

OQC quick guide for PDF template creation

<http://www.sugarforge.org/projects/openqc/>

By AndriusRin

Table of Contents

1 Introduction	3
2 Tools.....	3
2.1 Platform.....	3
2.2 Software	3
3 General notes.	3
3.1 Smarty variables.....	5
4 Example: Creating simple Quote template	7
4.1 Document architecture.	7
4.2 Creating page layout with OOo Writer.	7
4.3 Exporting to LaTeX using Writer2Latex plugin.....	7
4.4 Fine tune page appearance with TeXstudio.	8
4.5 Use Smarty to embed Quote data into template.....	8

1 Introduction

The PDF creation process is briefly described in OQC manual. However, many important details are missing and purpose of this manual is to provide these missing details as well as examples how pdf template can be created.

OQC uses LaTeX for creation of PDF files. So, at least minimal knowledge of LaTeX syntax is necessary. Excellent online resource about LaTeX can be found here : <http://en.wikibooks.org/wiki/LaTeX> . For quick reading I would recommend “The Not So Short Introduction to LaTeX” that can be found here: <http://ctan.tug.org/tex-archive/info/lshort/english/lshort.pdf> . These two resources explains quite well all the advantages and drawbacks You are getting when using LaTeX.

2 Tools

2.1 Platform

In this manual I will be using Windows XP (after all, it most used operating system at the moment). All of the tools required are available also for most of the other Windows or Linux flavors.

2.2 Software

I used following open-source or free software:

- OpenOffice.org, version 3.3. (<http://www.openoffice.org/>).
- Writer2Latex extension for OpenOffice - <http://extensions.services.openoffice.org/project/writer2latex> allows to convert files created with OpenOffice Writer into LaTeX code.
- MikTeX distribution can be downloaded from here: <http://www.miktex.org/> . Current stable version is 2.9. This package includes all tools required for compiling/displaying LaTeX code.
- LaTeX code editor. There are number of editors available, I used TeXstudio, that is available here: <http://texstudio.sourceforge.net/#home> . It has build-in compiler, syntax checker and other nice features plus it integrates with MikTeX easily.

Please look for installation instructions for each of the program on their respective web sites.

3 General notes.

The template files are located in /include/oqc/Pdf directory. Each template is put into separate directory; selection of templates are done from Editview of each module. The images that are used in templates should be placed in /include/oqc/Pdf/Graphics directory.

The file that determines the structure of the pdf document is Contract.tpl for Contrarcts, Offering.tpl for Quotes and ProductCatalog.tpl for Product Catalog. Take closer look to Offering.tpl.

As one can see, it is mixup of latex and Smarty code. As Smarty delimiters, < and > symbols are used (if You need to use > or < in Smarty expressions, use gt or lt instead).

The part of code from the beginning to \begin{document} is called *Preamble*. In this part the general settings of the document can be changed - document class, fonts, spacings – and all other relevant settings.

After Preamble, the document structure is determined. As one can see it cosist from three parts (the forth part is attachments, but it is not shown in the code). At the beginning the title page will be included, then Textbloks. them Product List Tables. At the end of the file Attachments will be added.

