



PL/OFFX DOCX Getting Started V4.7.0



Oracle Reporting Made Simple

Contents

1.	Introduction.....	3
2.	First steps	3
3.	Creating your first DocX document	3



1. Introduction

PL/OFFX DOCX is a PL/SQL package collection that you can use to create Microsoft Word compatible DOCX files and process existing DOCX documents. This guide will help you from the beginning to generate your first PL/OFFX DOCX sample file.

2. First steps

First, you need to install an Oracle Database on your system. The easiest way is to download the [Oracle Express Edition](#). After you finished the installation, you can download the PL/OFFX software from the following link: [PL/OFFX download](#). To install the latest version of PL/OFFX please use the install guide, called ploffx_install_v400.pdf. You also need a PL/SQL Developer environment, or you can use the SQL Plus application and a text editor.

3. Creating your first DocX document

After you finished the installation start the PL/SQL Developer or a simple text editor and let's begin:

create or replace procedure AddText is

The l_docx variable is a requirement. This variable will contain the docx file after the generation is finished.

l_docx blob;

You can create a lot of paragraphs, but you will need the references of these. Now we will create only one paragraph, so one variable is enough. The type is always pls_integer.

***l_paragraph_id pls_integer;
begin***

When you want to create DOCX files with the PL/OFFX DOCX application, you need to start with the Init procedure. This procedure will initialize the whole application environment. It has got two optional parameters. The first is the p_unit, which specifies the measurement unit. Its default value is millimeter (mm). The second parameter is p_orientation which specifies the page orientation for the whole document. The default value is portrait. If the default values are right for you, the only line you need, this:

ploffx_docx.init();

Now, the PL/OFFX is initialized. The smallest independent unit in the document is the text, but every text is inside of a paragraph. Texts can't be outside of a paragraph. So, the next step is to create an empty paragraph. The crParagraph function is the solution for that. It has a lot of parameters to customize the paragraphs' properties. The following line is the minimal form of use.

l_paragraph_id := ploffx_docx.crParagraph();

If we have an existing paragraph, we can add some text to the document using the addText procedure. With that procedure we can fully customize our text's properties. The parameter types and possible values are described in the documentation. The following code snippet is the minimal form of the AddText. It requires an existing paragraph reference and some text.

ploffx_docx.addText(



Oracle Reporting Made Simple

```
p_paragraph_id => l_paragraph_id,  
p_text => 'Init example'  
);
```

The paragraph now exist and the text is added too but currently the paragraph is not the part of the document structure. It is only in the memory. You can add this paragraph with the addParagraph procedure. It has only one parameter called p_paragraph_id. Use the same reference what you used in the previous part (addtext).

```
ploffx_docx.addParagraph(p_paragraph_id => l_paragraph_id);
```

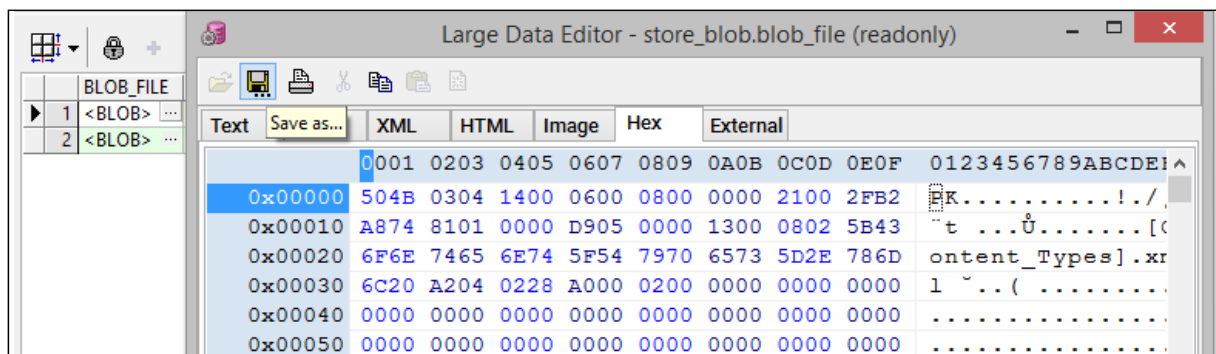
The sample document is finished. Now, you can generate the docx file into a simple blob variable. The getDoc procedure will help you in that case. It has only one parameter which specifies the target blob variable for the result file.

```
ploffx_docx.getDoc(p_blob_file => l_docx);
```

This is an optional step. After the file generated, you need to store it. The PL/OFFX installkit provides you a sample store table, called STORE_BLOB. The stored blob is a fully compatible docx file.

```
insert into store_blob(blob_file,created_date,filename)  
values (l_docx,sysdate,'Addtext.docx');  
commit;
```

If you use PL/SQL Developer, you can download the file. When you save it to a local file system use the qualified file name, for example: AddText.docx.



The generated example file in Microsoft Word:

