OpenQuotesAndContracts Manual

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

Version 2.2RC3

Lion Vollnhals, Ingo Jaeckel
AndriusRin

Table of Contents

1 Introduction

This document describes the installation process and the use of the SugarCRM module OpenQuotesAndContracts - also known as *openqc* [].

Four students have started the openqc-development in winter 2007/2008 at the Hasso-Plattner-Institut [3] which is situated in Potsdam, Germany. It is build together with a department called LDS[2], which provides IT-services for other German departments.

The aim of openqc is making business processes more transparent, increase productivity and improve service quality.

It is developed by two students of the Hasso-Plattner-Institut at the moment - the authors of this manual. Together with the components that are integrated into openqc (e.g. FckEditor, dhtmlxTree), openqc consists of more than 35000 lines of PHP and 65000 lines of JavaScript code.

At the time of writing OpenQuotesAndContracts has the following features:

- The creation of PDF files from contracts, quotes, contract additions and product catalogs. Documents can be attached (PDF, DOC) or embedded (all other formats) into the PDF file.
- The management of products, quotes, contracts and contract additions (see chapter 5).
    - Automatic versioning of products, quotes, contracts and contracts additions.
    - Creation of contracts from quotes.
    - Creation of contracts additions from contracts.
    - Integration of the SugarCRM autocompletion feature for selection of contacts and employees.
    - Select products to produce and configure lists of services that are sold in a contract.
    - Create product options that can be included into Quotes and Contracts only together with “master” product.
    - Select and edit textblocks with the integrated WYSIWYG editor: TinyMCE.
    - Upload documents that give further details about quotes or contracts.
- The management of product catalogs (see chapter 6).
    - Create product categories and change their hierarchies using drag and drop.
    - Create PDF file of the product catalog that includes products assigned to the categories tree.
    - Ability to generate pricelist PDF file of Products in different currencies, with applied discount or markup.

We appreciate for all kind of feedback to openqc but also to this manual. From summer of 2011 the openqc community forum at SugarForge is disabled. You can leave message on general SugarCRM forum here: http://www.sugarcrm.com/forums/f19/openquotesandcontracts-76169/ or send us e-mail (see Developers Tab on SugarForge for e-mail).

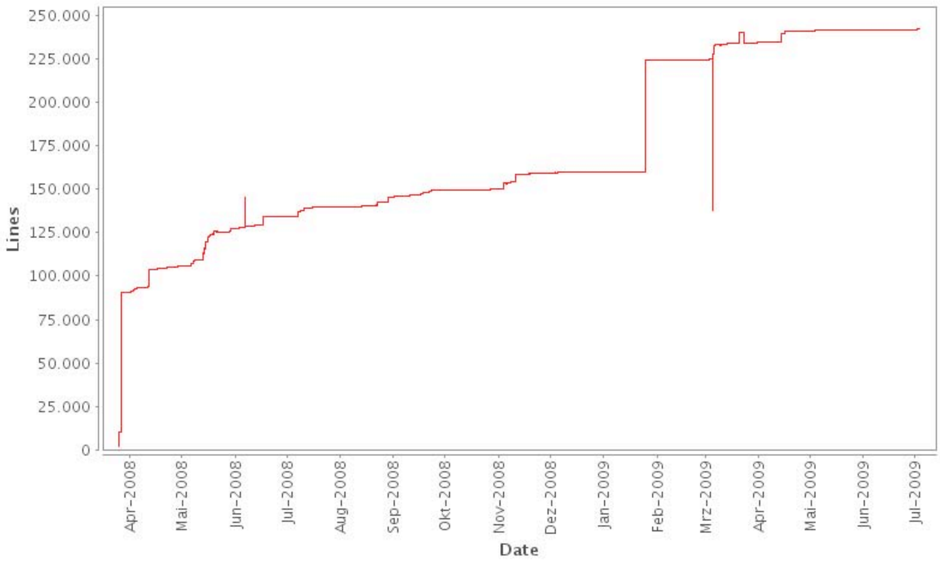


Figure 1: Lines of code of the openqc-project

2 Requirements

2.1 Client

- Firefox, Opera and Internet Explorer 7/8 (in IE8 mode). Internet Explorer 8 support appeared only from 1.7.7 version and might be not complete. The best rendering of UI is done by Firefox or Opera.

2.2 Server

- SugarCRM 6.0.x to SugarCRM 6.5.x .You probably have to edit some openqc source files to use openqc with another version of SugarCRM. **Note:** openqc 2.1 works correctly on SugarCRM 6.3 and higher when AjaxUI is disabled for OQC modules. OQC installer disables AjaxUI for OQC modules by default.
- pdftk
- pdflatex
- DocumentConverter.py
- OpenOffice.org with headless-support and python-uno-bindings
- Python
- Linux/Windows/MacOSX: We develop and test openqc for using it on Linux. However, all features but the document conversion (.doc -> .pdf) works on Windows as well. Please see PDF template creation manual for configuration on Windows machines.

3 Installation

3.1 Installation of the OpenQuotesAndContracts ZIP file

You can install OpenQuotesAndContracts using the module loader that comes with SugarCRM. This requires the following steps:

1. Download the current ZIP file from the openqc website [].
2. Upload the file into SugarCRM using the module loader (see figure 2).

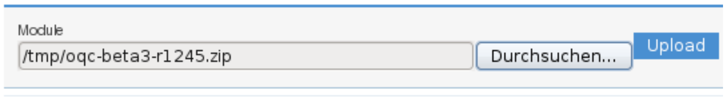


Figure 2: openqc ZIP file opened in the module loader.

3. Click on the “Install” button and accept the license (see figure 3).

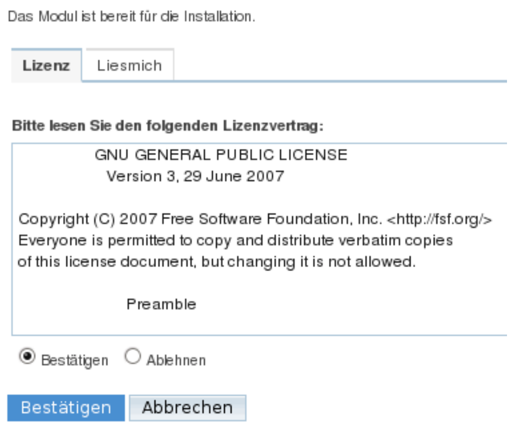


Figure 3: Starting the installation process

Everything but the pdf creation feature should work immediately after the installation. Activation of the pdf creation feature is discussed in the 3.3 section.

3.2 Updating OpenQuotesAndContracts to latest version.

One should uninstall the previous version of OQC before installing the latest version. When uninstalling, You will be prompted to choose if You want to remove Database Tables. Chooses “Do not remove” if You want to keep all the OQC data.

If uninstall was successful proceed with installation of latest version as described in 3.1.

3.3 Getting the pdf creation feature working

OQC uses pdflatex to generate pdf files. pdflatex is part of TexLive package that is available for all major operating systems: Linux, MacOSX and Windows.

pdftk program can be used optionally (if installed), for manipulation of created pdf files. It can merge several pdf files into one, include (attach) other files into final pdf file etc.

OpenOffice or LibreOffice programs allow to convert doc/odt or other text files into pdf automatically, which later can be included into resulting pdf file using pdftk.

While pdflatex is mandatory for getting pdf, other programs are optional.

The installation procedure for above mentioned software packages is specific to the deployed operating system, because package managing tools and package names differ. In chapters 3.3.1- 3.3.5 we briefly will describe general program installation flow for major operating systems. Chapter 3.3.7 describes pdf creation debug mode which You can use to troubleshoot issues.

Chapters 3.4 and 3.5 describes operating system and OQC setup in cloud environment (AmazonEC2). Create Sugar-on-Demand instance by yourself ☺

3.3.1 OpenSuse 11

You should have enabled the OpenSuse and the Packman repositories so that zypper can find the tools we want to install.

```
# zypper in python
# zypper in OpenOffice_org-pyuno pdftk texlive texlive-latex
# zypper in openoffice.org-writer openoffice.org-headless openoffice.org-core
# zypper in texlive-latex-extra texlive-base texlive-common
```

3.3.2 CentOS 5

All packages but pdftk can be installed from the default repositories using yum. To install pdftk you have to download a CentOS-compatible rpm from `http://dag.wieers.com/rpm/packages/pdftk/`. Depending on your architecture you should download the 32-bit or 64-bit version.

```
# yum install openoffice.org-headless openoffice.org-pyuno libstdc++
# yum install libgcj tetex tetex-latex tetex-fonts tetex-dvips
# wget http://dag.wieers.com/rpm/packages/pdftk/pdftk-1.41-1.el5.rf.i386.rpm
# rpm -ihv pdftk-1.41-1.el5.rf.i386.rpm
```

Also check the chapter 3.6 where detailed guide of CentOS configuration is included. For CentOS You will need to edit OQC config file (accessible from Admin panel) and set CentOSpatch =1 for pdf creation to work.

3.3.3 Ubuntu

All required packages can be easily installed by using apt. Also see chapter 3.3.6 for installation using pre-configured system images on AmazonEC2 cloud.

```
apt-get install python pdftk
apt-get install texlive-full
apt-get install openoffice.org-headless python-uno
```

3.3.4 MacOSX

First You have to choose MAMP stack You will work with. The instructions below are valid when using AMPPS from Softaculuos Inc., which can be downloaded from http://www.ampps.com/downloads.

After downloading and installing AMPPS, make sure that required php extensions are enabled (goto PHP->Php extension), and make sure than active PHP version is 5.2.x or 5.3.x. My list of enabled extensions is following (not all of them are really required): bz2.so; curl.so; gd.so; imap.so; ldap.so; mbstring.so; mysqli.so; pdo_mysql.so; pdo_sqlite.so; soap.so; sqlite.so; zip.so; zlib.so.

Install SugarCE and OQC. Installation is straightforward. SugarCE is available from AMPPS home menu under "ERP" entry. Push install button and follow setup instructions. After installation You should be able to access Your SugarCE instance using browser from 127.0.0.1/sugar/ . OQC then can be installed trough ModuleLoader.

TexLive packages for MacOSX are available from http://tug.org/mactex/. Full package is quite large, more than 2.3 GB when compressed, so my suggestion is to use smaller one called BasicTeX.

Download and install BasicTeX from http://mirror.ctan.org/systems/mac/mactex/mactex-basic.pkg

You probably will also need TeX editor. TexShop is good choice, which can be downloaded from here: http://pages.uoregon.edu/koch/texshop/texshop.html .

For pdf files manipulation download and install pdftk (now called pdftk Server) from http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/pdftk_server-2.02-mac_osx-10.6-setup.pkg .

You can also download some extra fonts, if You are planning to use them in templates. This is not required for default templates that are included together for OQC, though. Download and run installer mactex-additions from http://mirror.ctan.org/systems/mac/mactex/mactex-additions.pkg . When installing these, install only fonts (do not install Ghostscript and GUI tools). For this when You reach in "Installation Type" page, push button "Customise" and select checkboxes accordingly.

The versions of the programs are updated quite often, so if links above does not work, navigate to required files manually. Total size of on disk of all the software required for pdf creation is ~300 MB in this configuration.

Some of packages are not provided with BasicTex, but are required for OQC. You have to install them from terminal.

First, update the package manager by executing from the terminal:

```
sudo tlmgr update --self
sudo tlmgr install eurosym
sudo tlmgr install supertabular
sudo tlmgr install ulem
sudo tlmgr install titlesec
```

If some package are not installed there will be message like:

```
! LaTeX Error: File `ulem.sty' not found.
```

in log file and pdf creation will fail.

To see log file You have to set debugPdfCreation = 1 in config file which is accessible from admin panel.

Even when all the packages are in place, when trying to generate pdf file, pdflatex might fail with error like this:

```
!pdfTeX error: /usr/texbin/pdflatex (file ecsx1440): Font ecsx1440 at 600 not found.
```

This is quite often generated error, so we will discuss it more extensively.

When pdflatex is run for the first time, it needs to build fonts bitmaps that will be used for creating of pdf files. For creating bitmaps it uses external programs. The error above means that there is some issue with usage of these extrenal programs. Typical causes of this behavior are:

1. The bitmap creation program could not be found if PATH variable does not contains folder where it is located. PATH variable can be checked by running phpinfo() or diagnostics from SugarCE admin panel Administartion->System->Diagnostic Tool, see Apache Environment section. For MacOSX PATH should contain /usr/texbin entry.

2. The apache does not have enough permissions to access programs. Depending on the system You might need to modify the permissions for folders and files that are required for pdf creation.

3. Bitmap creation programs might not have enough permissions to write generated output to the files in some particular directory. Typically it tries to write to some folder in users HOME directory, for example, for MacOSX it is ~/Library/texlive. If user that runs apache does not have HOME directory, this causes the error, as is the case for CentOS (set CentOSpatch = 1 in config file to avoid this problem).

For this particular setup on MacOSX that was discussed above, the error is caused by /usr/texbin not being included into apache PATH variable.

To rectify the problem, execute from terminal:

```
sudo touch /etc/launchd.conf
sudo nano /etc/launchd.conf
```

In the editor that will open add following:

```
setenv PATH /usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/texbin
```

Save the changes and reboot MacOSX for these changes to take place.

3.3.5 Windows XP/Windows 7.

On Windows the users will be not able to convert and merge attached MS Word documents or pictures into the PDF file. Instead, documents will be included into resulting pdf file as “attachments”. However, if attached document is already in PDF format, they will be merged into final pdf file if pdftk program is installed.

Pdflatex is included into MikTeX distribution that can be downloaded from here: http://www.miktex.org/. Current stable version is 2.9.

The installer will ask You whether You want to update program “on the fly”. Select Yes from the dropdown.

Pdftk package can be downloaded from here: http://www.pdflabs.com/tools/pdftk-the-pdf-toolkit/ . Get PDFtk Server, which is command line tool, current version is 2.0.2 (September of 2013).

After installation of packages, check where executables of pdflatex and pdftk are installed, go to Admin->Open_Quotes_And_Contracts->Edit_config_file page and modify pathPdflatex and pathPdftk keys according actual executable file location. By default OQC is configured to use 64-bit version of pdflatex on Windows 7.

When pdflatex will be run for the first time, it will update the programs and download missing packages from the internet. So, make sure that computer has access to the internet.

First run might take some time, depending on speed of computer and internet connection. For the first run, it is recommended to switch on pdf creation debug mode in config file:

```
debugPdfCreation = 1
```

and configure Apache/PHP to accept longer script execution times. For this, edit “Resource limits” section of php.ini file

```
max_execution_time = 180
max_input_time = 180
memory_limit = 256M
```

and restart Apache server for these changes to take effect.

After the first successful run return to default values in php.ini.

The OpenOffice configuration section (chapter 3.3.6) can be skipped for Windows systems, since DocumentCoverter and other related software is not ported yet for Windows.

3.3.6 OpenOffice or LibreOffice document converter configuration.

NOTE: The method described below is outdated as of September of 2013. Some forums also reported that OOo produces memory leaks that eventually leads to the crash of the system. We are planning to rewrite OQC to use unoconv tool (http://dag.wieers.com/home-made/unoconv/) and/or LibreOffice command line interface for document conversion function.

Download the DocumentConverter script [] from http://artofsolving.com.

```
wget http://artofsolving.com/files/DocumentConverter.py
```

NOTE: Instead of OpenOffice, LibreOffice can be installed. LibreOffice package comes with required document conversion tools (DocumentConverter.py for example).

Copy all necessary files into a directory which is in $PATH (e.g. /usr/local/bin/). The DocumentModifier.py file can be found in the tools directory of the openqc ZIP file as well.

```
cp DocumentConverter.py /usr/local/bin/
cp DocumentModifier.py /usr/local/bin/
```

Update the permissions of these files.

```
chmod +x /usr/local/bin/DocumentConverter.py
chmod +x /usr/local/bin/DocumentModifier.py
```

Start OpenOffice/LibreOffice each time you start your SugarCRM server.

```
$openoffice_path/soffice "-
accept=socket,host=localhost,port=8100;urp;StarOffice.ServiceManager"
 -norestore -nofirststartwizard -nologo -headless &
```

$openoffice_path depends on Linux and OpenOffice flavor. For example, on 64-bit OpenSuse 11.0 the $openoffice_path is /usr/lib64/ooo-2.0/program , while on 32-bit OpenSuse 11.3 $openoffice_path is /usr/lib/ooo3/basis3.2/program .

NOTE: For LibreOffice the starting of headless server should be done with following command:

```
$openoffice_path/soffice -accept="socket,port=8100;urp;" -norestore -
nofirststartwizard -nologo -headless &
```

Locate documents.properties file in SugarCRM_root/include/oqc/conf directory. The following entries should be modified depending on actual installation path of pdf creation tools:

```
pathDocumentConverter = "PYTHONPATH=/usr/lib64/ooo-2.0/program python
/usr/local/bin/DocumentConverter.py"
pathPageInserter = "PYTHONPATH=usr/lib64/ooo-2.0/program python
/usr/local/bin/DocumentModifier.py"
```

The example above is typical for OpenSuse 64-bit environment. For 32-bit environments and OO 3.2 the typical paths are:

```
pathDocumentConverter = "PYTHONPATH=/usr/lib/ooo3/basis3.2/program python
/usr/local/lib/DocumentConverter.py"
pathPageInserter = "PYTHONPATH=usr/lib/ooo3/basis3.2/program python
/usr/local/lib/DocumentModifier.py"
```

3.3.7 Debugging pdf creation.

Since 2.2RC3 there is debugging mode of pdf creation. It can be switched on by setting entry debugPdfCreation = 1 in config file which is accessible from admin panel (Edit Config File entry).

When debugging mode is activated, instead of pdf file OQC will produce log file that consists from two parts:

1. Latex source file the pdflatex is using for pdf creation. You can Copy/Paste it into Tex editor of Your choice and use it for manual debugging of latex code.
2. The pdflatex console(terminal) output together with pdflatex log file, pdftk version information.

When pdf creation fails, there should be some messages like:

```
   ! LaTeX Error: File `ulem.sty' not found.
   ! pdfTeX error: /usr/texbin/pdflatex (file ecsx1440): Font ecsx1440 at 600 not
 found.
```

that would help to locate problem source.

Some useful information also can be found php_error.log file of your PHP module and at the SugarCRM log file sugarcrm.log. You have to increase the log level of SugarCRM to "Error" to get more detailed information by changing the logging settings in the Admin->System Settings page to Error (by default it is set to Fatal).

If you need more help, feel free to post a message at SugarCRM forum. We will try to help you as fast as we can.

3.4 Installing SugarCRM and OQC in cloud environment (Bitnami+AmazonEC2)

At the moment of writing (March of 2012) Amazon is providing one micro instance for free testing, while Bitnami is offering system images that includes SugarCRM 6.4.2. The LAMP stack includes Ubuntu 10.04, Apache 2.2, PHP 5.3.10 and MySQL 5.5.16 .You can start configuration process of Bitnami cloud from here: http://bitnami.org/cloud/signup .

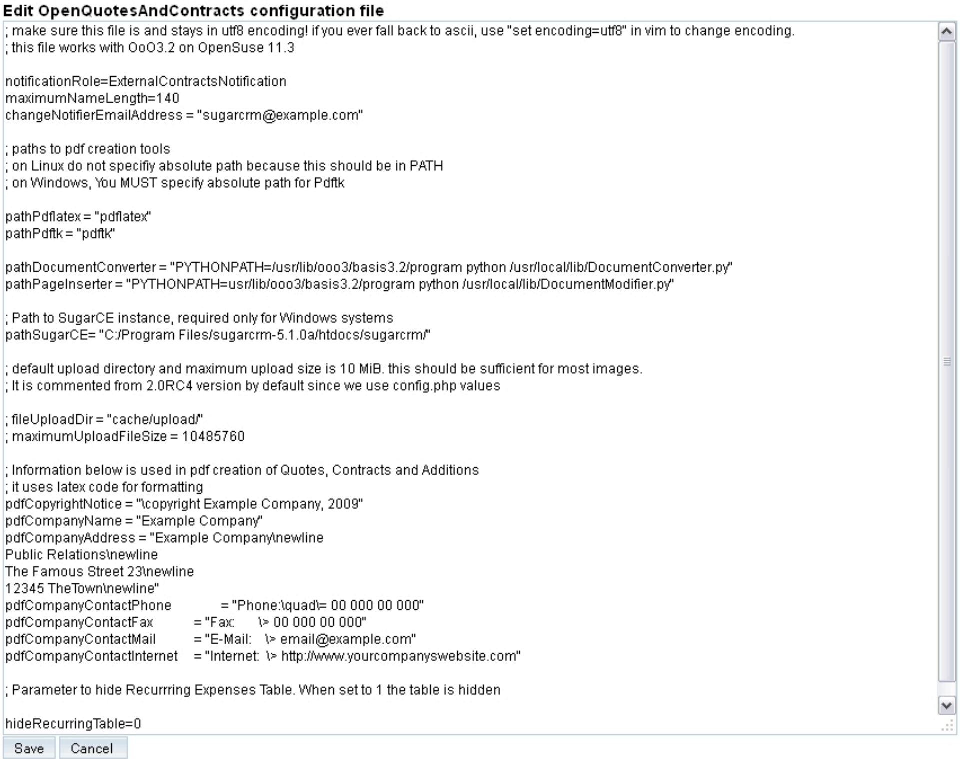
We will skip the steps that are required to setup SugarCRM since they are extensively described in the materials available on Bitnami and Amazon Cloud websites. Briefly, You have to sign for Amazon Cloud, and then configure it using Bitnami web-site.