The user can modify structure of document by changing the ordering or by removing/including the parts of the document.

```
% Preamble- this part of file determines global parameters of document
% scrartcl is KOMA script document class- please refer to KOMA script documentation.

<if "it_it" == $default_language>

% Italian
\documentclass[10pt,a4paper,italian,parskip,headsepline,DIV14]{scrartcl}
\usepackage[italian]{babel}

<else> <if "ge_ge" == $default_language>
```

```

% German
\documentclass[10pt,a4paper,german,parskip,headsepline,DIV14]{scrartcl}
\usepackage[ngerman]{babel}

<else>

% English
\documentclass[10pt,a4paper,english,parskip,headsepline,DIV14]{scrartcl}
\usepackage[english]{babel}

</if>
</if>

\usepackage[utf8]{inputenc} % this sets input file encoding; it is recommended to
use utf-8 for all new projects
\usepackage[T1]{fontenc} % Set required fonts encoding; refer to Latex Manual for
more info
\usepackage{longtable}
\usepackage{booktabs}
\usepackage{scrpage2}
\usepackage{typearea}
\usepackage{graphicx,ifthen} % allows graphics insertion and manipulation
\usepackage{eurosym}
\usepackage{pdflscape}
\usepackage{ifpdf}
\usepackage{fixltx2e} % allows subscript
\usepackage[normalem]{ulem} % allows underline and strikeout

\pagestyle{scrheadings}

% NOTE: this requires graphicx and ifthen to be loaded!!
% if image width does not exceed \textwidth it should be displayed in original size.
otherwise its width should be scaled to \textwidth
% to use this functionality use \pitpic instead of \includegraphics when inserting
your images!

\newlength{\testwd}
\newcommand{\fitpic}[1]{%
\settowidth{\testwd}{\includegraphics{#1}}%
\message{#1 width=\the\testwd, page=\the\textwidth}%
\ifthenelse{\lengthtest{\testwd>\textwidth}}{%
\noindent\includegraphics[width=\textwidth]{#1}}{%
\centering\includegraphics{#1}\par}%
}

% decrease left page margin and increase page width. this is useful for wide tables
and images
\addtolength{\textwidth}{+0.7in}
\addtolength{\textheight}{+0.7in}
\addtolength{\oddsidemargin}{-0.2in}

% match html paragraph indent width
\setlength{\parindent}{0in}

% End of preamble

\begin{document}

<if isset($titlePage)>
\newpage
<fetch file=$titlePage>
</if>

<if isset($textBlocks)>
\newpage
<fetch file=$textBlocks>
</if>

<if isset($services)>
\newpage

```

```
<fetch file=$services>
</if>
```

```
% After services the attachments are included that have appropriate settings of
document purpose field. If You do not want to attach files, set document purpose to
"For internal use".
\end{document}
```

3.1 Smarty variables.

Smarty is used to populate templates with the data from SugarCRM modules.

The available Smarty variables (only most important are listed, in general, all database fields are available) are listed in the table below. Please note, these variables are available in all parts of the document: in Titlepage.tpl, Textbloks.tpl, and Services.tpl.

There is one exception, though. When Smarty Variables are embedded directly into TextBlocks (not .tpl files), one should use {{ and }} as Smarty delimiters.

Please look into template files (especially Services.tpl) for usage hints of these variables.

Quote/Contract variables	Quote/Contract client variables
\$contract.name \$contract.date_modified \$contract.description \$contract.type \$contract.status \$contract.priority \$contract.abbreviation \$contract.svnumber \$contract.startdate \$contract.enddate \$contract.version \$contract.quote_leadtime (for Quotes only) \$contract.shipment_terms \$contract.payment_terms \$contract.quote_validity (for Quotes only) \$contract.warranty \$contract.currency_id \$contract.deadline \$contract.currency_symbol	\$contract.clientContact.description \$contract.clientContact.salutation \$contract.clientContact.first_name \$contract.clientContact.last_name \$contract.clientContact.title \$contract.clientContact.department \$contract.clientContact.phone_home \$contract.clientContact.phone_mobile \$contract.clientContact.phone_work \$contract.clientContact.primary_address_street \$contract.clientContact.primary_address_city \$contract.clientContact.primary_address_state \$contract.clientContact.primary_address_postalcode \$contract.clientContact.primary_address_country \$contract.clientContact.assistant \$contract.clientContact.assistant_phone \$contract.clientContact.account.name \$contract.clientContact.account.description \$contract.clientContact.account.account_type \$contract.clientContact.account.industry \$contract.clientContact.account.phone_fax \$contract.clientContact.account.billing_address_street \$contract.clientContact.account.billing_address_city \$contract.clientContact.account.billing_address_state \$contract.clientContact.account.billing_address_postalcode \$contract.clientContact.account.billing_address_country \$contract.clientContact.account.phone_office \$contract.clientContact.account.phone_alternate \$contract.clientContact.account.website \$contract.clientContact.account.ownership \$contract.clientContact.account.shipping_address_street \$contract.clientContact.account.shipping_address_city \$contract.clientContact.account.shipping_address_state \$contract.clientContact.account.shipping_address_postalcode \$contract.clientContact.account.shipping_address_country
SugarCE user issuing pdf document variables	General variables (sum values of YUI tables)
\$contract.userContact.description \$contract.userContact.salutation \$contract.userContact.first_name \$contract.userContact.last_name \$contract.userContact.title \$contract.userContact.department \$contract.userContact.phone_home \$contract.userContact.phone_mobile \$contract.userContact.phone_work	\$graphicsDir \$imagesExist \$OngoingCostsNetTotal \$OngoingCostsVAT \$OngoingCostsGrossTotal \$OngoingMonths \$ServicesOnceNetTotal \$ServicesOnceVAT \$ServicesOnceGrossTotal

