



# Reporting in your Oracle database

---

## **PL/PDF** Upgrade Guide from v2 to v4 **V4.1.0**



# Reporting in your Oracle database

---

## Contents

1.	Introduction.....	3
2.	Differences .....	3
3.	Migration steps.....	4
4.	PL/SQL code details .....	4



# Reporting in your Oracle database

---

## 1. Introduction

PL/PDF is a PL/SQL package collection that can be used to create PDF files. This guide will help you migrate your PL/PDF Generator v2 to PL/PDF v4. PL/PDF v4 is our brand new development, our team rewrote all PL/SQL codes for extended performance and productivity. Main benefits and features of PL/PDF v4 are:

- High performance - the PDF generation speed has increased to 10x since v2
- DOCX template - The MS Word based layout designer increases productivity
- New rich text formatting - You can use MS Word like texts in PDF the creation
- TrueType font - The new font subset makes smaller PDF files
- New Barcodes - QRCode, DataMatrix and AztecCode are new 2D barcodes in PL/PDF v4
- New Chart engine - MS Word like chart creation
- New Cell and Row printing with rich text formatting
- Header and Footer processing for better performance

## 2. Differences

We tried to preserve compatibility with our v2 product but we had to make some modifications. Here's the full list of changes:

- plpdf.row\_print\* procedures got removed, use plpdf\_row\_print package
- plpdf.setRtoL got removed and plpdf\_rtol is not supported anymore
- plpdf\_toc: moved TOC related procedures to the plpdf\_toc package
- plpdf.getPageNoAlter: new function
- plpdf\_acroform: moved the Acroform related procedures to plpdf\_acroform package
- CurrentPageNumber was renamed to getCurrPageNum
- plpdf\_pattern: moved PATTERN related procedures to plpdf\_pattern package
- CertKeyCheck renamed to checkCertKey
- plpdf.GetPrintFontKey new function
- plpdf\_annot: moved Annotation related procedures to the plpdf\_annot package
- p\_clipping number default 1: modified p\_clipping boolean default to true
- p\_fill number default 0: modified p\_fill boolean default to false
- setCompress changed to setNOCompress
- nopAlias: removed, use getCurrPageNum, getNumOfPages
- setJS changed to addJavaScript
- crInternalLinkDest new function
- StartClipping: removed p\_end\_prev parameter
- isInClipping: new function
- DefaultTemplate renamed to setPageDefaultTemplate
- useTemplate renamed to setPageTemplate
- plpdf.linebreak and plpdf.newLine are not the same



# Reporting in your Oracle database

## 3. Migration steps

Migration from v2 to PL/PDF v4 is not only a PL/SQL package replacement, we recommend the steps below:

1. Install PL/PDF v4 into a new separated user
2. Copy PL/PDF tables from v2 to v4. You can move the full v2 tables to v4 because v4 tables are compatible with v2 ones. You can use the standard Oracle export/import or any other third party tool.
3. Update v4 sequences: update start value of the PLPDF\_TEMPLATE\_S and PLPDF\_TTF\_S to v2 next values, use the ALTER SEQUENCE command.
4. Copy the developed PL/PDF related PL/SQL codes and other objects to v4.
5. Change the required codes (see the differences in the previous section)
6. Test reports and PDF outputs
7. Switch reporting access from v2 to v4

## 4. PL/SQL code details

Here are the details of the required modifications.

	Old code	v4 conform code
plpdf.row_print*	plpdf.Row_Print (	plpdf_row_print.Row_Print (
plpdf.setRtoL	plpdf_rtol.	-
plpdf toc	plpdf.AddTOC (	plpdf toc.AddTOC (
plpdf.getPageNoAlter	-	plpdf.getPageNoAlter
plpdf_acroform	plpdf.AcroForm_AddTextFie ld(	plpdf_acroform.addTextFie ld(
CurrentPageNum ber	plpdf.CurrentPageNumber	plpdf. getCurrPageNum
plpdf_pattern	plpdf.SetDashPattern (	plpdf_pattern.SetDashPatt ern(
CertKeyCheck	plpdf.CertKeyCheck	plpdf.checkCertKey
plpdf.GetPrintFont Key	--	plpdf.GetPrintFontKey
plpdf_annot	plpdf.AddTextAnnot (	plpdf_annot.AddTextAnnot (
p_clipping	<pre> plpdf.PrintCell(   p_w =&gt; 70,   p_h =&gt; 10,   p_border =&gt; '1',   p_txt =&gt; text',   p_clipping =&gt; 0 ); </pre>	<pre> plpdf.PrintCell(   p_w =&gt; 70,   p_h =&gt; 10,   p_border =&gt; '1',   p_txt =&gt; text',   p_clipping =&gt; false ); </pre>
p_fill	<pre> plpdf.PrintCell(   p_w =&gt; 70,   p_h =&gt; 10,   p_txt =&gt; 'No fill',   p_fill =&gt; 0 ); </pre>	<pre> plpdf.PrintCell(   p_w =&gt; 70,   p_h =&gt; 10,   p_txt =&gt; 'No fill',   p_fill =&gt; false ); </pre>
setCompress	plpdf.setCompress();	plpdf.setNOCompress();
nopAlias	plpdf.nopAlias;	plpdf_cell.printCell(



# Reporting in your Oracle database

	<pre>plpdf.PrintCell(   p_w =&gt; 0,   p_h =&gt; 10,   p_txt =&gt; to_char(plpdf.Current PageNumber)    '/{nb}',   p_border =&gt; '0',   p_ln =&gt; '0',   p_align =&gt; 'C' );</pre>	<pre>p_width =&gt; plpdf.getPageAvailableWid th,   p_height =&gt; null,   p_text =&gt; to_char(plpdf.getCurrPage Num)    '/'  to_char(plpdf.getNum OfPages()),   p_ln =&gt; plpdf.const.beside,   p_align =&gt; plpdf3.c_center );</pre>
setJS	plpdf.setJS(	plpdf.addJavaScript(
crInternalLinkDest	-	plpdf.crInternalLinkDest
StartClipping	<pre>plpdf.StartClipping(   p_x =&gt; 20,   p_y =&gt; 15,   p_w =&gt; 45,   p_h =&gt; 60,   p_end_prev =&gt; true );</pre>	<pre>plpdf.StartClipping(   p_x =&gt; 20,   p_y =&gt; 15,   p_w =&gt; 45,   p_h =&gt; 60 );</pre>
isInClipping		plpdf.isInClipping
DefaultTemplate	plpdf.DefaultTemplate(	plpdf.setPageDefaultTempl ate(
useTemplate	plpdf.useTemplate(	plpdf.setPageTemplate(
plpdf.linebreak and plpdf.newLine are not the same	-	-