

User's Guide and Reference Release 11.0.5 *Februari 2021* 



# Contents

General information	. 3
Introduction SQLWord architecture Software requirements Installation Upgrade from previous release	.3 .4 .5 .6
License key	. 8
SQLWord Developer	.9
Introduction       1         Scriptlets       1         Hints       1         Examples       1         Steps to create a source document       3         Options       3	.9 11 13 13 13 35
SQLWord Run	38
Introduction	38 40
SQLExcel	12
How to generate Microsoft Excel XSLX4	12
Apex integration	15
Installation4 Implementation explained4	45 48
PDF output (optional)	54
Installation on Unix Oracle Server       5         Installation on Microsoft Windows Server       5         Apex implementation PDF expained       5	54 56 58
Frequently asked questions	50
How can I create new pages in the output document?	50 50 51 52 52
Hints & Tips	54
Compile multiple documents	54 55 55

## **General information**

## Introduction

All Oracle database users know that it is quite difficult to retrieve Oracle data into Microsoft Word documents. Sequel Services has developed a report generator which gives you a powerful tool to solve this problem.

You can create your source documents in Microsoft Word and use SQLWord to generate Microsoft Word documents merged with data from your Oracle database.

The existing reporting tools mostly use their own specific format and don't integrate at all with Microsoft Word documents. Mailmerge in Microsoft Word has very limited possibilities and is difficult to use. It is not possible to create master-detail documents.

When using SQLWord you can create your own standard letters, contracts and reports, integrating with the data of your Oracle database.

Using the SQLWord Developer application, you can place several PL/SQL-statements enclosed by <% tag %> scriptlets inside the text of a Microsoft Word document. SQLWord follows the syntax of Oracle PSP (PL/SQL Server Pages) <% tag %> declarations.

Microsoft Word documents with <% tag %> scriptlets can be stored by the SQLWord Developer application into your Oracle database, compiled as PL/SQL-procedures. By calling the PL/SQL procedure that you created from your Microsoft Word document, SQLWord retrieves the data from your Oracle database and integrates it in the generated output document.

If you run SQLWord "client-server" by using the SQLWord Run application, the output document is created on the LAN and is opened with Microsoft Word.

If you run SQLWord from an Apex application, the output document is send to the webbrowser on the client, where the output document is opened with Microsoft Word. An Apex5 demo implementation example is included.

SQLWord uses the **DOCX format**. This format is internal based on XML.

## SQLWord architecture



## Software requirements

### Server side

• Oracle 11g / Oracle 12c

### Client

- Windows7, Windows8, Windows10. SQLWord can also run as a 32-bits application on Windows 64-bit OS.
- Microsoft Word 2016 / 2013 / 2010 / 2007 / 2003. For Microsoft Word 2003 you need to install the Microsoft Office Compatibility Pack 2007 for supporting the DOCX-format.
- 32-bits Oracle Client (11g / 12c) for Microsoft Windows. SQLWord is a 32-bits application and <u>cannot</u> connect to an 64-bit Oracle client.

## Installation

## Setup

From the file manager double-click on the file named <u>install\_sqlword11.exe</u> or <u>install\_sqlword11\_eval.exe</u> to start the setup program. Please follow the on-screen installation directions.

All necessary files will be installed by default in a folder at C:\SQLWord11



#### Create SQLWord tables and the SQLWord packages

Before installing the SQLWord tables and package we suggest to create a new user SQLWORD\_DEMO for evaluation purposes with the good old SQL\*Plus:

```
SQL> connect to a user with DBA-rights ...
SQL> create user sqlword_demo identified by sqlword_demo
default tablespace USERS;
SQL> grant connect, resource to sqlword_demo;
SQL> grant create view to sqlword demo;
```

Create a <u>shortcut</u> on your desktop for SQL\*Plus and specify the default directory <u>Start in:</u> C:\SQLWord11\SQL

Target location:	: bin
Target:	C:\oraclexe\app\oracle\product\11.2.0\server\b
Start in:	C:\SQLWord11\SQL
Shortcut key:	None
Run:	Normal window
Comment:	
Find	Target Change Icon Advanced

If your Oracle database is **XE** then first you must grant several sys-privileges to the SQLWORD\_DEMO user. Start SQL\*Plus from this shortcut and run script sys\_grants.sql (as sysdba).

SQL> @sys\_grants.sql

Grant succeeded. Grant succeeded. Grant succeeded. Grant succeeded. Grant succeeded.

- Now start SQL\*Plus from this shortcut, connect to user SQLWORD\_DEMO and run the installation script install\_sqlword11.sql
  - SQL> connect sqlword\_demo/sqlword\_demo
  - SQL> @install\_sqlword11.sql

Table created. Index created. Table altered. Table created. Index created. Table altered. Table altered. Table created. Index created. Table altered. Table altered. Table created. Index created. Table altered. Package created. Package body created. etc ...

When the installation script is finished check the log-files **install\_sqlword11.log** and **install\_sqlword11\_demo.log** for any errors.

### Upgrade from previous release

.

If you already have installed a previous SQLWord 11 release then you only need to update the sqlword package.

Start SQL\*Plus, connect to user SQLWORD\_DEMO and run the upgrade script upgrade\_sqlword11.sql

SQL> connect sqlword\_demo/sqlword\_demo
SQL> @upgrade\_sqlword11.sql
Upgrading SQLWord 11.0.4.0 ...
Package created.
Package-body created.

## License key

To install the SQLWord license key you need to run the supplied license script **sqlword\_license.sql** using SQL\*Plus.

SQL> @sqlword\_license.sql

The SQLWord license key is inserted into table SQLWORD\_PARAMETER and is verified every time you run SQLWord.

You can display the license information with SQL\*Plus:

SQL> select sqlword.show\_license from dual;

SHOW\_LICENSE

SQLWord is licensed to **<company-name>** for **<number>** users on Oracle database server **<oracle-database-server-name>**.

## **SQLWord Developer**

## Introduction

SQLWord Developer is a 32-bits Windows application to support users in the development of Microsoft Word template documents.

The SQLWord Developer application window is always displayed <u>on top</u> of all other applications so you can edit Microsoft Word documents and always have access to the SQLWord Developer application.

The SQLWord Developer application contains a button toolbar and a work area where you can edit <% tag %> scriptlets with PL/SQL-statements.

SQLWord <sup>11</sup> - X									
🧐 🚅 🚍 ∽   Q 🎼 X & 🥑   ✔ 🗐 🕃   [ 	]								

## Toolbar buttons:

<b>\$</b>	Shows a submenu where you can:
Connect	<ul> <li>Connect to your Oracle database.</li> </ul>
Options	<ul> <li>Display the options-window (described later in this section).</li> </ul>
Help	<ul> <li>Display this Users Guide &amp; Reference.</li> </ul>
About	• Show the about box.
w	Create a new Word document.
	Open a Word document. If you select <u>multiple</u> files you will get a new screen where you can compile the selected files in <u>one run</u> .
-	Save the active Word document.
5	Undo the last Word command.
Q	Find <% tag %> scriptlets in the active Word document and copy to the work area.
	Paste the <% tag %> scriptlets from the work area into the active Word document.
×	Clear the work area.
C	Insert standard <% tag %> scriptlets into the work area from a submenu.
<b>V</b>	Clear all <% tag %> scriptlets in the active Word document from invisible formatting code.
1	Compile the active Word document to a stored procedure.
	Show the stored procedure source code from the active Word document and edit several parameters (descriptions & lookup SQL-statements).



Remove stored procedures & content from the Oracle database.

Run SQLWord. The screen below shows up where you can select a report and specify values for file locations. When pressing on the <u>Run</u> button the screen below shows up where you can specify the input-parameters.

un		
Procedure EXAMPLE1A - example1a	Show output	
File locations		
Outfile	Read Only	
Parfile		
Logfile	<b>F</b>	
	Bun Clos	se

## Popup-menu

When pressing on the right mouse button in the work area a popup-menu appears:



## Scriptlets

SQLWord follows the syntax of Oracle PSP (PL/SQL Server Pages) <% tag %> declarations.

Scripting tags are enclosed within the <% and %> delimiters and the <u>first</u> character(s) after the opening delimiter <% determine the <u>type</u> of the scripting tag.

The following describe each scripting tag in detail.

```
Declarations <%! {plsql_declaration} %>
```

The declaration tag can be used to declare types, cursors and also define local procedures as well. Note that you need the <u>!</u> sign in this syntax.

Example: Declaring variables.

```
<%!
--
v1 number;
v2 varchar2(10) := '1234567890';
--
%>
```

Example: Declaring a cursor c1.

```
<%!
--
cursor cl
is
select emp.first_name || ' ' || emp.last_name as employee
, emp.salary
, job.job_title
from employees emp
, jobs job
where emp.job_id = job.job_id
order by emp.last_name;
--
%>
```

#### Statements <% {plsql\_statement} %>

All PL/SQL statements can be used such as for loops, assignments, calls to other stored procedures, etc. Note that a <u>terminating semicolon</u> is needed where PL/SQL requires it.

Example: for loop.

<%for r1 in c1 loop%> <%end loop;%>

NB: Always place every loop statement on a new line !!!

and <

Example: assignments.

```
<%
--
a := 'ABC' || 'DEF';
b := a || 'GHI';
c := my_procedure(a, b);
--
%>
```

Example: local block.

```
<%
--
declare
    a varchar2(10) := 'ABCDEF';
    b varchar2(10);
begin
    b := a || 'GHI';
end;
--
%>
```

Example: exception handler.

<% exception when NO\_DATA\_FOUND then null; %>

### Expressions <%= {plsql\_expression} %>

The expression tag <u>returns the value of any PL/SQL</u> expression including PL/SQL function calls and places the return value into the output document. Note that a terminating semicolon is not allowed.

Example:

```
<%= 10 + 2 %>
<%= 'ABC' || 'EFG' %>
<%= to_char(sysdate, 'dd.mm.yyyy hh24:mi:ss') %>
<%= r1.job%>
```

### Example:

```
<%!
___
cursor cl
is
select emp.first_name || ' ' || emp.last_name as employee
      emp.salary
,
       job.job_title
from employees emp
      jobs job
where emp.job_id = job.job_id
order by emp.last name;
___
응>
<%for r1 in c1 loop%>
                    <%= r1.salary%> <%= r1.job title%>
<%= r1.employee%>
<%end loop;%>
```

### Parameters <%@plsql parameter={ ... } %>

The parameter tag declares an input parameter to the document:

parameter="<name>" [ type="( varchar2 | number | date )" ] [ default="<default\_value>" ]

The VARCHAR2 is default type. Default text values must be enclosed within single quotes within the double quotes, eg default="xyz".

Example:

```
<%0 plsql parameter="P_EMPLOYEE_ID" type="number"%>
<%0 plsql parameter="P_LAST NAME" default="'KING'" %>
```

### Include <%include file={filename}%>

The include tag can be used to include PLSQL declarations from an external file:

file={filename}

Example:

<%@ include file="example1b.plsql"%>

### Hints

### **#SQLWORD FORMAT**

You can add the #SQLWORD\_FORMAT hint in select statements to inform SQLWord that you want to add extra XML formatting code (ie to set text bold or set colors, etc).

This option is meant for advanced MSWord users who are well known with the Microsoft Word XML formatting language. For more information about XML formatting check this nice website: <u>http://www.lenzconsulting.com/wordml</u>

#### Example:

```
select initcap(emp.first_name || ' ' || emp.last_name) as employee
       case
         when emp.salary > 3000 then
         '#SQLWORD FORMAT</w:t><w:rPr><w:i/><w:color
         w:val="FF0000"/></w:rPr><w:t>'
                                         trim(to_char(emp.salary, 'L999G999G999')) ||
          '</w:t></w:r><w:r><w:t>'
         else
           trim(to_char(emp.salary, 'L999G999G999'))
       end as salary
       job.job title
from
       employees emp
       jobs job
where
      emp.manager id = p manager id
and
       emp.job_id = job.job_id
order by emp.last_name;
```

Examine example1d.docx to see how it works.

## Examples

You can find several SQLWord templates at: C:\Program Files\Sqlword11\Examples\Docx

Example1a.docx Example1b.docx Example1c.docx Example1d.docx Example2.docx Example3.docx Example4.docx Example5.docx Example6.docx HR\_Dept.docx HR\_Dept\_Selection.docx HR\_Employee\_Contract.docx HR\_Employee\_Job\_Offer.docx HR\_Employee\_Selection.docx HR\_Manager.docx HR\_Tables\_Report.docx

Redwood, <%=to char(sysdate,'fm dd month yyyy')%> <%for r1 in c1(p\_employee\_id) loop%> Dear <%=r1.manager%>, We inform you about the current salary of your employees: Employee Job Salary <%for r2 in c2 (r1.manager\_id) <%=r2.job\_title%> <%=r2.salary% loop%><%=r2.employee%> <%end loop;%> Sincerely, Larry Ellison <%end loop;%> <%@ plsql parameter="P EMPLOYEE ID" "type="number"%> < % ! \_\_\_ cursor c1 (p\_employee\_id number) is select p\_employee\_id as manager\_id
, initcap(first\_name || ' ' || last\_name) as manager to char(sysdate, 'dd month yyyy') today from employees where employee\_id = p\_employee\_id; cursor c2 (p\_manager\_id number) is select initcap(emp.first name || ' ' || emp.last name) as employee trim(to\_char(emp.salary, 'L999G999G999')) , as salarv job.job\_title from employees emp jobs job where emp.manager\_id = p\_manager\_id emp.job\_id = job.job\_id and order by emp.last\_name;

Example1a.docx shows how to generate a letter by using <% tags %> described in the previous section. It uses the following tags:

Declaration-tag:

\_\_\_

```
<응!
___
cursor c1 (p_employee_id number)
is
select p employee id as manager id
      initcap(first name || ' ' || last name) as manager
.
      to char(sysdate,'dd month yyyy') today
from employees
where employee_id = p_employee_id;
cursor c2 (p_manager_id number)
is
select initcap(emp.first_name || ' ' || emp.last_name) as employee
      trim(to_char(emp.salary, 'L999G999G999'))
                                                 as salary
,
      job.job title
     employees emp
from
      jobs job
where emp.manager id = p manager id
and
     emp.job id = job.job id
order by emp.last name;
___
응>
```

• Parameter-tag:

<%@ plsql parameter="P\_EMPLOYEE\_ID" "type="number"%>

### <u>Statement-tags</u>:

<%for r1 in c1(p\_employee\_id) loop%>
<%for r2 in c2 (r1.manager\_id) loop%><%=r2.employee%>
<%end loop;%>

### <u>Assignment-tags</u>:

<%=to\_char(sysdate,'fm dd month yyyy')%>
<%=r1.manager%>
<%=r2.employee%>
<%=r2.job\_title%>
<%=r2.salary%>

## Compiling

Dear Steven King,



When pressing the compile button SQLWord Developer creates a stored procedure with the name EXAMPLE1A. Examine the PL/SQL source code on the next page to see how the <% tags %> are placed.

Employee		Salary
Gerald Cambrault	Sales Manager	€11.000
Lex De Haan	Administration Vice President	€17.000
Alberto Errazuriz	Sales Manager	€12.000
Adam Fripp	Stock Manager	€8.200
Michael Hartstein	Marketing Manager	€13.000
Payam Kaufling	Stock Manager	€7.900
Neena Kochhar	Administration Vice President	€17.000
Kevin Mourgos	Stock Manager	€5.800
Karen Partners	Sales Manager	€13.500
Den Raphaely	Purchasing Manager	€11.000
John Russell	Sales Manager	€14.000
Shanta Vollman	Stock Manager	€6.500
Matthew Weiss	Stock Manager	€8.000
Eleni Zlotkey	Sales Manager	€10.500

Sincerely,

Larry Ellison

*Output document generated from example1a.docx* 

Redwood, 3 januari 2014

```
CREATE OR REPLACE PROCEDURE EXAMPLE1A
      p employee id number)
(
is
cursor c1 (p_employee_id number)
is
select p_employee_id as manager_id
, initcap(first name || ' ' || last name) as manager
       to char(sysdate,'dd month yyyy') today
from
       employees
where employee_id = p_employee_id;
cursor c2 (p manager id number)
is
select initcap(emp.first_name || ' ' || emp.last_name) as employee
, trim(to_char(emp.salary, 'L999G999G999')) as salary
       job.job title
from employees emp
        jobs job
where emp.manager_id = p_manager_id
and
       emp.job id = job.job id
order by emp.last_name;
___
BEGIN
if sqlword.init report('EXAMPLE1A') then
sqlword.put_content('EXAMPLE1A',1);
for r1 in c1(p_employee_id) loop
sqlword.put_content('EXAMPLE1A',2);
sqlword.put_content('EXAMPLE1A',3);
sqlword.put_content('EXAMPLE1A',4);
sqlword.put_data(r1.manager);
sqlword.put content('EXAMPLE1A',5);
sqlword.put_content('EXAMPLE1A', 6);
for r2 in c2 (r1.manager id) loop
sqlword.put content('EXAMPLE1A',7);
sqlword.put_data(r2.employee);
sqlword.put_content('EXAMPLE1A',8);
sqlword.put data(r2.job title);
sqlword.put content ('EXAMPLE1A',9);
sqlword.put_data(r2.salary);
sqlword.put_content('EXAMPLE1A',10);
end loop;
sqlword.put content('EXAMPLE1A',13);
end loop;
sqlword.put_content('EXAMPLE1A',16);
sqlword.put_content('EXAMPLE1A',17);
sqlword.put_content('EXAMPLE1A',18);
sqlword.put_content('EXAMPLE1A',19);
sqlword.put content('EXAMPLE1A',20);
sqlword.merge_xml('word/header1.xml');
sqlword.put_content('EXAMPLE1A',21);
sqlword.put data(to char(sysdate,'fm dd month yyyy'));
sqlword.put content('EXAMPLE1A',22);
sqlword.put_content('EXAMPLE1A',23);
end if;
sqlword.end report;
```

Generated stored procedure for example1a.docx

	Redwood	d, <%=to_char(sysdate,'fm dd month yyyy')%	<i>6</i> >
<%for r1 in c1(p_employee_id) loop% Dear <%=r1.manager%>,	6>		
We inform you about the current salar	y of your employees:		
Employee	Job	Salary	
<% for r2 in c2 (r1.manager id)	<%=r2.job title%>	<%=r2.salary%>	
loop%><%=r2.employee%>			
<%end loop;%>			
Sincerely.			
,			
Larry Ellison			
<%endloop;%> <%@ plsql parameter="P_EMM <%@ include file="example:	PLOYEE_ID" "type="number"% lb.plsql"%>	%>	

Example1b.docx is about the same as example1a.docx with the difference that the PL/SQL declarations are included by an <u>external</u> file. In this way you can edit large PL/SQL declarations much easier.

include-tag:

<%@ include file="example1b.plsql"%>

Redwood, <%=to_char(sysdate,'fm dd month yyy										
<%for r1 in hr_cursors.c_mgr(p_emp Dear <%=r1.manager%>,	loyee_id) loop%>									
We inform you about the current sala	ry of your employees:									
Employee	Job	Salary								
<pre>&lt;%for r2 in hr_cursors.c_emp (r1.manager_id) loop%&gt;&lt;%=r2.employee%&gt; &lt;%end loop;%&gt;</pre>	<%=r2.job_title%>	<%=r2.salary%>								
Sincerely, Larry Ellison										
<%endloop;%> <%@ plsql parameter="P_EM	PLOYEE_ID" "type="number	·"%>								

Example1c.docx is about the same as example1a.docx with the difference that the PL/SQL declarations are included by a <u>reference</u> to the cursor declarations in package specification HR\_CURSORS. In this way you can keep controll of all your SQLWord SQL statements and modify the cursors quickly in case of database changes.

### <u>Statement-tags</u>:

```
<%for r1 in hr_cursors.c_mgr(p_employee_id) loop%>
<%for r2 in hr_cursors.c_emp (r1.manager_id) loop%>
<%end loop;%>
```

```
CREATE OR REPLACE PACKAGE HR CURSORS
IS
___
cursor c_mgr (p_employee_id number)
is
select p_employee_id as manager_id
, initcap(first_name || ' ' || last_name) as manager
       to char(sysdate, 'dd month yyyy') today
from employees
where employee id = p employee id;
--
cursor c_emp (p_manager_id number)
is
select initcap(emp.first_name || ' ' || emp.last_name) as employee
       trim(to char(emp.salary, 'L999G999G999'))
                                                           as salary
,
       job.job_title
from
      employees emp
       jobs job
,
where emp.manager_id = p_manager_id
       emp.job id = job.job id
and
order by emp.last name;
_ _
```

Redwood, <%=to\_char(sysdate,'fm dd month yyyy')%> <% for r1 in c1(p employee id) loop%> Dear <%=r1.manager%>, We inform you about the current salary of your employees: Job Employee Salary <%=r2.salary%> <% for r2 in c2 (r1.manager id) <%=r2.job title%> loop%><%=r2.employee%> <%end loop;%> Sincerely, Larry Ellison <%end loop;%> <%@ plsql parameter="P EMPLOYEE ID" type="number"%> < %! cursor c1 (p\_employee\_id number) is select p employee id as manager id '#SQLWORD FORMAT</w:t><w:rPr><w:b/></w:rPr><w:t>'|| chr(38) || '#160;' || , first\_name || chr(38) || '#160;' || '</w:t></w:r><w:t>' || last name as manager to char(sysdate, 'dd month yyyy') today from employees where employee id = p employee id; \_\_\_ cursor c2 (p\_manager\_id number) is select initcap(emp.first\_name || ' ' || emp.last\_name) as employee case , when emp.salary > 3000 then '#SQLWORD FORMAT</w:t><w:rPr><w:i/><w:color w:val="FF0000"/></w:rPr><w:t>' trim(to\_char(emp.salary, 'L999G999G999')) || '</w:t></w:r><w:r><w:t>' else trim(to char(emp.salary, 'L999G999G999')) end as salary job.job title employees emp from jobs job , where emp.manager\_id = p\_manager\_id and emp.job\_id = job.job\_id order by emp.last\_name;

Example1d.docx is about the same as example1a.docx with the difference that extra XML format code is added by using the **#SQLWORD\_FORMAT** hint. In this example the first name is formatted to **bold** and the salary color is set to red when greater than 3000.

For more information about XML formatting check this nice website: http://www.lenzconsulting.com/wordml

## Dear Adam Fripp,

We inform you about the current salary of your employees:

Employee	Job	Salary
Mozhe Atkinson	Stock Clerk	€2.800
Laura Bissot	Stock Clerk	€3.300
Alexis Bull	Shipping Clerk	€4.100
Anthony Cabrio	Shipping Clerk	€3.000
Julia Dellinger	Shipping Clerk	€3.400
James Marlow	Stock Clerk	€2.500
Tj Olson	Stock Clerk	€2.100
Nandita Sarchand	Shipping Clerk	€4.200

Sincerely,

Larry Ellison

Output document generated from example1d.docx

<%open c emp; fetch c emp into r emp;%> Dear Mr./Ms. <%=r emp.last name%>, Human Resources Inc. is pleased to offer you the position of <%=r\_emp\_job\_title%>. Your skills and experience will be an ideal fit for our <%=r emp.department%> department. As we discussed, your starting date will be <%=r\_emp.hire\_date%>. The salary scale for this job ranges from <%=r\_emp.min\_salary%> to <%=r\_emp.max\_salary%> per month. The salary is <%=r emp.year salary%> per year and is paid on a monthly basis. Direct deposit is available. Full family medical coverage will be provided through our company's employee benefit plan. Dental and optical insurance are also available. Human Resource Inc. offers a flexible paid-time off plan which includes vacation, personal, and sick leave. Time off accrues at the rate of one day per month for your first year, then increases based on your tenure with the company. We look forward to welcoming you to the Human Resource Inc. team. Please let me know if you have any questions or I can provide any additional information. Sincerely, <%=r\_emp.manager%> <%=r\_emp.mgr\_job\_title%>, department <%=r\_emp.department%> Human Resource Inc. <%close c emp;%> <%@ plsql parameter="P\_EMPLOYEE\_ID" "type="number"%> <응! cursor c emp is select emp.last name trim(to char(emp.hire date,'dd') || ' ' || , trim(to\_char(emp.hire\_date, 'month')) || ' ' || to\_char(emp.hire\_date,'yyyy')) as hire\_date trim(to\_char(emp.salary \* 12 , 'L999G999G999')) as year salary job1.job\_title as emp\_job\_title , trim(to\_char(job1.min\_salary, 'L999G999G999')) as min salary , trim(to\_char(job1.max\_salary, 'L999G999G999')) as max\_salary , trim(mgr.first\_name || ' ' || mgr.last\_name) as manager , job2.job title as mgr job title , dept.department\_name as department , loc.street\_address || ' ' || loc.city as department\_address , trim(to char(sysdate,'dd') || ' ' || , trim(to\_char(sysdate, 'month')) || ' ' || to char(sysdate,'yyyy')) as today from employees emp employees mgr , departments dept , jobs job1 , jobs job2 , locations loc emp.employee\_id = P\_EMPLOYEE\_ID where emp.job id = job1.job id and mgr.job id = job2.job id and and emp.department\_id = dept.department\_id (+) emp.manager\_id = mgr.employee\_id (+) and dept.location id = loc.location id (+); and \_\_\_ r emp c emp%rowtype; 응>

Example2.docx shows how to generate a job offer document for an employee.

<u>Declaration-tag:</u>

```
< %!
___
cursor c emp
is
select emp.last name
        trim(to char(emp.hire date,'dd') || ' ' ||
1
        trim(to char(emp.hire date, 'month')) || ' ' ||
        to_char(emp.hire_date,'yyyy')) as hire_date
        trim(to_char(emp.salary * 12 , 'L999G999G999'))
,
        as year_salary
        job1.job_title as emp_job_title
,
       trim(to_char(job1.min_salary, 'L999G999G999')) as min_salary
trim(to_char(job1.max_salary, 'L999G999G999')) as max_salary
,
       trim(mgr.first_name || ' ' || mgr.last_name) as manager
        job2.job_title as mgr_job_title
        dept.department name as department
        loc.street address || ' ' || loc.city as department address
,
        trim(to_char(sysdate,'dd') || ' ' ||
        trim(to_char(sysdate, 'month')) || ' ' ||
        to_char(sysdate,'yyyy')) as today
from employees emp
        employees mgr
1
       departments dept
,
       jobs job1
1
       jobs job2
1
        locations loc
where emp.employee_id = P_EMPLOYEE_ID
and emp.job_id = job1.job_id
and mgr.job_id = job2.job_id
and emp.department_id = dept.department_id (+)
and emp.manager_id = mgr.employee_id (+)
and dept.location_id = loc_location_id (
and
       dept.location id = loc.location id (+);
r_emp c_emp%rowtype;
응>
```

• <u>Parameter-tag:</u>

<%@ plsql parameter="P EMPLOYEE ID" "type="number"%>

• <u>Statement-tags</u>:

<%open c\_emp; fetch c\_emp into r\_emp;%>

<%close c\_emp;%>

<u>Assignment-tags</u>:

```
<%=r_emp.last_name%>
<%=r_emp.emp_job_title%>
<%=r_emp.department%>
<%=r_emp.hire_date%>
<%=r_emp.min_salary%>
<%=r_emp.max_salary%>
<%=r_emp.year_salary%>
<%=r_emp.manager%>
<%=r_emp.mgr_job_title%>
<%=r_emp.department%>
```

### Dear Mr./Ms. Fripp,

Human Resources Inc. is pleased to offer you the position of Stock Manager. Your skills and experience will be an ideal fit for our Shipping department.

As we discussed, your starting date will be 10 april 2005.

The salary scale for this job ranges from €5.500 to €8.500 per month. The salary is €98.400 per year and is paid on a monthly basis. Direct deposit is available.

Full family medical coverage will be provided through our company's employee benefit plan. Dental and optical insurance are also available.

Human Resource Inc. offers a flexible paid-time off plan which includes vacation, personal, and sick leave. Time off accrues at the rate of one day per month for your first year, then increases based on your tenure with the company.

We look forward to welcoming you to the Human Resource Inc. team.

Please let me know if you have any questions or I can provide any additional information.

Sincerely,

Steven King President, department Shipping

Human Resource Inc.

Output document generated from example2.docx

This is an example of an advanced Tables Report demonstrating some interesting constructions.

DEPARTMENTS										
Departments table that show Contains 27 rows; references	vs details of departme s with locations, empl	nts where emp oyees, and job	ployees work. b_history tables.							
Columns										
Name	Datatype	Nullable	Comments							
DEPARTMENT_ID	TMENT ID number (4) not null Primary key column of departments table.									
DEPARTMENT_NAME	PARTMENT_NAME       varchar2 (30)       not null       A not null column that shows name of a department. Administration, Marketing, Purchasing, Human Resources, Shipping, IT, Executive, Public         Palations       Salas       Salas       Einspec       and Accounting									
MANAGER_ID	ANAGER_ID number (6) n									
LOCATION_ID	number (4)		Location id where a department i	s located. Foreign key t	to location_id					
Name DEPARTMENTS	Colun depart	nn tment_id			Status enabled					
Name DEPARTMENTS Foreign keys	Colun depart	nn hment_id			Status enabled					
Name DEPARTMENTS Foreign keys Name	Colun depart Colun	nn tment_id	Related to table	Column	Status enabled Status					
Name DEPARTMENTS Foreign keys Name DEPT_MGR_FK	Colun depart Colun maaa	nn tment_id nn ger_id	Related to table EMPLOYEES	Column employee_id	Status enabled Status enabled					
Name DEPARTMENTS Foreign keys Name DEPT_MGR_FK DEPT_LOC_FK	Colun depart Colun manag locatio	nn tment_id nn ger_id on_id	Related to table EMPLOYEES LOCATIONS	Column employee_id location_id	Status enabled Status enabled enabled					
Name DEPARTMENTS Foreign keys Name DEPT_MGR_FK DEPT_LOC_FK Check constraints	Colum depart Colum manag locatio	nn hment_id nn ger_id on_id	Related to table EMPLOYEES LOCATIONS	Column employee_id location_id	Status enabled Status enabled enabled					
Name DEPT_MGR_FK DEPT_LOC_FK Check constraints Name	Colun depart Colun manag locatio	nn tment_id nn ger_id on_id	Related to table EMPLOYEES LOCATIONS	Column employee_id location_id	Status enabled Status enabled enabled					
Name DEPARTMENTS Foreign keys Name DEPT_MGR_FK DEPT_LOC_FK Check constraints Name DEPT_NAME_NN	Colun depart Colun manag locatio Condi "DEP.	nn tment_id nn ger_id on_id tion ARTMENT_1	Related to table EMPLOYEES LOCATIONS	Column employee_id location_id	Status enabled Status enabled enabled					
Name DEPARTMENTS Foreign keys Name DEPT_MGR_FK DEPT_LOC_FK Check constraints Name DEPT_NAME_NN	Colun depart Colun manag locatio Condi "DEP.	nn tment_id nn ger_id on_id tion ARTMENT_1	Related to table EMPLOYEES LOCATIONS	Column employee_id location_id	Status enabled Status enabled enabled					
Name DEPARTMENTS Foreign keys Name DEPT_MGR_FK DEPT_LOC_FK Check constraints Name DEPT_NAME_NN Indexes	Colun depart Colun manag locatio Condi "DEP.	nn tment_id	Related to table EMPLOYEES LOCATIONS	Column employee_id location_id	Status enabled status enabled enabled					
Name DEPT_MGR_FK DEPT_LOC_FK Check constraints Name DEPT_NAME_NN Indexes Name	Colun depart Colun manag locatio "DEP. Uniqu	nn tment_id	Related to table         EMPLOYEES         LOCATIONS	Column employee_id location_id	Status enabled Status enabled enabled					
Name DEPT_MGR_FK DEPT_LOC_FK Check constraints Name DEPT_NAME_NN Indexes Name DEPT_ID_PK DEPT_ID_PK DEPT_ID_PK	Colun depart Colun manag locatio "DEP. "DEP.	nn tment_id	Related to table         EMPLOYEES         LOCATIONS	Column employee_id location_id	Status enabled status enabled enabled enabled status valid					

Output document generated from example3.docx

### Example4.docx

This is an example to show if your Oracle database characterset works fine with the unicode characterset from Microsoft Word. It works 100% fine when you use the Oracle database characterset AL32UTF8.

### Declaration-tag:

```
<%!
___
cursor cl
is
select parameter
       value
from
      nls database parameters
where parameter in ( 'NLS_LANGUAGE', 'NLS_TERRITORY', 'NLS_CHARACTERSET')
order by parameter;
___
cursor c2
is
select unistr( '\0627\0644\0639\0631\0628\064A\0629' )
                                                              as arabic
       unistr( '\4E2D\6587')
                                                               as chinese
,
       unistr( 'English' )
                                                               as english
,
       unistr( 'Fran\00E7ais' )
                                                               as french
       unistr( 'Deutsch' )
                                                              as german
       unistr( '\0395\03BB\03BB\03B7\03BD\03B9\03BA\03AC' ) as greek
       unistr( '\05E2\05D1\05E8\05D9\05EA' )
                                                              as hebrew
       unistr( '\65E5\672C\8A9E' )
                                                              as japanese
       unistr( '\D55C\AD6D\C5B4' )
                                                              as korean
       unistr( 'Portugu\00EAs' )
unistr( '\0420\0443\0441\043A\0438\0439' )
                                                              as portuguese
                                                              as Russian
       unistr( 'Espa\00F1ol' )
                                                              as Spanish
       unistr( '\0E44\0E17\0E22' )
                                                               as Thai
from
       dual;
_ _
r2 c2%rowtype;
응>
```

arameter	Value
ILS_CHARACTERSET	AL32UTF8
LS_LANGUAGE	AMERICAN
ILS_TERRITORY	AMERICA
	<b>.</b>
anguage	Text
Arabic	فيبرع
Chinese	中文
English	English
French	Français
German	Deutsch
Greek	Ελληνικά
lebrew	תירבע
apanese	日本語
Korean	한국어
Dartuquada	Dortuguês

Output document generated from example4.docx

### Example5.docx

This is an example showing that you can include a picture from file system or url.

#### Declaration-tag:

```
응>
```

```
<%forr1 in c1 loop%>
Dear <%=r1.manager%>,
This demo shows that you can include a picture from file system or url.
Examine the SQLWord manual where we explain how to do it!
Sincerely,
Larry Ellison
×
                                            ×
<%end loop;%>
<$ !
___
cursor c1
is
select initcap(first_name || ' ' || last_name) as manager
       'http://www.sequel.nl/download/larry_ellison.jpg'
                                                            as image1
,
       'file:///C:/SQLWord11/Examples/Docx/SQLWordBox.png' as image2
from
      employees
where employee_id = 100;
*>
```

### How to include the pictures dynamicly:

Step 1: insert a field into the Word document



Step 2: choose IncludePicture and fill in a scriptlet in field Filename or URL:

<%=r1.image1%> and <%=r1.image2%>

	5	- 5	🖬 =												D	ocument	1 - Wor	d (Product /	Activatio	n Failed)	l.	
FILE	н	OME	INSERT	DESIG	N P	AGE LAY	OUT	REFERE	NCES	MAILINGS	REVIEV	V VIEW										
Cover Page *	Blank Page	Page Break	Table	Pictures	Online Pictures	Shapes	SmartArt	Chart	Screensh	ot Apps for Office *	Online Video	Hyperlink	Bookmark	Cross- reference	Comment	Header	Footer	# Page Number •	Text Box *	Quick Parts *	<b>A</b> WordArt	A Signature Line □ Drop Cap * Object *
	Pages		Tables			Illus	trations			Apps	Media		Links		Comments	He	ader & F	ooter				Text





Output document generated from example5.docx

### Example6.docx

This is an example showing that you can include a picture from a picture stored in the Oracle database as a BLOB. You do need to import file <u>C:\SQLWord11\SQL\image\_demo.dmp</u> to get table IMAGES\_DEMO with the two pictures !

Declaration-tag:

```
<응!
function get image (p id in number) return blob
is
  l_blob blob;
begin
  select data
  into l blob
  from image_demo
  where id = p_id;
  return(l blob);
end;
응>
   <%for r1 in c1 loop%>
   Dear <%=r1.manager%>,
   We send you the latest pictures from our Ocean race.
   ×
                                          ×
   These pictures are stored in table IMAGES DEMO.
   You do need to import file C:\SQLWord11\SQL\image_demo.dmp.
   Examine the SQLWord manual where we explain how to do it!
   Sincerely,
   Larry Ellison
   <%end loop;%>
   <%!
   cursor cl
   is
   select initcap(first_name || ' ' || last_name) as manager
   from employees
where employee_id = 100;
   function get_image (p_id in number) return blob
   is
l_blob blob;
   begin
     select data
    into l_blob
from image_demo
where id = p_id;
    return(l_blob);
   end;
   $>
```

### How to include the pictures dynamicly:

Step 1: insert a field into the Word document

w		5 - I	5	- -												D	ocument1	- Wore	d (Product A	Activation	n Failed)					
FIL		HOME		INSERT	DESIGN	N PA	GE LAYO	UT	REFERE	NCES N	AILINGS	REVIEV	V VIEW								-					
Cove Page	r Blar * Pag Pag	k Paj e Bre	] ge ak	Table • Tables	Pictures (	Online S Pictures	Shapes Si	martArt	Chart	Screenshot	Apps for Office * Apps	Online Video Media	Hyperlink B	lookmark Links	Cross- reference	Comment Comments	Header Hea	Footer •	# Page Number *	A Text Box *	Quick Parts • (A) A (C) D (C) D (C) D	VordArt vordArt utoText ocument	A Sign Drop Cap + Obje	e & Time e & Time ect ~ }	π Equatio	Ω n Symbol mbols
													Ĩ								E Bu	uil Inse	rt Field t a field.			

Step 2: choose IncludePicture and fill in a scriptlet in field Filename or URL:

<%=get\_image(1)%> and %=get\_image(2)%>





Output document generated from example6.docx

## HR\_\*.docx

The HR\_\*.docx templates belong to the Apex demo application.

## Steps to create a source document

## Step 1: Start SQLWord Developer



On starting SQLWord Developer the Oracle logon screen shows up.

Now connect to the Oracle schema where SQLWord tables and packages are installed by your DBA.

Oracle Logon		×
Usen	name	
Pass	word	
Data	base	-
Data		

After connecting to the Oracle database the SQLWord Developer toolbar appears and Microsoft Word is started from the SQLWord Developer application.



The SQLWord Developer toolbar can be moved to another position on your desktop. The best place is to position it in the upper right corner of your screen.

The next time when SQLWord Developer is started the SQLWord Developer toolbar is positioned on the last positon where you left it.

#### Step 2: Create a new source document

Press the 💴 button in SQLWord Developer toolbar. A new document is opened in Microsoft Word.

### Step 3: Type in your "static" content

For this of course you can use all Microsoft Word features. Be sure that the layout of all "static" content is done before you go to the next step.

Prepare your <% tag %> scriptlet in the work area. You can use the button  $\mathscr{P}$  to select the tag that you need and paste it from the submenu into the work area.

SQLWord <sup>11</sup>	- ×
🍪 🖬 🗀 🗔 🗠 🔍 Q  🛣 Z d	<% %> <%= %> <%! %> <%@ parameter= %>

Type in the work-area your SQL-statements:

SQLWord <sup>11</sup>	- ×
😳 🖬 🧀 🗔 ∽   🭳 🎼 🗙 🔗 🍼   ✔ 🍕 <%for r1 in c1(p_employee_id) 1	0 00 %>

Now paste the prepared scriptlet from the work area to the Word source document by pressing the paste **b**utton. Do this for all the scriptlets that you need.

### Step 5: Save the source document

Press the  $\square$  button to save the Word source document.

### Step 6: Compile the source document to a stored procedure

<u>Compile</u> the active source document to a <u>stored procedure</u> by pressing the **v** button.

After finishing this screen shows up if you did not make mistakes 🙂

Informa	ation	×
į)	Procedure <example1a> created succesfully!. Do you want to view the created PL/SQL-code?</example1a>	
	Yes No	

If the generated stored procedure contains errors <sup>(2)</sup> you probably want to see <u>the invalid</u> PL/SQL code. In that case choose "Yes" to examine the invalid read-only PL/SQL source code.

Always fix errors in your source document and recompile the source document until it is valid.

If you want to specify input parameters press the 💷 button. Change to the second tab "<u>Parameters</u>" to fill in a prompt and description or specify a lookup SQL-statement.

Note that lookup SQL-statements always must have two columns:

- 1. The <u>first</u> column must give an unique identifier that will be assigned to the input parameter after choosing.
- 2. The second column must give the description (varchar2) that is displayed in the lookup list.

Procedure: EXAMPLE1A - Valid ×								
Procedure Parameters								
Description								
example1a Commit								
Comme								
Parameters								
Parameter	Prompt	Description						
▶ P_EMPLOYEE_ID	Select a manager							
Lookup SQL statement select empl.employ , empl.first	Lookup SQL statement select emp1.emp1oyee_id as return_value , emp1.first name    ' '    emp1.last name as display value							
trom employees e	empl							
where exists (sei	um employees emp2							
whe	ere emp2.manageri(	d = empl.employee id)						
order by empl.last	c name	· · · · _ ·						
	_							

NB: Lookup's are only usefull when running SQLWord interactively "client-server".

#### Step 8: Run it

You can run the report that you just created by pressing the Run button from the SQLWord Developer toolbar.

Run SQLWord. The screen below shows up where you can select a stored procedure and specify values for file locations.

Run			1
Procedure	EXAMPLE1A - example1a	-	Show output
File location	15		1
Outfile	c:\temp1\EXAMPLE1A_out.docx	<b>a</b>	Read Only
Parfile		<b></b>	
Logfile	c:\temp1\EXAMPLE1A.log	<b>≧</b>	
			<u>R</u> un <u>C</u> lose

When pressing the Run button 🔲 a parameter screen appears where you can specify input parameters:

Parameters for proce	edure: EXAMPLE1A	×
Select a manager		
	Gerald Cambrault	~
	Alberto Errazuriz	
	Adam Fripp	
E	Nancy Greenberg	
	Michael Hartstein	
	Shelley Higgins	
	Alexander Hunold	
	Payam Kaufling	
	Neena Kochhar	
	Kevin Mouraos	
	Karen Partners	
	Den Raphaely	
	John Russell	
	Shanta Vollman	
	Matthew Weiss	100
	Eleni∠lotkey	×.

## Options

Press the menu button 💿 from the SQLWord toolbar and choose "Options" from the submenu.



The options window shows up. This window contains 3 tabs, which are described below.

### License tab

On the license tab you can see the license information or install your SQLWord license.

License Email	Client Server	
Licensed to		Install license
Max users		
Server name		

## Email tab

On the email tab you can specify email settings in case you want to send an output document by email from your Oracle database. SQLWord uses the Oracle UTL\_SMTP package.

citora op	ions		
icense Email	Client Server		
SMTP host	smtp.kpn.nl		Save
SMTP port	25		

<u>SMTP host:</u>	The name of your SMTP-server or IP-address.
<u>SMTP port:</u>	The port number, usually this is port 25.

For more information how to send email, examine section "Frequently asked questions".

## **Client Server tab**

On the Client Server tab you can specify the client settings specific for your PC.

SQLWord - Options		×
SQLWord - Options         License       Email         Client Server         Word editor         Image: Microsoft Word         Show maximized         Other executable		×
	Ok	Cancel

## Or

ense Email Client Server	_
Microsoft Word	
C:\Program Files\OpenOffice 4\program\soffice.exe	
Ok Cancel	

Microsoft Word:	If you choose this option SQLWord will use Microsoft Word as the default editor for all your Word documents.
Other executable:	If you want to use a different program than Microsoft Word (for example <u>Open Office</u> ) then you must specify which executable SQLWord should use to show the output document. SQLWord Developer doesn't use this editor for editing your docx-templates.
Show maximized:	Indicates if Microsoft Word should maximize on opening.

## **SQLWord Run**

## Introduction

SQLWord Run is a 32-bits Windows application for running SQLWord reports interactively or in batch mode from the command line.



## Menu options:

File: Shows a submenu where you can:

- Connect to your Oracle database.
- Display the options-window.

<u>Run</u>: The screen below shows up where you can select a stored procedure and specify values for file locations.

Run			×
Procedure	EXAMPLE1A - example1a	▼ Show c	output
File location	18		
Outfile	c:\temp1\EXAMPLE1A_out.docx	Read O	nly
Parfile		<i>⊯</i>	
Logfile	c:\temp1\EXAMPLE1A.log	<b></b>	
		[ <u>Bun</u> ]	<u>C</u> lose

When pressing the Run button a parameter screen appears where you can specify input parameters:

Parameters for proc	edure: EXAMPLE1A	×
Select a manager	Gerald Cambrault Lex De Haan Alberto Errazuriz Adam Fripp Nancy Greenberg Michael Hartstein Shelley Higgins Alexander Hunold Payam Kaufling Steven King	
	Neena Kochhar Kevin Mourgos Den Raphaely John Russell Shanta Vollman Matthew Weiss Eleni Zlotkey	

Help: Shows a submenu where you can:

- Display this Users Guide & Reference.
- Show the about box.



## **Command line syntax**

You can run the SQLWordRun executable from the command line with the following syntax: SQLWordRun.exe keyword=<value> keyword=<value> keyword=<value> etc.