We assume, that You went through configuration process, and have SugarCRM instance operating, accessible from Your computer.

1. First step is to install OQC as described in 3.1 chapter.

2. Starting from 2.1 version You can configure internal OQC settings from Admin screen.

“Edit config file” link opens configuration file editor. Most of the settings are configured by default. You can enter Your company address that will be used for Quotes/Contracts/ProductCatalog pdf file creation and decide if You want to hide Recurring Expenses table or not (see chapter 3.6):



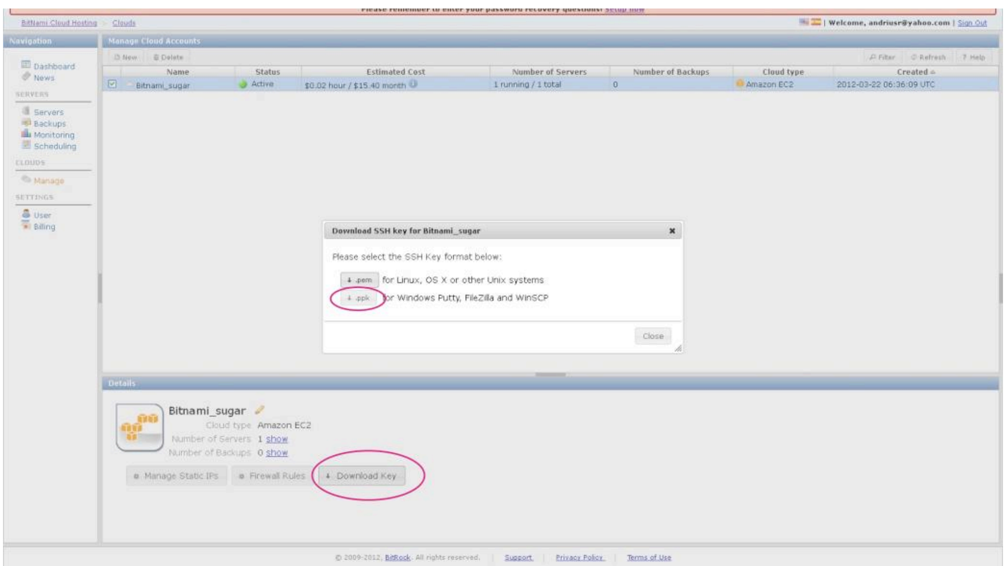
Edit OpenQuotesAndContracts configuration file

```
; make sure this file is and stays in utf8 encoding! if you ever fall back to ascii, use "set encoding=utf8" in vim to change encoding.
; this file works with OoO3.2 on OpenSuse 11.3

notificationRole=ExternalContractsNotification
maximumNameLength=140
changeNotifierEmailAddress = "sugarcrm@example.com"

; paths to pdf creation tools
; on Linux do not specifiy absolute path because this should be in PATH
; on Windows, You MUST specify absolute path for Pdftk

pathPdflatex = "pdflatex"
pathPdftk = "pdftk"

pathDocumentConverter = "PYTHONPATH=/usr/lib/ooo3/basis3.2/program python /usr/local/lib/DocumentConverter.py"
pathPageInserter = "PYTHONPATH=usr/lib/ooo3/basis3.2/program python /usr/local/lib/DocumentModifier.py"

; Path to SugarCE instance, required only for Windows systems
pathSugarCE= "C:/Program Files/sugarcrm-5.1.0a/htdocs/sugarcrm/"

; default upload directory and maximum upload size is 10 MiB. this should be sufficient for most images.
; It is commented from 2.0RC4 version by default since we use config.php values

; fileUploadDir = "cache/upload/"
; maximumUploadFileSize = 10485760

; Information below is used in pdf creation of Quotes, Contracts and Additions
; it uses latex code for formatting
pdfCopyrightNotice = "\copyright Example Company, 2009"
pdfCompanyName = "Example Company"
pdfCompanyAddress = "Example Company\newline
Public Relations\newline
The Famous Street 23\newline
12345 TheTown\newline"
pdfCompanyContactPhone          = "Phone:\quad\= 00 000 00 000"
pdfCompanyContactFax        = "Fax:      \> 00 000 00 000"
pdfCompanyContactMail       = "E-Mail:   \> email@example.com"
pdfCompanyContactInternet   = "Internet: \> http://www.yourcompanyswebsite.com"

; Parameter to hide Recurrring Expenses Table. When set to 1 the table is hidden

hideRecurringTable=0
```

Save | Cancel

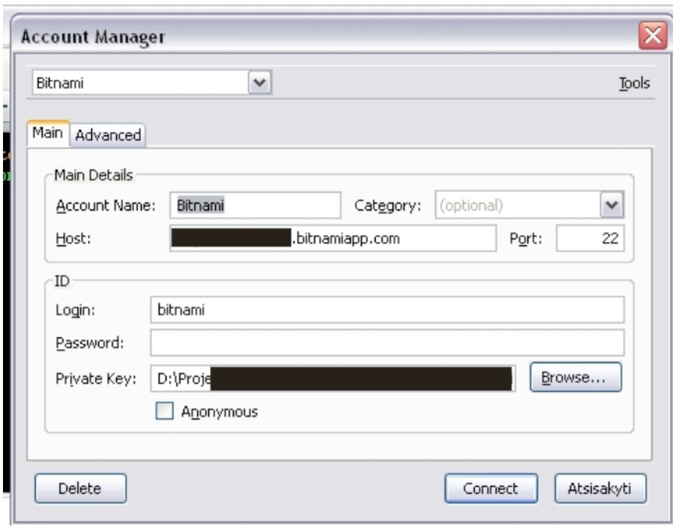
3. For getting pdf creation and scheduler to work, You have to setup SSH client for Your Linux instance access. For example, you can use Putty or FireSSH addon for Firefox. For following steps I used FireSSH.

At first, download private key from Bitnami:



Save .ppk file somewhere in Your computer system.

Next, open FireSSH and configure connection:



where Host is Your SugarCRM instance public IP address, and Private Key- location of Your downloaded .ppk file.

4. When You successfully connected to Linux instance using FireSSH, install pdf generation tools. Execute following commands:

```
sudo apt-get install pdftk
sudo apt-get install texlive-full
```

After completion of installation, You should be able to produce “basic” Quote/Contract/ProductCatalog pdf files. Attachments to the Quotes/Contracts that are in pdf format will be attached as well.

More advanced configuration involves installation of OpenOffice and other auxiliary files that would allow You to convert and attach MS Word and other files to the “basic” Quote/Contract/ProductCatalog pdf file. Please review 3.3 chapter for instructions if You want this extra feature.

5. **TeamTask** module requires crontab job to be setup in order to send reminders when tasks are overdue. The instructions given in Bitnami Wiki did not worked for me. It should be noted, that commands below depends on Linux configuration. When different Linux versions are used, You have to check if file paths, user names and permissions are the right ones. It is important to setup crontab job for the same user who is running apache server. You can check the correct user name by looking at the owner name of any OQC file once it is installed in Your SugarCRM instance.

