

PL/SQL 操作 EXCEL 方法比较

Author:jarwang

Since:2008-12-16 19:30

1. 使用 UTL_FILE 包	1
2. 使用 OLE2 接口。	1
3. 使用 ORACLE COM 函数.....	2
4. 使用 JAVA 存储过程（通过 POI 包，这也是本文的重点）	2
5. 使用其他编程语言操作.....	4

引言:

最近 LEADER 安排了一下 JOB,就是要让 EBS 自动发相关的 EXCEL 报表至用户邮箱。经过近一周的前后测试,算是好了。特意写了本文档,希望能对大家有用,能少走一点弯路。PL/SQL 操作 EXCEL 有多种方法,但都不理想。要么就是有操作系统的限制,要么就是实现不了。比较了一下,还是通过 JAVA 的方法比较适用。

1. 使用 UTL_FILE 包

说明:此方法操作文本尚可,对于 EXCEL 这种 COM 文件并不可行。网络大多数所谓的 PL/SQL 操作 EXCEL,也是使用 cvs 这种简单的格式。如果要读入稍微复杂点的 EXCLE 模板文件,utl_file 就挂掉了。

Code:

```
declare
    l_file utl_file.file_type;
BEGIN
    l_file :=utl_file.fopen('U_FIEL','test1.xls','w');
    utl_file.put_line(l_file,'jobs 表导出数据');
    utl_file.fflush(l_file);
end;
```

2. 使用 OLE2 接口。

说明: Object Link and embed,有学过 VB/VC 的同学们应该很熟悉。可惜 ORACLE 本身并不支持 OLE2,OLE2 是 ORACLE FORMS 集成的一个 OLE 接口。所以限制此方法的使用范围。另外更重要的 OLE2 无法在 B/S 架构中使用,只能在传统的 FORM C/S 程序中使用。

Code:

```
declare
    application OLE2.OBJ_TYPE;
    workbooks   OLE2.Obj_Type;
    workbook    OLE2.Obj_Type;
    worksheets OLE2.Obj_Type;
    worksheet   OLE2.Obj_Type;
```

```

        args      OLE2.List_Type;
        cell      OLE2.Obj_Type;

begin
    application:=OLE2.CREATE_OBJ('Excel.Application');
    OLE2.Set_Property(application,'Visible','True');
    workbooks := OLE2.GET_OBJ_PROPERTY(application,'Workbooks');
    workbook:=OLE2.INVOKE_OBJ(workbooks,'Add');
    worksheets := OLE2.GET_OBJ_PROPERTY(workbook, 'Worksheets');
    worksheet:=OLE2.INVOKE_OBJ(worksheets,'Add');
    args:=OLE2.CREATE_ARGLIST;
    OLE2.ADD_ARG(args, 4);
    OLE2.ADD_ARG(args, 2);
    cell := OLE2.GET_OBJ_PROPERTY(worksheet,'CELLS',args);
    OLE2.DESTROY_ARGLIST(args);
    OLE2.Set_Property(cell, 'Value', 'Hello Excel!');
    args:=OLE2.CREATE_ARGLIST;
    OLE2.ADD_ARG(args, 5);
    OLE2.ADD_ARG(args, 3);
    cell := OLE2.GET_OBJ_PROPERTY(worksheet,'CELLS',args);
    OLE2.DESTROY_ARGLIST(args);
    OLE2.Set_Property(cell, 'Value', 'Hello Excel!');

exception
    when others then
        FND_MESSAGE.DEBUG('ERROR:'||SQLERRM);

end;
```

3. 使用 ORACLE COM 函数

说明: 这种方法就与通过 extproc 来实现与 excle 通信, 这种方法直接与 window api 打交道。大致的代码是:

```
ordcom.CreateObject('Excel.Application', 0, servername,applicationToken);
```

测试了一下, 似乎只能要求 ORACLE 的宿主系统必须是 window。这也就有严重的局限性了。

4. 使用 JAVA 存储过程 (通过 POI 包, 这也是本文的重点)

4.1 至 <http://poi.apache.org> 下载 POI 包, 注意与 ORACLE JVM 版本匹配 (旧版本的 POI 下载: <http://archive.apache.org/dist/jakarta/poi/release/bin/>)。至 ORACL 官网下载 JDBC 驱动, 这一步不是必须的, 一般安装 ORACLE 时, 都有内置了 JDBC 包。

4.2 在 ORACLE 注册 POI。

4.2.1 先将 POI 包上传至 ORACLE 服务器。目录为: \$ORACLE_HOME/javavm/lib , 例: /data/book/bookdb/9.2.0/javavm/lib, 必须是 javavm/lib 这个目录或其下层目录。

4.2.2 使用 DBMS_JAVA 注册

```

call dbms_java.loadjava('-r -v -definer -g
public /data/book/bookdb/9.2.0/javavm/lib/poi-2.5.1-final-20040804.jar');
call dbms_java.loadjava('-r -v -definer -g
public /data/book/bookdb/9.2.0/javavm/lib/poi-contrib-2.5.1-final-20040804.jar');
call dbms_java.loadjava('-r -v -definer -g
public /data/book/bookdb/9.2.0/javavm/lib/poi-scratchpad-2.5.1-final-20040804.jar');

```

注：这个问题折腾了我很久，注册 JAR 包有两种方法。1.使用 DBMS_JAVA，这种方法要自行上传 jar 包到服务器上。2.是使用 loadjava。这也是能 google 到最多的方法，但 loadjava 在 ORACLE9i 以后就不单独提供了，而是集成在 JDeveloper 中，而可恶的 JDeveloper 是有版本兼容性的问题。

4.2.3 查询一下是否注册成功（注：如果注册成功，STATUS 应该为 VALID）。

```

Select OBJECT_NAME,
       OBJECT_TYPE,
       STATUS,
       SYS.DBMS_JAVA.LONGNAME(OBJECT_NAME) LONGNAME,
       LAST_DDL_TIME,
       OBJECT_ID,
       uo.created
from sys.user_objects uo
where object_type in
      ('JAVA CLASS', 'JAVA SOURCE', 'JAVA RESOURCE', 'JAVA DATA')
and uo.object_name like '%poi%'

```

4.3 创建 JAVA SOURCE

```

create or replace and compile java source named testjdbc as
import org.apache.poi.hssf.usermodel.HSSFWorkbook;
import org.apache.poi.hssf.usermodel.HSSFSheet;
import org.apache.poi.hssf.usermodel.*;
import java.io.*;
public class TestJDBC{
    public static void exportExcel(){
        try{
            HSSFWorkbook wb = new HSSFWorkbook();
            HSSFSheet sheet = wb.createSheet("new sheet");
            wb.setSheetName( 0,
                           "SomeUnicodeName",
                           HSSFWorkbook.ENCODING_UTF_16 );
            HSSFRow r = sheet.createRow( 0 );
            HSSFCell c = r.createCell( (short)0 );
            c.setCellValue("X 中国");
            // Write the output to a file
            FileOutputStream fileOut = new
FileOutputStream("/data/book/k.xls");
            wb.write(fileOut);

```

```
        fileOut.close();
    } catch (Exception e) {
        System.out.println("poi error :"+e);
        e.printStackTrace();
    }
}
```

4.4 创建调用的 PL/SQL 包

```
create or replace package TestJDBC is
    procedure exportExcel;
    procedure insert_table(t varchar2);
    function query_table return varchar2;
end TestJDBC;

create or replace package body TestJDBC is
    procedure exportExcel is
    language java
        name 'TestJDBC.exportExcel()';
end TestJDBC;
```

4.5 测试 PL/SQL 块

```
begin
    testjdbc.exportExcel;
end;
```

5. 使用其他编程语言操作

说明：此方法完全脱离了 ORACLE，已超出了 ORACLE 的范围，本文不讨论。