Keyword	Description
userid=username/password@host	Oracle connect string.
procedure=stored-procedure-name	Name of the stored procedure to execute.
outfile=filename	Name and location of the output-file. In the run window the output file name can be specified and the directory can be choosen using the button on the right of the field.
[parfile=filename]	Name and location of the parameter file. In the run window the parameter file can be choosen using the button on the right of the field.
[logfile=filename]	Name and location of the log file. This file contains the logging information of the execution. In the run window the log file can be choosen, using then select button on the right of the field.
[editor=Y/N]	Open the output document with the default Word editor.
[readonly=Y/N]	Protect the output document by setting a <u>secret</u> password on the output document. This option is only available for Microsoft Word.
[print=Y/N]	Print the output document. This option is only available for Microsoft Word.
[copies=number]	The number of copies to print. This option is only available for Microsoft Word.
[printer=printername]	The name of the printer. This option is only available for Microsoft Word.
[showerror=Y/N]	To suppress all interactive messages (usefull for batch jobs).
[wordmacro=macroname]	Run a Word Macro on opening of the output document. This option is only available for Microsoft Word.
[role=rolename/password]	You can enable a database role when running SQLWord

The keywords between the straight [brackets] are optional keywords.

### Example1:

SQLWordRun.exe" userid=sqlword\_demo/sqlword\_demo@my\_db procedure=example1a outfile="C:\Temp\example1a\_out.docx" parfile="C:\SQLWord11\Examples\Docx\example1.par"

### Example2:

```
SQLWordRun.exe" userid=sqlword_demo/sqlword_demo@my_db
procedure=example1a outfile="C:\Temp\example1a_out.docx"
parfile="C:\SQLWord11\Examples\Docx\example1.par"
readonly=Y
```

### Example3:

```
SQLWordRun.exe" userid=sqlword_demo/sqlword_demo@my_db
procedure=example1a_outfile="C:\Temp\example1a_out.docx"
parfile="C:\SQLWord11\Examples\Docx\example1.par"
editor=N print=Y copies=1 printer="HP OFFICEJET G SERIES"
```

You can find a sample batch script at:

C:\SQLWord11\Examples\Docx\run\_cs\_example1a.bat

#### **Parameter file**

In a parameter file you can specify the values for the input parameters.

SQLWordRun reads the parameter file before execution the stored procedure and assigns the values to the input parameters to the called stored procedure.

The parameter file has the following syntax:

### <PARAMETER>=<VALUE>

Example:

DEPTNO=10 HIRE\_DATE=02-07-2009 ENAME='JONES'

You can find a sample parameter file at:

C:\SQLWord11\Examples\Templates\example1.par

## **SQLExcel**

end;

## How to generate Microsoft Excel XSLX

SQLWord11 supports generating Microsoft Excel files by package SQLEXCEL.

SQLEXCEL has several functions to create and write data to XLSX documents. For more information examine the package specifications.

You can find an example how to create a stored procedure for generating a Microsoft Excel file at:

C:\SQLWord11\Examples\SQL\excel\_example1.sql

You can find an example how to generate Microsoft Excel output at:

C:\SQLWord11\Examples\SQL\ run\_excel\_example1.sql

First you must use the Oracle "create directory" command (ask your DBA to do this).

```
SQL> create or replace directory SQLWORD_OUTPUT_DIR as 'C:\Temp';
SQL> grant read, write on directory SQLWORD_OUTPUT_DIR to public;
begin
---
excel_example1;
--
sqlexcel.save( p_directory => 'SQLWORD_OUTPUT_DIR'
, p_filename => 'excel_example1.xlsx');
```

Open the Excel-example1.xlsx file from the file location on your Oracle database server. This spreadsheet has <u>two</u> tab sheets:

X	ILE HOME INSE	RT PAGE LAYOUT F	ORMULAS DATA REVIE	W VIEW	I
A	1 • : ×	✓ f <sub>*</sub> Region			
	А	В	С	D	
1	Region	Country	Departments		
2	Americas	Argentina			
3		Brazil			
4		Canada	Marketing		
5		Mexico			
6		United States of America	Accounting		
7			Administration		
8			Benefits		
9			Construction		
10			Contracting		
11			Control And Credit		
12			Corporate Tax		
13			Executive		
14			Finance		
15			Government Sales		
16			IT		
17			IT Helpdesk		
18			IT Support		
19			Manufacturing		
20			NOC		
21			Operations		
22			Payroll		
23			Purchasing		
24			Recruiting		
25			Retail Sales		
26			Shareholder Services		
27			Shipping		
28			Treasury		
29	Asia	Australia			
30		China			
31		India			
32		Japan			
33		Malaysia			
34		Singapore			
35	Europe	Belgium			
36		Denmark			
37		France	1		
38		Germany	Public Relations		
39		Italy			
40		Netherlands			
41		Switzerland			
42		United Kingdom	Human Resources		
43			Sales		
44	Middle East and Africa	Egypt			
45		Israel			
46		Kuwait			
47		Nigeria			
48		Zambia			
49		Zimbabwe			
50					
51					
- 1					_
	Regions	<b>Countries Departments</b>	Departments Employees	(+)	

## **Departments Employees**

2	D
jer sistant	
sistant	
	ŝ
e Preside	ent
e Preside	ent
	1
	1
Represe	ntative
entative	
er	
Representa	ative
	<u></u>
ger	
tive	
tive	
tive	
tivo	
tive	
uve	
tivo	
tivo	
tive tive	ees

## **Apex integration**



An Apex demo application (for Apex 5.0.3 or higher) is available at: C:\SQLWord11\Examples\Apex5.0.

The Apex demo application is based on Oracle HR-tables and demonstrates how to integrate SQLWord within Apex.

## Installation

## Step 1: Import the SQLWord11 HR DEMO application

Open the Apex Application Builder and import file: C:\SQLWord11\Examples\Apex5.0\f100.sql

	Application Builder 🛇 SQL Workshop 🛇 Team Development 🛇 Packaged Apps 🛇	Q & OV Q ADMIN V
(n) Import		
	Interpret         Methods         Interpret         Interpret	
Q admin 📾 ws_sqiword ⊕ en	Copyright @ 1999, 2015, Oracle. All rights reserved.	Application Express 5.0.3.00.03

Q admin 📓 ws\_sqlword ⊕ en

## After finishing the import you should see these pages:

ORACLE AF	plication Express Application Builder 😔 SQL We	rkshop 🕑 🛛 Team Development 🤇	Packaged Apps					Q,	9 <sub>≠</sub> ~ ⑦~ ℚ admir
Application 10	0							€ %	A 🗹 1 🕑
Application 100 -	SQLWord demo - Human Resources						Edit Ap	pication Properties	Tasks
									Delete this Application
									Copy this Application
									Recently Edited Pages
		SQL			•				0. Global Page - Desktop
	Run Application	Supporting Objects	Shared	Components	Utilities		Export / Import		1030. Departments
								-	1020. Employees
Q~	Go 🔠 🖬 Actions	~						Create Page >	1031. Department detail
Page <u>⊾</u> Î	Name	Updated	Updated By	Page Type	Group	User Interface	Lock	Run	1001. Manager detail
	0 Global Page - Desktop	5 hours ago	admin	Global Page	Unassigned	Desktop	6		6. HR Tables report
	1 Home	6 months ago	adk	Home	Unassigned	Desktop	6	۲	7. Excel report
	2 Architecture	6 months ago	adk	Static HTML	Unassigned	Desktop	6	۲	
	4 Reports	6 months ago	adk	Navigation Page	Unassigned	Desktop	6	۲	
	5 Managers report	6 months ago	adk	Static HTML	Unassigned	Desktop	2	۲	
	6 HR Tables report	6 months ago	adk	Static HTML	Unassigned	Desktop	2	۲	
	7 Excel report	6 months ago	adk	Static HTML	Unassigned	Desktop	ъ	۲	
10	01 Login Page	7 months ago	adk	Login	Unassigned	Desktop	ъ	۲	
50	0 Page	6 months ago	adk	Static HTML	Unassigned	Desktop	6	۲	
50	01 SQLWordChooseFormat	6 months ago	adk	Static HTML	Unassigned	Desktop	6		
100	0 Managers	6 months ago	anonymous	Interactive Report	Unassigned	Desktop	2	۲	
100	11 Manager detail	6 months ago	adk	Report	Unassigned	Desktop	6		
102	20 Employees	5 months ago	admin	Interactive Report	Unassigned	Desktop	ъ	۲	
103	Employee detail	6 months ago	admin	DML Form	Unassigned	Desktop	6		

## Step2: Compile all HR\*.docx files

Start SQLWord Developer and connect to user SQLWORD\_DEMO

Click on button in the SQLWord toolbar and select all HR\_\*.docx files at C:\SQLWord11\Examples\Docx

Zoeken in:	Docx	-		
4	Naam 🔺	Gewijzigd op	Туре	Grootte
	Example1a.docx	11-3-2015 10:00	Microsoft Office	18 kB
onelle toegang	Example1b.docx	11-3-2015 10:00	Microsoft Office	17 kB
	Example1c.docx	11-3-2015 10:00	Microsoft Office	18 kB
Bureaublad	Example1d.docx	29-5-2015 16:48	Microsoft Office	16 kB
	Example2.docx	11-3-2015 10:01	Microsoft Office	19 kB
-	Example3.docx	18-1-2017 16:18	Microsoft Office	39 kB
Bibliotheken	Example4.docx	11-3-2015 10:02	Microsoft Office	17 kB
	Example5.docx	13-3-2015 17:37	Microsoft Office	19 kB
	🔁 Example6.docx	13-3-2015 16:17	Microsoft Office	22 kB
Deze pc	HR_Dept.docx	18-1-2017 16:19	Microsoft Office	118 kB
	HR_Dept_Selection.docx	18-1-2017 16:19	Microsoft Office	113 kB
Network	HR_Employee_Contract.docx	18-1-2017 16:19	Microsoft Office	112 kB
INCLINEIK	HR_Employee_Job_Offer.docx	18-1-2017 16:19	Microsoft Office	115 kB
	HR_Employee_Selection.docx	18-1-2017 16:20	Microsoft Office	113 kB
	HR_Manager.docx	18-1-2017 16:20	Microsoft Office	116 kB
	HR_Tables_Report.docx	18-1-2017 16:20	Microsoft Office	136 kB
	Bestandsnaam: "HR_Tables_Report.d	locx" "HR_Dept.docx"	"HR_Dept_Selection.docx" "H	F • Opener
	Bestandstypen: Word-documents (* do	ac *docx *ttf)		<ul> <li>Annuler</li> </ul>

The selected files are displayed in this window.

## Now press on button Create All

File	Stored procedure	Status	
C:\SQLWord11\Examples\Docx\HR_Dept_Selection.docx			Lreate al
C:\SQLWord11\Examples\Docx\HR_Employee_Contract.docx			Class
C:\SQLWord11\Examples\Docx\HR_Employee_Job_Offer.docx			Ciose
C:\SQLWord11\Examples\Docx\HR_Employee_Selection.docx			
C:\SQLWord11\Examples\Docx\HR_Job.docx			
C:\SQLWord11\Examples\Docx\HR_Job_Selection.docx			
C:\SQLWord11\Examples\Docx\HR_Location_Selection.docx			
C:\SQLWord11\Examples\Docx\HR_Manager.docx			
C:\SQLWord11\Examples\Docx\HR_Tables_Report.docx			



			×
File	Stored procedure	Status	
C:\SQLWord11\Examples\Docx\HR_Dept.docx	sqlword11.HR_DEPT	Valid	Ureate all
C:\SQLWord11\Examples\Docx\HR_Dept_Selection.docx	sqlword11.HR_DEPT_SELECTION	Valid	Church
C:\SQLWord11\Examples\Docx\HR_Employee_Contract.docx	sqlword11.HR_EMPLOYEE_CONTRACT	Valid	Liose
C:\SQLWord11\Examples\Docx\HR_Employee_Job_Offer.docx	sqlword11.HR_EMPLOYEE_JOB_OFFER	Valid	
C:\SQLWord11\Examples\Docx\HR_Employee_Selection.docx	sqlword11.HR_EMPLOYEE_SELECTION	Valid	
C:\SQLWord11\Examples\Docx\HR_Job.docx	sqlword11.HR_JOB	Valid	
C:\SQLWord11\Examples\Docx\HR_Job_Selection.docx	sqlword11.HR_JOB_SELECTION	Valid	
C:\SQLWord11\Examples\Docx\HR_Location_Selection.docx	sqlword11.HR_LOCATION_SELECTION	Valid	
C:\SQLWord11\Examples\Docx\HR_Manager.docx	sqlword11.HR_MANAGER	Valid	
C:\SQLWord11\Examples\Docx\HR_Tables_Report.docx	sqlword11.HR_TABLES_REPORT	Valid	

## **Implementation explained**

In this section the implemention of a SQLWord Job offer letter in Apex is explained.