At first, we have to add own user, bitnami, to daemon group in order to have the same file access rights as apache server (daemon user runs apache). This would allow to access SugarCRM instance files from ftp client like FireFTP. Execute following:

```
sudo usermod -a -G daemon bitnami
```

Log off and back in from FireSSH. Check if Your user is included in daemon group with:

```
groups
```

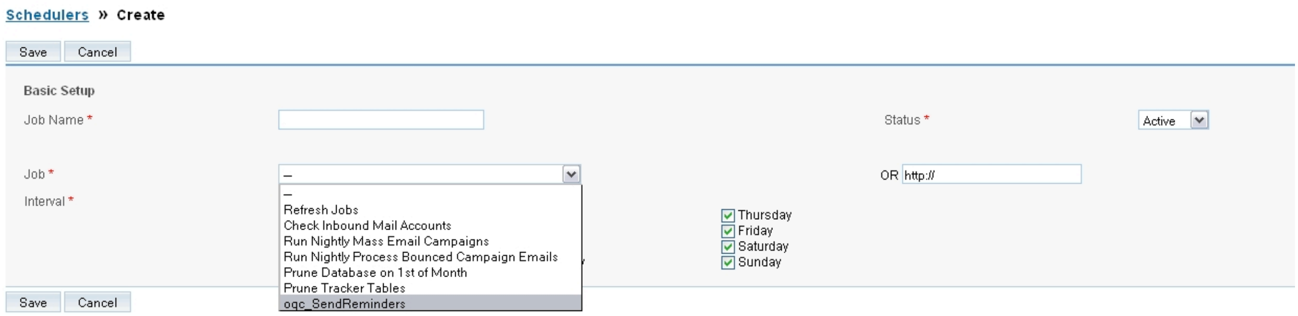
Next, edit daemon crontab (daemon is user that runs apache):

```
sudo crontab –u daemon –e
```

In crontab editor add following line:

```
* * * * * cd /opt/bitnami/apps/sugarcrm/htdocs; /opt/bitnami/php/bin/php -c
/opt/bitnami/php/etc/php.ini -f cron.php > /dev/null 2>&1
```

Lastly, You have to setup Scheduler job oqc_SendReminders from Administration screen:



3.5 CentOS configuration (Amazon EC2).

Configuration procedure described below is for CentOS 6.3 image downloaded from AWS Marketplace in March of 2012, named CentOS-6.3-x86_64-GA. Right now it is not available at AWS Marketplace (current version is 6.4), however, You can find it by searching AMIs from AWS management console using keyword "CentOS-6.3-x86".
This image is build with minimal profile and has only SSH server installed.
In the guide below we will use least resistance method to install all programs required to run OQC and pdf creation tools. The production servers should use more secure configurations of iptables, selinux, apache and phpMyAdmin. So, please read statement below…

NOTE: DO NOT USE THE INSTRUCTIONS BELOW FOR SETUP OF YOUR PRODUCTION SERVER! THE SOLE PURPOSE OF THIS GUIDE IS TO TEST AND DEMONSTRATE OPERATION OF OQC PDF CREATION TOOLS RUNNING UNDER CENTOS 6.3 .

1. Before launch of the instance, You have to configure the security group it will belong to. Once instance launched, You will not be able to change security group, so You might not be able to access required services. Default security group is Ok to use, since it has ports 22, 80 and 443 open. However, if You are planning to use also email client or other services, You will need to create Your custom security group that would open standard ports mentioned above AND extra ports You want to use. For Gmail that would be 993 (IMAP) and 587 (SMTP), for example. Then, when starting Your instance, associate it with security group You just created.
2. Next You have to configure Your SSH client of choice to access Your instance. See 3.4 chapter above for some hints how to do this. The user name You have to use for access is root.
3. Configure iptables. Since we are using AWS build-in firewall, we can allow full access in iptables. The "#" sign in the commands below means command prompt of SSH console, so You do not need to enter it. The lines without "#" are not commands, usually these lines are what You need to enter in nano or other editor.

```
# iptables -P INPUT ACCEPT
# iptables -F
# iptables -P FORWARD DROP
# iptables -P OUTPUT ACCEPT
```

Save rules

```
# /sbin/service iptables save
```

4. Configure SELinux. Since we are following least resistance route, we will disable SELinux, so it will not interfere with installation or running of other programs.
To disable SELinux, execute:

```
# setenforce 0
```

This command will disable SELinux till next reboot. To disable it permanently, after You install nano editor (see step 5 below), run:

```
# nano /etc/selinux/config
```

If You are planning to use SELinux at the later time, edit line SELINUX=enforcing to be

```
SELINUX=permissive
```

or, if You have no plans to use SELinux

```
SELINUX=disable
```

and save the changes (***using key sequence Ctrl+X -> Y -> Enter).***

5. Get useful utilities

```
# yum install wget
# yum install nano
# yum install unzip
```

6. Now install AMP stack:

```
# yum install httpd
# yum install php php-mysql php-gd php-imap php-ldap php-odbc
# yum install php-pear php-xml php-xmlrpc php-mbstring
# yum install mysql mysql-server mysql-libs
```

Start Apache and configure it to start at boot

```
# /etc/init.d/httpd start
# chkconfig --levels 235 httpd on
```

Start MySQL and configure it to start on boot.

```
# /sbin/service mysqld start
# chkconfig --levels 235 mysqld on
```

Set MySQL password, You will need it at later time when configuring Your SugarCRM instance, so remember it.

```
# mysqladmin -u root password 'somepassword'
```

If everything went right, now You should be able to access Apache start page at www.yourinstance.com. (See Public DNS for the address in Description tab in EC2 Management Console.

7. Before installing SugarCRM, we will need to configure Apache and PHP accordingly.
PHP configuration is done by editing php.ini file. Set the timezone (use Your location here):

```
# nano +946 /etc/php.ini
date.timezone = Europe/Paris
```

After each file edit with nano editor, save changes using key sequence Ctrl+X -> Y -> Enter.

Increase memory limit for php script. 192M is minimum requirements for pdf creation tools to work. If You will be creating large product catalogs, it might be required to set it larger:

```
# nano +457 /etc/php.ini
memory_limit = 192M
```

Increase upload file size:

```
# nano +878 /etc/php.ini
upload_max_filesize = 30M
```

Increase POST max size:

```
# nano +729 /etc/php.ini
post_max_size = 12M
```

Set apache server time zone:

```
# ln -sf /usr/share/zoneinfo/Europe/Paris /etc/localtime
```

Allow htaccess file modification, edit line 338 of httpd.cong:

```
# nano +338 /etc/httpd/conf/httpd.conf
AllowOverride All
```

Make the configuration changes take effect.

```
# service httpd restart
```

8. Now we are ready to install SugarCRM:

```
# cd /var/www/html
# wget http://www.sugarforge.org/frs/download.php/10754/SugarCE-6.5.15.zip
# unzip SugarCE-6.5.15.zip
# mv SugarCE-Full-6.5.15 sugarcrm
# rm SugarCE-6.5.15.zip
```