\$contract.userContact.phone_fax \$contract.userContact.primary_address_street \$contract.userContact.primary_address_city \$contract.userContact.primary_address_state \$contract.userContact.primary_address_postalcode \$contract.userContact.primary_address_country \$contract.userContact.email_address \$contract.userContact.account.name \$contract.userContact.account.description \$contract.userContact.account.phone_fax \$contract.userContact.account.billing_address_street \$contract.userContact.account.billing_address_city \$contract.userContact.account.billing_address_state \$contract.userContact.account.billing_address_postalcode \$contract.userContact.account.billing_address_country \$contract.userContact.account.phone_office \$contract.userContact.account.website \$contract.userContact.account.shipping_address_street \$contract.userContact.account.shipping_address_city \$contract.userContact.account.shipping_address_state \$contract.userContact.account.shipping_address_postalcode \$contract.userContact.account.shipping_address_country \$contract.userContact.account.parent_id	
Product list variables (from YUI table), {n} is number of the YUI table row, starting from zero (0,1,2,3...).	Recurring expenses variables (from YUI table), {n} is number of the YUI table row, starting from zero (0,1,2,3...).
\$oneTimeServices.{n}.name \$oneTimeServices.{n}.date_modified \$oneTimeServices.{n}.modified_by_name \$oneTimeServices.{n}.description \$oneTimeServices.{n}.oqc_product_number \$oneTimeServices.{n}.type \$oneTimeServices.{n}.status \$oneTimeServices.{n}.price \$oneTimeServices.{n}.oqc_relatedcategory \$oneTimeServices.{n}.oqc_vat \$oneTimeServices.{n}.personincharge \$oneTimeServices.{n}.assigned_employee \$oneTimeServices.{n}.unit \$oneTimeServices.{n}.catalog_id \$oneTimeServices.{n}.catalog_name \$oneTimeServices.{n}.price_text \$oneTimeServices.{n}.monthsguaranteed \$oneTimeServices.{n}.cancellationperiod \$oneTimeServices.{n}.image_mime_type \$oneTimeServices.{n}.position \$oneTimeServices.{n}.quantity \$oneTimeServices.{n}.discount_value \$oneTimeServices.{n}.zeitbezug \$oneTimeServices.{n}.discount_select \$oneTimeServices.{n}.zeitbezug_translated \$oneTimeServices.{n}.discounted_price \$oneTimeServices.{n}.discounted_price_tax_free \$oneTimeServices.{n}.currency_symbol \$oneTimeServices.{n}.image_url \$oneTimeServices.{n}.has_image \$oneTimeServices.{n}.startdate \$oneTimeServices.{n}.enddate	\$services.{n}.name \$services.{n}.date_modified \$services.{n}.modified_by_name \$services.{n}.description \$services.{n}.oqc_product_number \$services.{n}.type \$services.{n}.status \$services.{n}.price \$services.{n}.oqc_relatedcategory \$services.{n}.oqc_vat \$services.{n}.personincharge \$services.{n}.assigned_employee \$services.{n}.unit \$services.{n}.catalog_id \$services.{n}.catalog_name \$services.{n}.price_recurring_text \$services.{n}.monthsguaranteed \$services.{n}.cancellationperiod \$services.{n}.image_mime_type \$services.{n}.position \$services.{n}.quantity \$services.{n}.discount_value \$services.{n}.zeitbezug \$services.{n}.discount_select \$services.{n}.zeitbezug_translated \$services.{n}.discounted_price \$services.{n}.discounted_price_tax_free \$services.{n}.currency_symbol \$services.{n}.image_url \$services.{n}.has_image \$services.{n}.startdate \$services.{n}.enddate