Document <u>HR Employee Job Offer.docx</u> is the source document for this letter.

- Start the SQLWord HR Demo application and choose <u>Employees</u> from the menu.
- Click on *I* from employee <u>David Austin</u> to go to the Employee detail (modal) page 1021:

≡ SQLWord - Sa	mple HR	Application								
# SQLWord										
🛎 Employees	107	SQLWord <sup>11</sup>								difficu.
嶜 Managers	18									
. Departments	27		a Actions Y							Selection report
Reports	~		ACTIONS -							
Architecture		Employee	Job	Hire date	Department	Managed by	_	Salary	Phone number	Email
		Steven King	President Employee	detail			×	€24.000	515.123.4567	s.king@gmail.com
		Neena Kochhar	Administrat		Jot	offer 📲 🖬 Labor contract 📲	2	€17.000	515.123.4568	n.kochhar@gmail.com
		Lex De Haan	Administrat First Name	David			0	€17.000	515.123.4569	I.de haan@gmail.com
		Alexander Hunold	Programme Last Name	Austin			0	€9.000	590.423.4567	a.hunold@gmail.com
		Bruce Ernst	Programme	Programmer		0	0	€6.000	590.423.4568	b.ernst@gmail.com
		David Austin	Programme	25.05.2005		-		€4.800	590.423.4569	d.austin@gmail.com
		🖉 Valli Pataballa	Programme	25-06-2005			0	€4.800	590.423.4560	v.pataballa@gmail.com
		Diana Lorentz	Programme Salary	4800			0	€4.200	590.423.5567	d.lorentz@gmail.com
		Nancy Greenberg	Finance Ma Managed by	Alexander Hunold		0	0	€12.008	515.124.4569	n.greenberg@gmail.com
		Z Daniel Faviet	Accountant Department	IT		0	0	€9.000	515.124.4169	d.faviet@gmail.com
		🖉 John Chen	Accountant Email	d.austin@gmail.co	m		0	€8.200	515.124.4269	j.chen@gmail.com
		🖉 Ismael Sciarra	Accountant	590.423.4569			0	€7.700	515.124.4369	i.sciarra@gmail.com
		🖉 Jose Manuel Urman	Accountant					€7.800	515.124.4469	j.urman@gmail.com
		🖉 Luis Popp	Accountant					€6.900	515.124.4567	l.popp@gmail.com
		🖉 Den Raphaely	Purchasing				_	€11.000	515.127.4561	d.raphaely@gmail.com
		Alexander Khoo	Purchasing			Apply Change	•	€3.100	515.127.4562	a.khoo@gmail.com
		🖉 Shelli Baida	Purchasing Clerk	24-12-2005	Purchasing	Den Raphaely		€2.900	515.127.4563	s.baida@gmail.com
		🖉 Sigal Tobias	Purchasing Clerk	24-07-2005	Purchasing	Den Raphaely		€2.800	515.127.4564	s.tobias@gmail.com
		🖉 Guy Himuro	Purchasing Clerk	15-11-2006	Purchasing	Den Raphaely		€2.600	515.127.4565	g.himuro@gmail.com
		🖉 Karen Colmenares	Purchasing Clerk	10-08-2007	Purchasing	Den Raphaely		€2.500	515.127.4566	k.colmenares@gmail.com
		🖉 Matthew Weiss	Stock Manager	18-07-2004	Shipping	Steven King		€8.000	650.123.1234	m.weiss@gmail.com
		🖉 Adam Fripp	Stock Manager	10-04-2005	Shipping	Steven King		€8.200	650.123.2234	a.fripp@gmail.com
		Payam Kaufling	Home 🗹 Application 100	🗹 Edit Page 1021	) Session 🗐 View Debu	g 💢 Debug 🆽 Show Grid 👀 Quict	k Edit 🔓 Then	ne Roller &	550.123.3234	p.kaufling@gmail.com

• Click on button Job Offer and choose option Microsoft Word to generate the Microsoft Word document.

Employee de	tail		x
		Job offer	Labor contract
First Name	David		()
Last Name *	Austin	Choose output format	× ?
Job *	Programmer		• ?
Hire date *	25-06-2005	Microsoft Word	Vord <sup>11</sup>
Salary	4800	Adobe PDP	(?)
Managed by	Alexander Hunold	Worset	And Sector Loss
Department	IT		• ?
Email	d.austin@gmail.cor		0
Phone	590.423.4569		
			Apply Changes ✔

Human Resources Inc. Department IT 2014 Jabberwocky Rd 26192 Southlake United States of America

#### Dear Mr./Ms. Austin,

Human Resources Inc. is pleased to offer you the position of Programmer. Your skills and experience will be an ideal fit for our IT department.

As we discussed, your starting date will be 25 june 2005.

The salary scale for this job ranges from \$4,000 to \$10,000 per month.

The salary is \$57,600 per year and is paid on a monthly basis. Direct deposit is available.

Full family medical coverage will be provided through our company's employee benefit plan. Dental and optical insurance are also available.

Human Resource Inc. offers a flexible paid-time off plan which includes vacation, personal, and sick leave. Time off accrues at the rate of one day per month for your first year, then increases based on your tenure with the company.

We look forward to welcoming you to the Human Resource Inc. team.

Please let me know if you have any questions or I can provide any additional information.

Sincerely,

#### Alexander Hunold

Programmer, department IT

Human Resource Inc.

### How does it work?

## > Open page 1021. This page is <u>a modal page</u> called from page 1020

ORACLEA	pplication Express	Application Builder	SQL Workshop	C Team Development	Packaged Apps 😔										C	2 8	~~ (	D~ (?		· ~
Application 10	0 Page Designer									🗅 🗠 1021 G	6	ଅ ୯ +	~ 2°~	1	P <sub>R</sub> ∨	Ð	æ	Save	۲	
B	4	¢p	æ	Grid Layout		Messages		Page Search		Help			Button							
Rendering		12		⊇ ⊕ ⊭"									÷ Ξ						4	~
Regions			*	D. Employee datail								▼ Identificatio	on							-
Dialog H	eader			Employee detail								Rutton Name	Btn JobO	for						51
→ Content E	Sody			DIALOG HEADER									Datoobo							÷
<ul> <li>Gene</li> </ul>	raitems (Global Page)			CONTENT BODY								Label	Job offer-	anbsp&nbs	p &nbs	sp	anbsp&n	bsp		1
T () Emple	ovee			> Employee								▼ Lavout								
:= A1	tributes																			
→ Ite	ams			TOP OF REGION								Sequence	20							. 1
9	Ø P1021_EMPLOYEE,	_ID					BINJODU	binshow	riemplateJobOff	er		Region	Employe	9					$\diamond$ >	ī
9	Ø P1021_URL						BtnLabor	Contract	nMSWordTempla	teLaborContract		Button	Top of Re	alon					0	ā.
I	P1021_FIRST_NAM	E -										Position								1
LL.	PIUZI_LASI_NAME	-		PREVIOUS								Horizontal	Right						0	í.
10	2 P1021_JOB_ID			ITEMS								Alignment								18
1	P1021 SALARY			P1021_FIRST_NA	ME							· Appoarance								
	P1021_COMMISSIO	IN_PCT		P1021 LAST NA	IE							• Appearance	20							- 1
IS	P1021_MANAGER_	ID										Button	Text						$\diamond$ >	
IS	P1021_DEPARTMEN	NT_ID		I P1021_JOB_ID								remplate								
Ι	P1021_EMAIL			P1021 HIRE DAT	F							Hot	Yes	No						
I	P1021_PHONE_NU	MBER										Template			Line Temple	ate Defe	dia Casali			1
⊤ 🗁 R	egion Buttons			P1021_SALARY								Options			Ose lempla	.te Delau	ins, oman			4
▼ <sup>1</sup> / <sub>2</sub>					ON ROT							CPP Classes	maurard	loop class						7
*	Dynamic Actions			E Pitzi_commiss	on_PO1							COO Classes	msworu_	icon-ciass						4
	✓ 4 DaJobOffer			DI P1021_MANAGER	ID							Icon CSS							^	
	<ul> <li>True</li> <li>Execu</li> </ul>	ite PL/SQL Code		P1021_DEPARTM	ENT_ID							▼ Behavior								
	Execu Chi False	ite JavaScript Code		T P1021_EMAIL								Action	Defined	w Dunamia	Action				. :=	
	btnShowTemplateJo	bOffer											Denned	ly Dynamic	Accon				~ =	
▶ 🛱	BtnLaborContract			Designe Home Du	1000		÷					Execute	Yes	No						
_	btnMSWordTemplate	LaborContract		Regions items bu	IONS						8.	validations								
🔻 🗁 Dialog Fo	ooter								-			Database	- Select -						$\diamond$	
	ns						J×	Ø	⊞r.			Action								
I AI	ttributes			Dreaucrumo Calenda	Chart	Cassac Report	(based on	metp Text	miler active Repo	at Legacy Calendar		<ul> <li>Advanced</li> </ul>								
* 🗁 R	egion Buttons						Function)													
	\$ SAVE			:= 0	Î≣							Static ID								
Post-Render	ing		v	List Map Cha	t PL/SQL Dynamic	Region Display	Static Content	Tabular Form	Tree	URL	-	Custom							^	•

Button BtnJobOffer has a Dynamic Action DaJobOffer with 2 true actions:

Execute PL/SQL code

The PL/SQL code fills page-item :P1021\_URL with the url for calling page-501 in the next step.

Page-0 items :P0\_PK and P0\_PROCEDURE are also set here.

Execute Javascript code

```
var l_url = apex.item('P1021_URL').getValue();
eval(l_url);
```

This piece of javascript calls page 501 where the user is asked to choose an output format.



