|  |
| --- |
| **How to load Excel spreadsheet data into Oracle tables**  *Oracle Database Tips by Donald Burleson*  *April 16, 2016* |

**Question**: I want to load Excel spreadsheet data into Oracle.  I know that I can convert the spreadsheet into a CSV and load the spreadsheet as an Oracle external table, but I want a method for loading the Excel data directly into an Oracle table.  What's the best way to load spreadsheet data into Oracle?

**Answer**:  As you noted, there are several approaches for getting Excel data into Oracle:

* **Load data directly from the spreadsheet** - The best way to do this fast is with the SQL Developer product. You can load directly from within the Excel spreadsheet without the clumsy ODBC, very fast using a native API.  It's also great for end-users since it adheres to all Oracle security access policies.
* **Define the spreadsheet as an Oracle external table** - If you convert the spreadsheet from an xls file to a csv (comma-delimited) format, you can define the spreadsheet as an [external table](http://www.dba-oracle.com/art_ext_tabs_spreadsheet.htm).
* **Load Excel Spreadsheet from SQL\*Forms** - [This thread](http://dbaforums.org/oracle/index.php?showtopic=4704) shows some code for loading data from a spreadsheet directly into an Oracle table:

DECLARE   
application OLE2.OBJ\_TYPE;   
workbooks OLE2.OBJ\_TYPE;   
workbook OLE2.OBJ\_TYPE;   
worksheets OLE2.OBJ\_TYPE;   
worksheet OLE2.OBJ\_TYPE;   
cell OLE2.OBJ\_TYPE;   
args OLE2.OBJ\_TYPE;   
ctr NUMBER(12);   
cols NUMBER(2);  
name\_var1 VARCHAR2(2000);   
name\_var2 VARCHAR2(2000);   
name\_var3 VARCHAR2(2000);   
name\_var4 VARCHAR2(2000);   
filename varchar2(100);  
  
PROCEDURE OLEARG IS  
args OLE2.OBJ\_TYPE;   
BEGIN  
args := OLE2.CREATE\_ARGLIST;   
ole2.add\_arg(args,ctr); -- row value  
ole2.add\_arg(args,cols); -- column value  
cell := ole2.GET\_OBJ\_PROPERTY(worksheet,'Cells',args); -- initializing cell  
ole2.destroy\_arglist(args);   
END;  
  
BEGIN  
filename :=GET\_FILE\_NAME('c:\', File\_Filter=>'Excel Files (\*.xls)|\*.xls|'); -- to pick the file  
application := OLE2.CREATE\_OBJ('Excel.Application');   
ole2.set\_property(application,'Visible','true');   
workbooks := OLE2.GET\_OBJ\_PROPERTY(application, 'Workbooks');   
args := OLE2.CREATE\_ARGLIST;   
ole2.add\_arg(args,filename); --'c:\13s002.xls'); -- file path and name  
workbook := ole2.GET\_OBJ\_PROPERTY(workbooks,'Open',args);   
ole2.destroy\_arglist(args);   
args := OLE2.CREATE\_ARGLIST;   
ole2.add\_arg(args,'Sheet1');   
worksheet := ole2.GET\_OBJ\_PROPERTY(workbook,'Worksheets',args);   
ole2.destroy\_arglist(args);   
  
ctr := 2; --row number  
cols := 1; -- column number  
  
FIRST\_RECORD;   
  
LOOP   
OLEARG;  
name\_var1 := ole2.get\_char\_property(cell,'Value'); --cell value of the argument  
cols := cols+1;  
  
OLEARG;  
name\_var2 := ole2.get\_char\_property(cell,'Value'); --cell value of the argument  
cols := cols+1;  
  
OLEARG;  
name\_var3 := ole2.get\_char\_property(cell,'Value'); --cell value of the argument  
cols := cols+1;  
  
OLEARG;  
name\_var4 := ole2.get\_num\_property(cell,'Value'); --cell number value of the argument  
  
IF ctr = 1 then   
:tf1 := name\_var1;   
:tf2 := name\_var2;   
:tf3 := name\_var3;   
:tf4 := name\_var4;   
ELSE   
NEXT\_RECORD;   
:tf1 := name\_var1;   
:tf2 := name\_var2;   
:tf3 := name\_var3;   
:tf4 := name\_var4;  
END IF;   
  
EXIT WHEN length(name\_var1) = 0 or length(name\_var1) is null;   
ctr := ctr + 1;   
cols := 1;  
  
END LOOP;   
OLE2.INVOKE(application,'Quit');   
OLE2.RELEASE\_OBJ(cell);   
OLE2.RELEASE\_OBJ(worksheet);   
OLE2.RELEASE\_OBJ(worksheets);   
OLE2.RELEASE\_OBJ(workbook);   
OLE2.RELEASE\_OBJ(workbooks);   
OLE2.RELEASE\_OBJ(application);   
END;

One of the most common problems with extracting Oracle data into spreadsheets are the cumbersome interfaces and slow performance of ODBC. The easiest way to export data into a format that can be read by Excel is to spool SQL\*Plus output to a file and select the data in comma delimited form.  This will effectively create a CSV file that can be opened by excel:

set pages 0  
  
set lines 120  
  
spool myexcel.csv  
  
select  
   col1||','||col2||','col3  
from mytable;  
  
spool off

This simple example shows how to get a CSV file for a spreadsheet from an Oracle table.