4 Example: Creating simple Quote template

If You are not familiar with the LaTeX, I would not recommend to try to write template from the scratch. Fortunately, there are tools available that makes learning of LaTeX much easier.

The chapters below describes creation of simple custom Quote template. I tried to keep it as simple as possible. All the resulting .tpl files can be found in /include/oqc/Pdf/Offering2 directory; source and intermediary files are located in /include/oqc/Pdf/Offering2/Sources .

4.1 Document architecture.

At first You have to decide how the document should look like. For this example I want to have one page Product List YUI table content (Recurring expenses Table is not required) followed by TextBlocks describing Quote terms and conditions in more detail, then followed by attachments of the Quote that will likely include more detailed description of the Products.

NOTE: Even if You do not use some parts of the default document structure, the corresponding. tpl files should stay in template directory. In our case, the TitlePage.tpl is included into directory, even if it is not used.

4.2 Creating page layout with OOo Writer.

Using OOo Writer I can create how the page should look like. See the result at /include/oqc/Pdf/Offering2/Sources/Quote template_2.odt . Few advices regarding use of Writer.

1. Using LaTeX it is difficult to squeeze a lot of text into single page. This is because LaTeX automatically sets all the formatting for easy reading. There is one exception, though, - tables. I organized all information on the page as one big table. Such a approach also helps to position the text in the exact location You want.
2. I do not recommend to use headers and footers at this stage. Writer2Latex converter is not very good in converting them. If needed, headers/footers can be added at later stage, directly into LaTeX code. Refer to LaTeX manual regarding use of headers and footers, for example, here: http://en.wikibooks.org/wiki/LaTeX/Page_Layout#Customising_with_fancyhdr .
3. Try to avoid graphics resizing. I would recommend to edit graphics in the way that it would fit the whole page width. For example, the picture incuded into this example has 867 pixels width at 96 dpi resolution. Simple calculation ($867 / 96 * 25.4 = 229.39\text{mm}$) shows that width of picture is similar to the width of A4 page in landscape orientation (297 mm).

4.3 Exporting to LaTeX using Writer2Latex plugin.

The next step is to export page into LaTeX. After You install Writer2Latex plugin, under File->Export... an option to export file to .tex format appears. The settings used for export can be seen in Figure 1.