Button BtnRun on page 501 has a Dynamic Action daRunSQLWord with true action:

Execute Javascript code

```
v_filetype = $('input[name=radioName]:checked', '#myForm').val();
```

```
parent.runSQLWord(v_filetype);
```

Here we call a javascript function runSQLWord from page 1020.

```
function runSQLWord(p_filetype) {
    apex.item('P0_FILETYPE').setValue(p_filetype);
    window.pageloadWait = apex.util.showSpinner();
    setTimeout(function() { apex.event.trigger(document, 'CustomEventRunSQLWord'); }, 10);
}
```

This function calls CustomEventRunSQLWord from page 1020 with true actions:

Execute PL/SQL code

```
declare
  l blob
                  blob;
                 varchar2(4000);
 l_filename
 v_where_clause varchar2(4000);
               varchar2(4000);
 v_report_id
begin
  if apex_collection.collection_exists('SQLWORD_BLOB') then
   apex collection.delete collection('SQLWORD BLOB');
  end if;
  if :P0 PROCEDURE = 'HR_EMPLOYEE_SELECTION' then
    1 filename := 'EmployeeSelection-' || to char(sysdate,'dd-mm-yyyy-hh24.mi.ss') || '.' ||
                  lower(:P0 FILETYPE);
    v where clause := 'where 1=1 ' || apex ir query.ir query where( app id in
                                                                                          => :APP ID
                                                                     , page_id_in
                                                                     , page_id_in => :APP_PAGE_ID
, session_id_in => :APP_SESSION
                                                                     , base_report_id_in => :P1020_REPORT ID
                                                                     );
    hr employee selection( p where clause => v where clause);
    ---
  else
    select substr(lower(replace(first_name || '_' || last_name || '.' ||
lower(:P0_FILETYPE), ' ', '_')), 1, 4000)
          l_filename
employees
    into
    from
    where employee id = :PO PK;
    if upper(:PO_PROCEDURE) = 'HR_EMPLOYEE_JOB_OFFER' then
      l filename := 'job offer ' || l filename;
      hr employee job offer(p employee id => :P0 PK);
    elsif upper(:PO PROCEDURE) = 'HR EMPLOYEE CONTRACT' then
      1 filename := 'employee contract ' || 1 filename;
      hr_employee_contract(p_employee_id => :P0_PK);
    end if;
  end if;
  if upper(:P0_FILETYPE) = 'PDF' then
    sqlword.save output pdf( p utl file location => 'SQLWORD OUTPUT DIR'
                            , p_pdf_filename => l_filename );
    l blob := sqlword.get output pdf( p utl file location =>'SQLWORD OUTPUT DIR'
                                    , p pdf filename => l filename);
  else
   l blob := sqlword.get output docx;
  end if;
  apex_collection.create_or_truncate_collection(p_collection_name => 'SQLWORD_BLOB');
  apex_collection.add_member( p_collection_name => 'SQLWORD_BLOB'
                             , p_c001 => l_filename
                             , p blob001 => l blob );
end;
```

As you can see in this piece of PL/SQL code we call generated SQLWord stored procedures and then get the output into a blob and put the blob into a collection which is picked up later from page 500.

### Execute Javascript code

```
window.location = 'f?p=&APP_ID.:500:&APP_SESSION.:::::::;;
```

window.pageloadWait.remove();

Here we call page 500 a generic page for downloading output documents.

### ➢ Open page 500

This is a empty page for downloading documents from a collection.



### Examine the source code from Pre-Rendering/process/DOWNLOAD

```
declare
  ___
 cursor cl
 is
 select c001
       blob001
 from apex collections
 where collection name = 'SQLWORD BLOB';
  --
 l_blob blob;
l_filename varchar2(4000);
begin
  --
 open cl;
 fetch c1 into l_filename, l_blob;
 if c1%found then
    htp.p('content-length: ' || dbms lob.getlength(l blob));
   htp.p('content-disposition: attachment; filename="' || 1_filename || '"');
   owa_util.http_header_close;
    wpg docload.download file(1 blob);
 end if;
  _ _
 close cl;
end;
```

### Explanation

- The output document and filename is retrieved from the Apex collection SQLWORD\_BLOB.
- The HTML header is prepared.
- By calling Apex procedure WPG DOCLOAD.DOWNLOAD FILE the download will start.
- Page 500 is closed and the browser returns back to page 1020.

## **PDF output (optional)**

SQLWord can produce PDF output after the .docx generation by transforming the generated .docx to a .pdf file. This is done by calling a Java program on the Oracle server that uses the Aspose.Words for Java library. To use this functionality you need to buy a license at <u>http://www.aspose.com</u>. Default the evaluation version of Aspose.Words is installed which places some extra evaluation text in the pdf-file.

How does it work?

- First SQLWord generates a .docx file on the output directory on the Oracle server defined by a CREATE DIRECTORY command.
- When the .docx generation is finished then SQLWord runs an OS command using DBMS\_SCHEDULER. It calls a <u>shell script</u> on the Oracle server that <u>executes a Java program</u> to transform the .docx file into a .pdf file.

### **Installation on Unix Oracle Server**

We assume that you already have compiled all SQLWord demo examples by SQLWord Developer from C:\SQLWord11\Examples\Docx

### Steps to follow:

Create on the Oracle server a Unix user: <u>sqlword</u>

All the scripts that we made assume that the <u>home directory</u> of the Unix sqlword user is /<u>home/sqlword</u>. If you have a different home directory (for example /<u>opt/sqlword</u>) then you need to edit these scripts:

- sqlword\_output\_dir.sql
- sqlword2pdf.sh
- sqlword2pdf.java
- FTP the file <u>sqlword2pdf.zip</u> to the home directory of the sqlword user at /<u>home/sqlword</u>
- Unzip this file:

 $\geq$ 

\$ cd /home/sqlword

\$ unzip sqlword2pdf.zip.

Now a directory sqlword2pdf with 4 subdirectories are created (bin, log, output and sql)

- Change permissions:
  - \$ cd /home/sqlword
- \$ chmod 777 sqlword2pdf
- \$ cd /home/sqlword/sqlword2pdf
- \$ chmod 777 bin output log
- \$ cd /home/sqlword/sqlword2pdf/bin

\$ chmod +x \*.sh

Make sure that <u>pam</u> is properly installed. In particular make sure that the file <u>libpam.so</u> exists and is linked to a valid existing file.

For exampe when you have Redhat OS create a link as root: # ln -s /lib64/libpam.so.0 /lib64/libpam.so

Run script grants.sql as a DBA-user:

SQL> @ grants.sql

Edit script create\_credential.and change the password to the password of the sqlword Unix user. Then run this script as the Oracle schema owner where you installed the SQLWord tables and packages

SQL> @ create credential.sql

> Run script sqlword output dir.sql as a DBA-user:

SQL> @ sqlword\_output\_dir.sql

Let's see if we now can create a .docx on the Oracle server by running script test\_generate\_docx.sql from the Oracle schema owner where you installed the SQLWord tables and packages

SQL>@test\_generate\_docx.sql

Now you should see the file example1a out.docx at /home/sqlword/sqlword2pdf/output

Download JDK 1.8 (or higher) from Oracle <u>http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html</u> and install it properly. Check if you have the right <u>javac</u> version

\$ javac -version should give back: javac 1.8 or higher

NB: Make sure that the bin-directory of the JDK is in your PATH-environment variable !

Run file compile.sh and check if the Java compilation created file sqlword2pdf.class.

\$ cd /home/sqlword/sqlword2pdf/bin

\$./compile.sh

Now you have a Java executable that uses the evaluation version of Aspose.Words.

If you bought an Aspose license then place your <u>Aspose.Words.lic</u> in directory /home/sqlword/sqlword2pdf/bin

Edit file <u>sqlword2pdf.java</u> and remove the // comments.

License license = new License();

license.setLicense("Aspose.Words.lic");

Again run file /home/sqlword/sqlword2pdf/bin/compile.sh and check if the Java compilation created

a new class file at /home/sqlword/sqlword2pdf/bin/sqlword2pdf.class

Now test if the Java progam sqlword2pdf.class can create a .pdf file:

\$ cd /home/sqlword/sqlword2pdf/bin

\$ ./sqlword2pdf.sh example1a\_out.docx example1a\_out.pdf

Now you should see <u>example1a\_out.pdf</u> in directory <u>/home/sqlword/sqlword2pdf/output</u>

Now finally test if SQLWord now can generate pdf by running script test\_generate\_pdf.sql from the Oracle schema owner where you installed the SQLWord tables and packages.

SQL> @test\_generate\_pdf.sql

Now you should see the file <u>example1b\_out.pdf</u> at /home/sqlword/sqlword2pdf/output

## **Installation on Microsoft Windows Server**

We assume that you already have compiled all SQLWord demo-examples with SQLWord Developer from C:\SQLWord11\Examples\Docx

### Steps to follow:

> Logon on your Windows Server as <u>administrator</u> and go to <u>Services</u>.

Q,		Services					x
<u>B</u> estand <u>A</u> ctie Be	eel <u>d H</u> elp						
(* •) 🖬 🗐 🦉	Q 🔒 🛛 🖬 🕨 🔳 🕪						
🤹 Services (lokaal)	Services (lokaal)						
	OracleJobSchedulerORCL1	Naam	Beschrijving	Status	Opstarttype	Aanmelden als	^
		Oracle ORCL1 VSS Writer Service			Handmatig	Local System	
	De service <u>starten</u>	CracleJobSchedulerORCL1			Handmatig	Local System	
		OracleMTSRecoveryService		Wordt uitgevoerd	Automatisch	Local System	
		OracleOraDb11g_home1TNSListener		Wordt uitgevoerd	Automatisch	Local System	
		OracleServiceORCL1		Wordt uitgevoerd	Automatisch	Local System	
		Reformance Counter DLL Host	Enables rem		Handmatig	Local Service	~
	Uitgebreid Standaard	1989			102	406	

Start the Windows Service <u>OracleJobScheduler%ORACLE\_SID%</u> and change the settings of this service to <u>Automatic</u>.

<b>Q</b>		Services					x
<u>B</u> estand <u>A</u> ctie Be	eel <u>d H</u> elp						
	🗟 🔒 🛛 📷 🕨 🖬 🕪						
🧟 Services (lokaal)	Services (lokaal)	-					
	OracleJobSchedulerORCL1	Naam	Beschrijving	Status	Opstarttype	Aanmelden als	^
		Oracle ORCL1 VSS Writer Service			Handmatig	Local System	
	De service stoppen	CracleJobSchedulerORCL1		Wordt uitgevoerd	Automatisch	Local System	
	De service opnieuw starten	OracleMTSRecoveryService		Wordt uitgevoerd	Automatisch	Local System	
		OracleOraDb11g_home1TNSListener		Wordt uitgevoerd	Automatisch	Local System	
		OracleServiceORCL1		Wordt uitgevoerd	Automatisch	Local System	
		Reformance Counter DLL Host	Enables rem		Handmatig	Local Service	~
	Uitgebreid Standaard	1989					

> Unzip the <u>SQLWord2.zip</u> to <u>C:\SQLWord2pdf</u> on your Windows server.



▶ Run script grants.sql as a DBA-user:

SQL> @ grants.sql

Run script sqlword\_output\_dir.sql as a DBA-user:

SQL> @ sqlword\_output\_dir.sql

Let's see if we now can create a .docx on the Oracle server by running script test\_generate\_docx.sql from Oracle schema owner where you installed the SQLWord tables and packages

SQL> @test\_generate\_docx.sql

Now you should see the file <u>example1a\_out.docx</u> at <u>C:\SQLWord2pdf\output</u>

Download JDK 1.8 (or higher) from Oracle <u>http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html</u> and install it properly.

Check if you have the right javac version:

NB: Make sure that the bin-directory of the JDK is in your PATH-environment variable !!!