Now You need to setup file permissions properly. There are detailed documentation available on SugarCRM site how to do this, for example here: http://support.sugarcrm.com/04_Find_Answers/02KB/02Administration/100Install/Required_File_System_Permissions_on_Linux .

```
# chown -R apache:apache /var/www/html/sugarcrm
# find /var/www/html/sugarcrm -type d -exec chmod 755 {} \;
# find /var/www/html/sugarcrm -type f -exec chmod 644 {} \;
```

Now You should be able to access Sugar instance from http://www.yourinstance.com/sugarcrm/index.php. The SugarCRM installation wizard will start and will ask You to enter several database (and other) configuration parameters. Use following settings:

```
Database host = localhost
Database Administrator Username = root
Database Administrator password = somepassword (the same as You entered for MySQL).
```

TeamTask module requires crontab job to be setup in order to send reminders when tasks are overdue. At first we configure to use nano editor for editing crontab:

```
# export EDITOR=nano
```

Now edit crontab:

```
# crontab -u apache -e
```

Enter following line if you want to execute scripts every minute:

```
* * * * cd /var/www/html/sugarcrm; php -f cron.php > /dev/null 2>&1
```

Or, your can enter following for script execution every ten minutes:

```
*/10 * * * *  cd /var/www/html/sugarcrm; php -f cron.php > /dev/null 2>&1
```

Other useful modification is to direct all PHP errors to sugarcrm.log. This way it is possible to collect more information about issues You might be having and make troubleshooting a lot easier.

```
# nano +634 /etc/php.ini
error_log = /var/www/html/sugarcrm/sugarcrm.log
```

Save changes and restart apache for these changes to take effect:

```
# service httpd restart
```

9. Pdf creation tools setup requires extra repositories to be available. pdftk install requires RPMForge repository:

```
# cd /tmp
# wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
# yum install rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
# yum install pdftk
# yum install texlive texlive-latex
```

Install LibreOffice if You are planning to conver documents (see chapter 3.3.6 for details).

```
# yum install libreoffice-writer libreoffice-headless
```

10. Now You are ready to install OQC. See chapter 3.1 for instructions.
After installation, You have to edit OQC config file from Administration panel. Set

```
CentOSpatch =1
```

11. The last step is optional. Below are installation instructions for phpMyAdmin, useful database manipulation tool. It is not required for daily operation.
Enable epel repository:

```
# cd /tmp
# wget http://mirror-fpt-telecom.fpt.net/fedora/epel/6/i386/epel-release-6-8.noarch.rpm
# yum install epel-release-6-8.noarch.rpm
```

```
# yum install phpmyadmin
# chown -R apache:apache /usr/share/phpMyAdmin
# chmod -R 755 /usr/share/phpMyAdmin
```

Vanilla permissions are very restrictive and allows access only from localhost(127.0.0.1), so we need to edit several other files in order to be able to access instance from the web. Run

```
# nano /etc/httpd/conf.d/phpMyAdmin.conf
```

Find the entries as printed below and and modify them to look like this:

```
<Directory /usr/share/phpMyAdmin/>
  Order Allow,Deny
  Allow from All
</Directory>

<Directory /usr/share/phpMyAdmin/setup/>
  Order Allow,Deny
  Allow from All
</Directory>
```

Copy sample config file to real one

```
# cp /usr/share/phpMyAdmin/config.sample.inc.php /usr/share/phpMyAdmin/config.inc.php
```

and change authentication type to http:

```
# nano +29 /usr/share/phpMyAdmin/config.inc.php
$cfg['Servers'][$i]['auth_type'] = 'http';
```

Now try to connect to http://www.yourinstance.com/phpmyadmin. You will be asked for password. Enter user name root and password “somepassword”- the same as You entered for MySQL (see step 6).
This completes installation of OQC on CentOS.

3.5 Configuration of VAT (Tax) and other dropdown lists.

Most of the customization of the OQC can be done trough Studio. Prior to 2.x the VAT value was set in document.properties file in line `vat=0.19`. This setting is no longer supported in 2.x version. You have to configure VAT list in Studio.
Open Developer Tools -> Dropdown Editor and select oqc_vat_list for editing. By default OQC ships with following configuration:

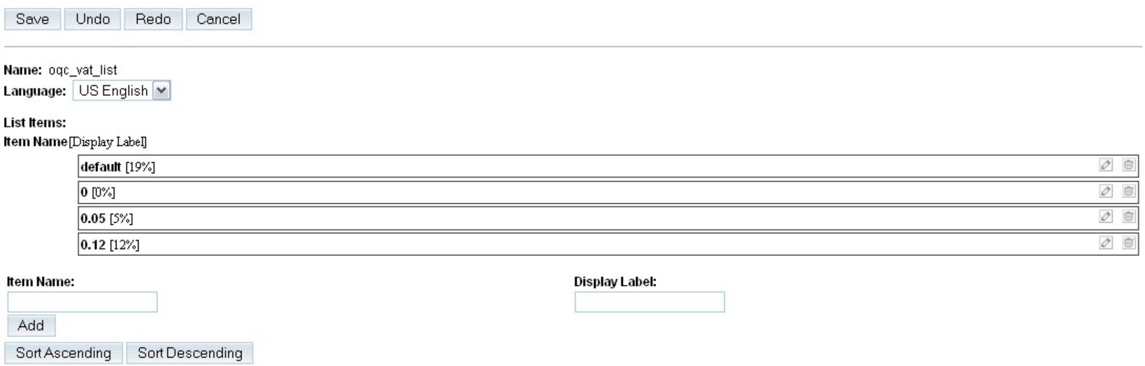


Figure 3a: Default VAT list configuration

Default of VAT should be set to the most often used value, while when changing other values one should follow the format as shown in the picture. The Item name should be the real number of VAT value while Display label – percentage of VAT.
For example, if one wants to change default to 0%, the resulting list will looks like this:

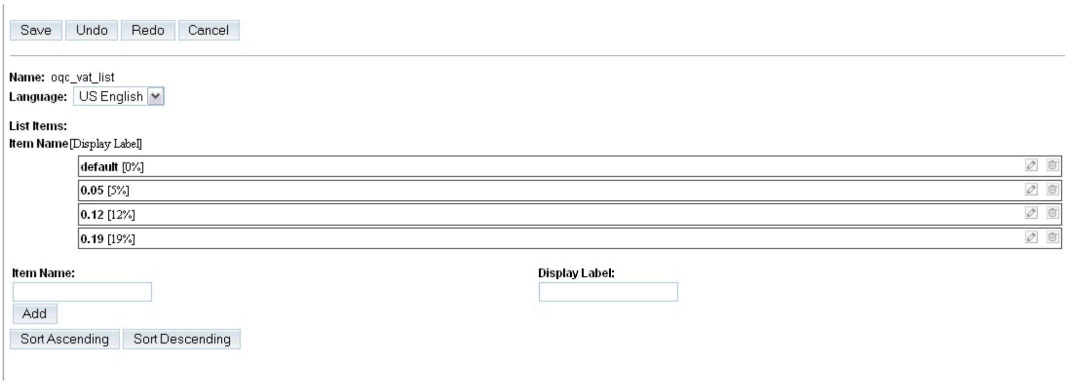


Figure 3b: VAT configuration after editing

Other dropdowns related to the OQC modules are unit_list or shipment_terms_list that also can be edited here. For unit_list it is important to keep index “pieces”intact since javascript code depends on availability of this index.

