常，一般来说都是超长或打开方式byte型和非byte型
    DBMS_OUTPUT.PUT_LINE('===INVALID_OPERATION===' || I_ID);
    RAISE;
  WHEN UTL_FILE.INVALID_MAXLINESIZE THEN--无效的最大长度，VARCHAR2最大4000，RAW最大32676，超过回报这个异常，所以一般要进行循环操
作
    DBMS_OUTPUT.PUT_LINE('===INVALID_MAXLINESIZE===' || I_ID);
    RAISE;
  WHEN UTL_FILE.ACCESS_DENIED THEN--拒绝进入指定路径，可能是授权问题
    DBMS_OUTPUT.PUT_LINE('===ACCESS_DENIED===' || I_ID);
    RAISE;
  WHEN UTL_FILE.INVALID_FILEHANDLE THEN--文件处理错误，不常见
    DBMS_OUTPUT.PUT_LINE('===INVALID_FILEHANDLE===' || I_ID);
    RAISE;
  WHEN UTL_FILE.WRITE_ERROR THEN--写入错误，处理该异常最好的方式是将要写入的文件简单化，然后找准错误原因
    DBMS_OUTPUT.PUT_LINE('===WRITE_ERROR===' || I_ID);
    RAISE;
  WHEN NO_DATA_FOUND THEN--SELECT时候未找到数据，不是UTL_FILE的异常
    DBMS_OUTPUT.PUT_LINE('===NO_DATA_FOUND===' || I_ID);
    UTL_FILE.FCLOSE(L_FILE);
    RAISE;
  WHEN OTHERS THEN
    IF UTL_FILE.IS_OPEN(L_FILE) THEN
      UTL_FILE.FCLOSE(L_FILE);
      RAISE;
    END IF;
END GET_TEST_BLOB;
```

传入参数，调用该存储过程。

```
begin
  -- Call the procedure
  get_test_blob('T1');
end;
```

执行后会在对应目录下生成文件

```
eg2 :
DECLARE
l_file utl_file.file_type;
v1 varchar2(32767);
BEGIN

dbms_output.put_line('lalalalalalalalaa');
l_file := utl_file.fopen('FILEPATH','2.txt','R');
LOOP
      UTL_FILE.GET_LINE(l_file,v1);
      dbms_output.put_line('v1: ' || v1);
END LOOP;
utl_file.fclose(l_file);
EXCEPTION
    WHEN NO_DATA_FOUND THEN
    UTL_FILE.FCLOSE(l_file);
END;
/
```

```
DECLARE
l_file utl_file.file_type;

BEGIN
l_file := utl_file.fopen('FILEPATH', '122.txt', 'W');
for i in
(select t.empno||'    '||t.ename||'    '||t.job||'   '||t.mgr||'   '||to_char(t.hiredate,'YYYY-MM-DD')||'   '||t.sal||'
'||t.comm||'   '||t.deptno result  from emp t)
loop
utl_file.put_line(l_file,i.result);
end loop;
utl_file.fclose(l_file);
END;
/
```

法二 ：

SQLPLUS中可以设置spool参数 ：

参考 ： https://www.cnblogs.com/lingbing/p/6386944.html

```
spool abc.txt;                          #将结果输出为abc.txt
select * from wap_subscribe where Telcomcompanyid = '20200' and (Orderdate like '2005%' or Disorderdate like '2005%');
spool off;
```