C:\SQLWord2pdf\bin> javac -version should give back: javac 1.8 or higher

Run file <u>compile.bat</u> and check if the Java compilation created at C:\SQLWord2pdf\bin\sqlword2pdf.class C:\SQLWord2pdf\bin> compile.bat

Now you have a Java executable that uses the evaluation version of Aspose.Words

Figure 1 If you have an Aspose license then place your Aspose. Words.lic in directory C:\SQLWord2pdf\bin

Edit file sqlword2pdf.java and remove the // comments.

```
License license = new License();
```

license.setLicense("Aspose.Words.lic");

Again run file compile.bat and check if the Java compilation created

a new class file at C:\SQLWord2pdf\bin\sqlword2pdf.class

Now test if the Java progam sqlword2pdf.class can create a .pdf file:

C:\SQLWord2pdf\bin> sqlword2pdf.bat example1a\_out.docx example1a\_out.pdf

Now you should see <a href="mailto:example1a\_out.pdf">example1a\_out.pdf</a> in directory <a href="mailto:C:\SQLWord2pdf\output">C:\SQLWord2pdf\output</a>

Now finally test if SQLWord now can generate C:\SQLWord2pdf\output\example1b\_out.pdf by running script test\_generate\_pdf.sql from the Oracle schema owner where you installed the SQLWord tables and packages

SQL> @test\_generate\_pdf.sql

## Apex implementation PDF expained

As already explained before there is a Custom event CustomEventRunSQLWord on page 1020 with a true action:

### Execute PL/SQL code

declare

```
l blob
                  blob;
 l_blob blob;
l filename varchar2(4000);
 v_where_clause varchar2(4000);
 v_report_id varchar2(4000);
begin
 if apex collection.collection exists('SQLWORD BLOB') then
   apex collection.delete collection('SQLWORD BLOB');
  end if;
 if :P0 PROCEDURE = 'HR EMPLOYEE SELECTION' then
    1 filename := 'EmployeeSelection-' || to char(sysdate,'dd-mm-yyyy-hh24.mi.ss') || '.' ||
                 lower(:P0_FILETYPE);
    ___
    v_where_clause := 'where 1=1 ' || apex_ir_query.ir_query_where( app_id_in
                                                                                          => :APP ID
                                                                     e( app_id_in => :APP_ID
, page_id_in => :APP_PAGE_ID
, session_id_in => :APP_SESSION
                                                                     , base report id in => :P1020 REPORT ID
                                                                     );
    hr employee selection( p where clause => v where clause);
  else
    select substr(lower(replace(first name || ' ' || last name || '.' ||
           lower(:P0_FILETYPE), ' ', '_')), 1, 4000)
    into l_filename
from employees
    where employee id = :P0 PK;
    if upper(:P0_PROCEDURE) = 'HR_EMPLOYEE_JOB_OFFER' then
      l filename := 'job offer ' || l filename;
     hr_employee_job_offer(p_employee_id => :P0_PK);
    elsif upper(:P0 PROCEDURE) = 'HR EMPLOYEE CONTRACT' then
      l filename := 'employee contract ' || l filename;
      hr_employee_contract(p_employee_id => :P0_PK);
    end if;
  end if;
  if upper(:PO FILETYPE) = 'PDF' then
    sqlword.save output pdf( p utl file location => 'SQLWORD OUTPUT DIR'
                            , p_pdf_filename => l_filename );
    l_blob := sqlword.get_output_pdf( p_utl_file_location =>'SQLWORD_OUTPUT_DIR'
                                    , p pdf filename => l filename);
  else
   l_blob := sqlword.get_output_docx;
  end if;
 apex_collection.create_or_truncate_collection(p_collection_name => 'SQLWORD_BLOB');
  apex collection.add member( p collection name => 'SQLWORD BLOB'
                             , p_c001 => l_filename
                             , p_blob001 => l blob );
  ---
end;
```

## Execute Javascript code

```
window.location = 'f?p=&APP_ID.:500:&APP_SESSION.:::::::;;
window.pageloadWait.remove();
```

Here we call page 500 a generic page for downloading output documents.

## Explanation

- Procedure SQLWORD.SAVE\_OUTPUT\_PDF is called to create the PDF-file in the directory on your Oracle server.
- The output document is stored in a local variable L\_BLOB by calling function SQLWORD.GET\_OUPUT\_PDF.
- An Apex collection is used to store the data.
- From the javascript we call page 500 where the PDF file will be downloaded.

## **Frequently asked questions**

## How can I create new pages in the output document?

You can call procedure SQLWORD.NEW\_PAGE to create a new page.

If you want to create new pages inside a for loop try this construction:

```
<%!
--
cursor cl
is
select ename
from emp
order by ename;
--
%>
<%for r1 in c1 loop%>
<%if cl%rowcount > 1 then sqlword.new_page; end if;%>
Employee: <%=r1.ENAME%>
<%end loop;%>
```

### How can I change the presentation of decimal values in the output document?

The presentation of decimal values in the output document depend on:

- The language settings from your Oracle database. Ask your DBA to check the value of parameter NLS\_LANGUAGE (select \* from V\$NLS\_PARAMETERS).
- The Microsoft Windows language settings on your PC. You can change this in the Windows Control panel in the "Regional and Language" options screen:

Standards and This option af	l formats fects how som	e programs fo	rmat numbers	. currencies.
dates, and tin	ne.			
<u>S</u> elect an iten your own form	n to match its p nats:	references, o	r click Custon	nize to choose
Dutch (Neth	erlands)		~	Customize
Samples				
Number:	123.456.789	,00		
Currency:	€ 123.456.78	9,00		
Time:	17:15:49			
Short date:	3-1-2005			
Long date:	maandag 3 ja	anuari 2005		
Location				
<u>I</u> o help servio weather, sele	ces provide you ct your present	i with local inf location:	ormation, suc	ch as news and
Netherlands,	The			*
2				

## How can I send an output document by email from my Oracle database?

SQLWord can send an email from the Oracle database with the output document as an attachment. SQLWord uses the Oracle UTL\_SMTP package:



## ACL-access

Using UTL\_SMTP to send email from your Oracle database has changed in Oracle 11g since accessing the remote network has changed. In Oracle 11g you have to configure (grant) each and every network access point using so called Access Control Lists (ACL's). Run SQL-script C:\SQLWord11\\SQL\ACL-access.sql as sysdba.

• Configure the <u>email settings</u> in the options screen from SQLWord Developer to your provider.

Lwora - Opti	ons	
icense Email	Client Server	
SMTP host	smtp.kpn.nl	Save
SMTP port	25	

• Edit SQL-script C:\SQLWord11\Examples\SQL\send\_email.sql and change the address for the email\_sender and email\_recipients.

```
begin
--
example1a(p_employee_id => 121);
--
sqlword.send_email( p_from => 'scott@tiger.com'
, p_to => 'my_email@gmail.com'
, p_subject => 'Hello, we send you a document
generated by SQLWord 11'
, p_text_msg => 'Hi,' || chr(10) || 'we send you a
document generated by SQLWord 11'
, p_file_name => 'example1a_out.docx'
);
--
end;
```

Start SQL\*Plus and run:

```
SQL> @ C:\SQLWord11\Examples\SQL\send_email.sql
PL/SQL-procedure successfully completed.
```

SQL>

Now check your email and see if there is a new email with an MSWord-document attached to it.

## How can I write an output document on the Oracle database server using UTL FILE?

Edit script C:\SQLWord11\Examples\SQL\write\_utl\_file.sql and modify the file locations to your environment.
 First you must use the Oracle "create directory" command (ask your DBA to do this).

- Start SQL\*Plus and run: <u>write utl file.sql</u>
   SQL> @write\_utl\_file.sql
   SQL> PL/SQL-procedure successfully completed.
- Check if the file is created on the specified file-location on your Oracle database server.

### How can I save the output document into an Oracle table?

The output document is available as a BLOB through function GET\_OUTPUT\_DOCX which is available in package SQLWORD.

First create a table where you want to save the generated output

Now call stored procedure EXAMPLE1A and immediately get the generated Word document by calling function SQLWORD.GET\_OUTPUT\_DOCX.

```
begin
   --
   example1a(121);
   --
   insert into my_output
   (        doc_name
   ,        dd_created
```

```
, source)
values
( 'EXAMPLE1A_' || to_char(sysdate,'ddmmyyyy-hh24:mi:ss')
, sysdate
, sqlword.get_output_docx);
___
commit;
___
end;
```

## Hints & Tips

## **Compile multiple documents**

You can quickly create stored procedures or recompile multiple source documents in one run.

Press the open file button from the SQLWord Developer toolbar and select the documents to compile:

Zoeken in:	Docx				
<b>1</b>	Naam 🔺		Gewijzigd op	Туре	
	🖳 Example1a	.docx	11-3-2015 10:00	Microsoft Of	fice
nelle toegang	Example1b	.docx	11-3-2015 10:00	Microsoft Of	fice
1 - 12	Example1c	.docx	11-3-2015 10:00	Microsoft Of	fice
Bureaublad	Example1d	.docx	29-5-2015 16:48	Microsoft Of	fice
	Example2.0	docx	11-3-2015 10:01	Microsoft Of	fice
-	Example3.	docx	18-1-2017 16:18	Microsoft Of	fice
Bibliotheken	Example4.	docx	11-3-2015 10:02	Microsoft Of	fice
	Example5.	docx	13-3-2015 17:37	Microsoft Of	fice
	Example6.	docx	13-3-2015 16:17	Microsoft Of	fice
Deze pc	HR_Dept.d	ocx	18-1-2017 16:19	Microsoft Of	fice
	HR_Dept_S	election.docx	18-1-2017 16:19	Microsoft Of	fice
Maturada	HR_Emplo	yee_Contract.docx	18-1-2017 16:19	Microsoft Of	fice
Nelwerk	HR_Emplo	yee_Job_Offer.docx	18-1-2017 16:19	Microsoft Of	fice
	HR_Emplo	yee_Selection.docx	18-1-2017 16:20	Microsoft Of	fice
	HR_Manag	er.docx	18-1-2017 16:20	Microsoft Of	fice
	HR_Tables	Report.docx	18-1-2017 16:20	Microsoft Of	fice
	•				
	Bestands <u>n</u> aam:	"HR_Tables_Report.docx" "H	HR_Dept.docx" "HR_Dept_Sel	ection.docx 🔻	Openen
	Bestandstypen:	Word doouments (* doo_* do	ov * #f)		Annuleren

### The next screen shows up

File	Stored procedure	Status	
C:\SQLWord11\Examples\Docx\HR_Dept_Selection.docx			Ureate al
C:\SQLWord11\Examples\Docx\HR_Employee_Contract.docx			Class
C:\SQLWord11\Examples\Docx\HR_Employee_Job_Offer.docx			CIUSE
C:\SQLWord11\Examples\Docx\HR_Employee_Selection.docx			
C:\SQLWord11\Examples\Docx\HR_Job.docx			
C:\SQLWord11\Examples\Docx\HR_Job_Selection.docx			
C:\SQLWord11\Examples\Docx\HR_Location_Selection.docx			
C:\SQLWord11\Examples\Docx\HR_Manager.docx	8		
C:\SQLWord11\Examples\Docx\HR_Tables_Report.docx			

Press the button "<u>Create all</u>" and the selected source documents are processed. If one or more is invalid after the compilation then the <u>color</u> of the row changes to <u>red</u>.

You can open a source document in Word by double-clicking on the row in the grid.

## **Clearing all scriptlets**

V

All your scriptlets should be without any formatting code.

You can clear all <% tag %> scriptlets from invisible underwater formatting code by pressing the <u>Clear all</u> button on the SQLWord Developer toolbar.

## Always place <% loop %> statements on a new line

To prevent a corrupt output document allways place loop statements on a new line.

Do not use constructions like this:

<%for r1 in c1 loop%><%=r1.department%><%end loop;%>

Use constructions like this:

<%for r1 in c1 loop%>

<%=r1.department%>

<%end loop;%>

## Placing <%loop statements %> in a word table

To prevent a corrupt output document allways place <% for statements%> on the <u>second row</u> from a word table and put the <% end loop;%> on the <u>last record</u>.

	Salary
<%=r2.job_title%>	<%=r2.sa lary%>
lob	Salary
	o analy
	Job