3.6 Enable/disable Recurring Expenses table in Quotes/Contracts module.

If desired, the Recurring Expenses table can be made invisinble in Quotes/Contracts DetailView and EditView. You can edit configuration from Administration page. Set hideRecurringTable=1 parameter for table to be hidden.

3.7 Clear Firefox Cache

After you installed a new OQC zip file on the SugarCRM server you have to clear the Firefox cache (Ctrl+F5) on all clients accessing the SugarCRM installation.

A new version of OQC changes files that are usually stored in the browser cache after downloading them the first time. The old files have to be removed from the cache. New versions of the files will be downloaded when the updated server is accessed by the client the first time.

When you cleared your cache and accessed the updated server again your browser should use the updated JavaScript files from the new zip file.

3.8 Known Issues

When some information-rich (many versions, products, attachments, subpanels etc.) products/quotes/contracts are saved, You may see the message “Error: Query limit of 1000 reached for oqc_xxx module.”. In this case You need to increase query limit in config.php file. Find [“resource_management][“default_limit”] key in config.php and increase value to 2000 (default value is 1000).

Italian and German translation for new Teamtask module is not available. Translation of some other modules (Products) can be better… Can somebody help with translation?

Hovering in Product ListView on colored records does not works out-of-the box. To make it work one has to edit themes/Sugar5/css/style.css file.

Find the section as below and add !important:

```
.evenListRowS1:hover td,
.oddListRowS1:hover td
{background: #f6f6f6 !important;}
```

For Modern Aqua theme (themes/ModernAqua/css/style.css):

```
.evenListRowS1:hover td { background: #DEEAFD !important; }
.oddListRowS1:hover td { background:  #F6F6F6 !important; }
```

3.9 Uninstalling OQC.

Before uninstalling OQC You should remove oqc_SendReminders Scheduler job, otherwise the system might generate errors since the functions that are required to perform this job will be no longer available.

4 Restrictions

4.1 Formatting Texts

If you are filling in the description fields and want to use enumerations or itemizations, you should use the built-in formatting feature. Copying text from other sources, such as Microsoft Word, into the description fields will insert special characters that are not interpreted correctly. This can result in problems during the pdf creation process. That is why you should avoid copying text from other sources into the description fields that have formatting support.

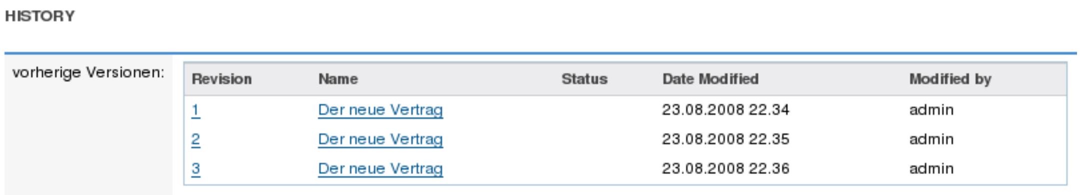
Hint: TinyMCE editor has button for importing of the text from MS Word. Use it at Your own risk. In most cases it works well.

5 Products/Quotes/Contract Management

OpenQuotesAndContracts allows the user to manage products, quotes, contracts and contract additions. We suggest the following workflow.

1. Create Product Catalog (see chapter 6 for details) and populate it with Product Categories.
2. Create Products. This is not necessary step since the quotes and contracts can be created with on-the-fly made products, but it is highly recommended since Products can be described and characterized in much greater detail.
3. Create a Quote. The quotes can be created in several ways, for example from Quotes subpanel when user is in Opportunity DetailView or from Quotes ListView by pushing button “Create New Quote”. In the former case the relationship between Opportunity and Quote will be created automatically.
4. Create a Contract from the previously created quote. The contracts can be also created as stand-alone documents, without links to previously created quotes.
5. Create a contract Addition from the previously created contract.

All changes that are done in the modules, are automatically controlled by an integrated versioning system.


HISTORY

vorherige Versionen:

Revision	Name	Status	Date Modified	Modified by
1	Der neue Vertrag		23.08.2008 22.34	admin
2	Der neue Vertrag		23.08.2008 22.35	admin
3	Der neue Vertrag		23.08.2008 22.36	admin

Figure 4: Versioning of a contract

It is possible to create pdf files from the quotes and contracts all the time.

In the next sections we will describe the basics of modules usage.

5.1 Products

Products have the following main attributes that can be entered or modified in Edit view of the Product:

- Product name- can be entered when in Edit view.
- Product code.
- Price.
- Currency.
- “Is Recurring Expenses?” attribute – only products marked with this attribute can be included into “Recurring Expenses List” table of Quotes and Contracts.
- Version- new version of Product is created every time when modification is done to the attributes of the Product. Only latest version of the Product is displayed in the List view, earlier versions can be accessed from History panel in Product Detail view.
- “Active” tag- only the products that are set active can be entered into Quotes and Contracts.
- “Is Product Option?” attribute- Products, that are marked with this attribute can be included into Quotes or Contracts only together with “master” Product. To associate Product as option with another Product, use “Add Option” button in Product EditView.
- Position in Product Catalog- this attribute determines ordering of products in created ProductCatalog pdf file. Starting from 2.2RC1 version this field is removed from EditView by default. The ordering

can be changed in ProductCatalog module, see chapter 6 for details. Natural order algorithm is used for sorting, products with higher position appears higher. It is not necessary to create exact numbered list, for example products with numbers 3, 15 and 100 will be ordered in the order 3,15,100.

- Publishing state- only products that are set published are included into Product Catalog PDF file.
- Status attribute- only products that have Status set to “Default” will be shown in Add Default Item popup when in Quotes/Contracts EditView.
- Contact person
- Category – category should be assigned if one wants the product to be included into Product Catalog Pdf file.
- Description- short description of the product that will be included into Quotes or Contracts.
- Changes from previous revision (version) description – the attribute for description of the changes between Product versions.

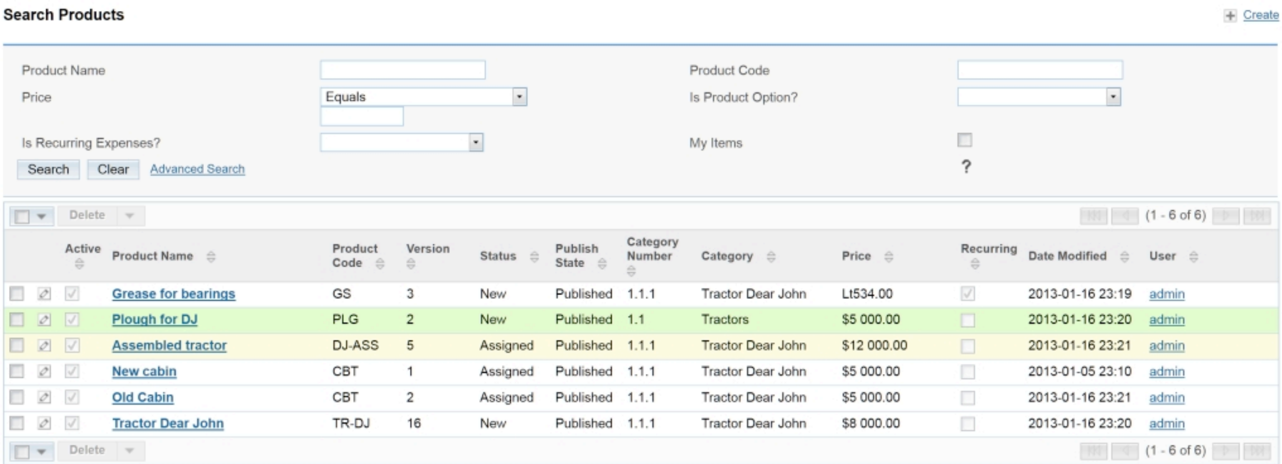


Figure 5: List view of products

The Products can be assembled into packages. Just add existing products to another product in the product Edit view. Use the "update" button to refresh/reload products in the package if Product version has changed. The “update” selection of button is shown in Packaged Product panel only if new version is available.