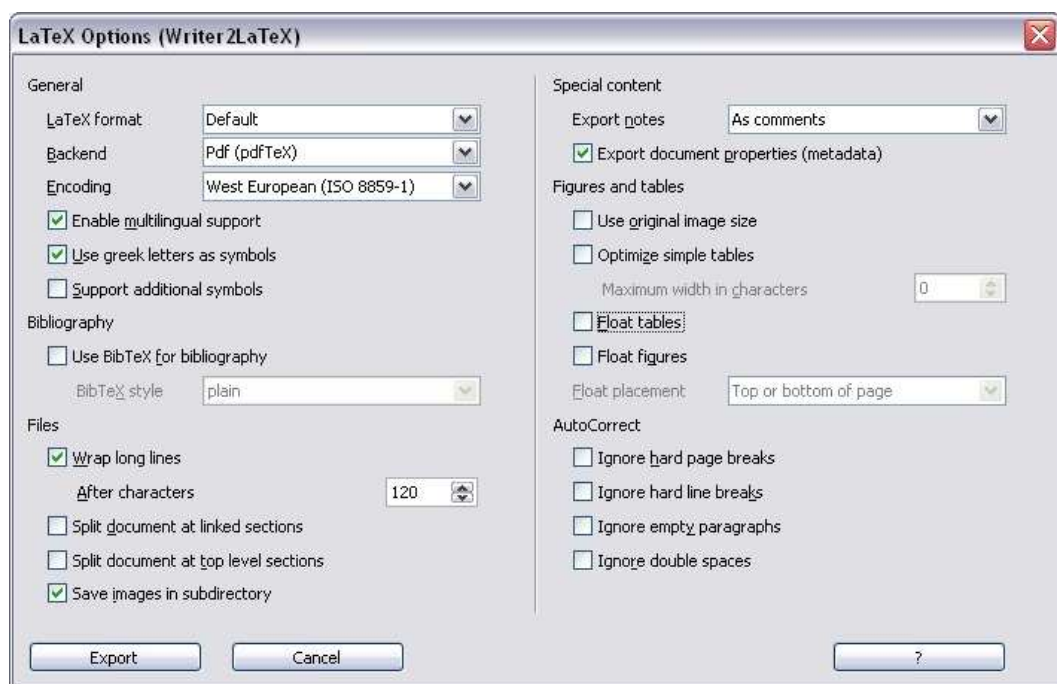


Figure 1. Writer2LaTeX settings for export.

4.4 Fine tune page appearance with TeXstudio.

Open produced .tex file with TeXstudio. You can create and view pdf file by pushing PDFLATE/AdobeReader button on menu ribbon. It might look not exactly as expected ☺, so fine tuning of TeX file might be necessary.

In addition, one can make changes that is required for better compatibility with the rest of Quote document. For this particular example document the following changes were made in preamble section:

1. `\documentclass` command was modified to use KOMA-Script class (<http://developer.berlios.de/projects/koma-script3/>) which has more possibilities for page layout customisation. In addition default font changed to 10pt (more text can fit on page).
2. Added `\usepackage[english]{babel}` command for english hyphenation.
3. Modified `\usepackage[utf8]{inputenc}` command utf8 instead of latin1. It is recommended to use utf8 encoding for new projects for better compatibility.
4. Added following extra packages :
 - `\usepackage{fixltx2e}` % allows subscript
 - `\usepackage[normalem]{ulem}` % allows underline and strikethrough
 - `\usepackage{eurosym}` %this is required for euro symbol display
 - `\usepackage{pdflscape}` % enables landscape page orientation
5. Since we changed default page orientation from landscape to portrait, it is necessary to adjust page layout- this is done with `\setlength` commands.

When You do any modification to the LaTeX file, at any step You can check how these changes will influence the resulting PDF file appearance by using TeXstudio.

4.5 Use Smarty to embed Quote data into template.

The document section that is started after `\begin{document}` command was moved to Services.tpl file.

The next task is to modify LaTeX code to include Smarty variables that are described in chapter 3.1. In addition, for creation of rows of the table I used *foreach* construct.

After smarty code is included, it will be no longer possible to visualise changes using TeXstudio. One has to copy new files to SugarCRM instance and try to create PDF documents.

If something goes wrong, one can look into tmp directory on Linux systems, where the final assembled .tex file should be present (it might not have .tex extension, though). On Windows systems, the file will be located in C:/Temp/pdflatex/ directory. You can open this file with TeXstudio and check if there are any errors.