The Packaged products are highlighted yellow in the List view, as shown in Figure 5.

You can easily associate existing documents with the product to store additional information about it.

You can add one image to each product. Just upload it in Edit view and view it in Detail view and the created pdf files (product catalog and quotes/contracts). The product image will be automatically resized to 700 pixels width after upload (prevents breakdown of Detail view layout). For this feature to work, GD2 PHP extension should be installed (one can check if it is installed by running system diagnostics or phpinfo()).

5.1.1 Product Options

The user can create product that later can be associated with another “master” product. For this, tick the checkbox “Is Product Option?” in EditView (see Figure 5a).

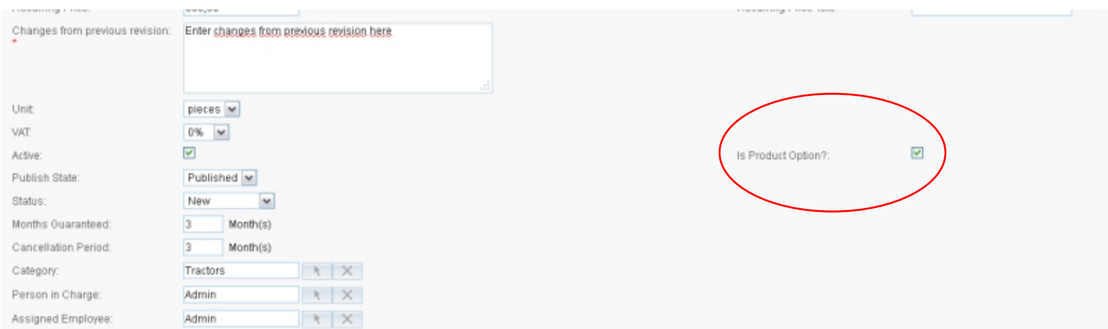


Figure 5a: Product Option checkbox.

Products that are marked as Product Option, will be highlighted in green color in ListView (see Figure 5). These products are editable and versioned the same way as “normal” products. The only difference is that these products cannot be included into Quotes or Contracts as stand-alone items. They should be associated with “normal” product at first. For this, use Product Options panel when in Product EditView (see Figure 5b).

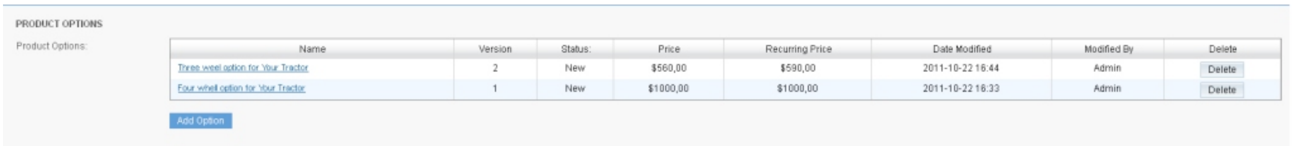


Figure 5b: Product Options panel.

Items in Product Options table can be re-arranged by drag&drop.

Once one has Options associated, it is possible to include these into Quotes and Contracts by pushing “Add Option” button in Product List table (see Figure 7a).

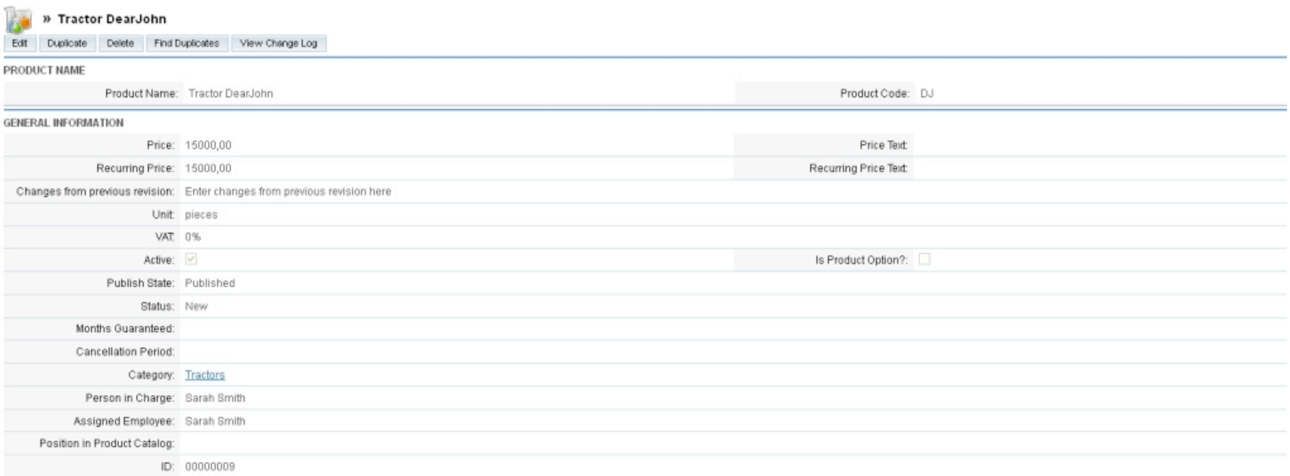


Figure 6: Detail view of a product

5.2 Product Categories

Products are organized in categories which can be structured in hierarchies. Every product is associated with exactly one category. The hierarchy of the categories itself is called the product catalog which is complex enough to be a separate module (see chapter 6). You cannot manage categories directly but within a product catalog.

5.3 Quotes

Quotes have the following main attributes that can be entered or modified in Edit view of the Quote:

- Quote name- can be entered when in Edit view.
- Quote number- OQC generates number automatically by default. The number is constructed from Abbreviation, Date in yymmdd format and unique internal record number. The Quote number can be entered also manually, if corresponding field is not left empty (Quote Number field is hidden by default, so You have to make it visible in Studio for manual Quote Number configuration).
- Abbreviation – Used to generate unique Quote number.
- Version- new version of Quote is created every time when modification is done to the attributes of the Quote. Only latest version of the Quote is displayed in the List view, earlier versions can be accessed from History panel in Quote Detail view.
- Status;

- Account/Contact information fields- customer and/or internal.
- Product list- contains the quote entries with product names, descriptions, pricing, discounts and VAT value.

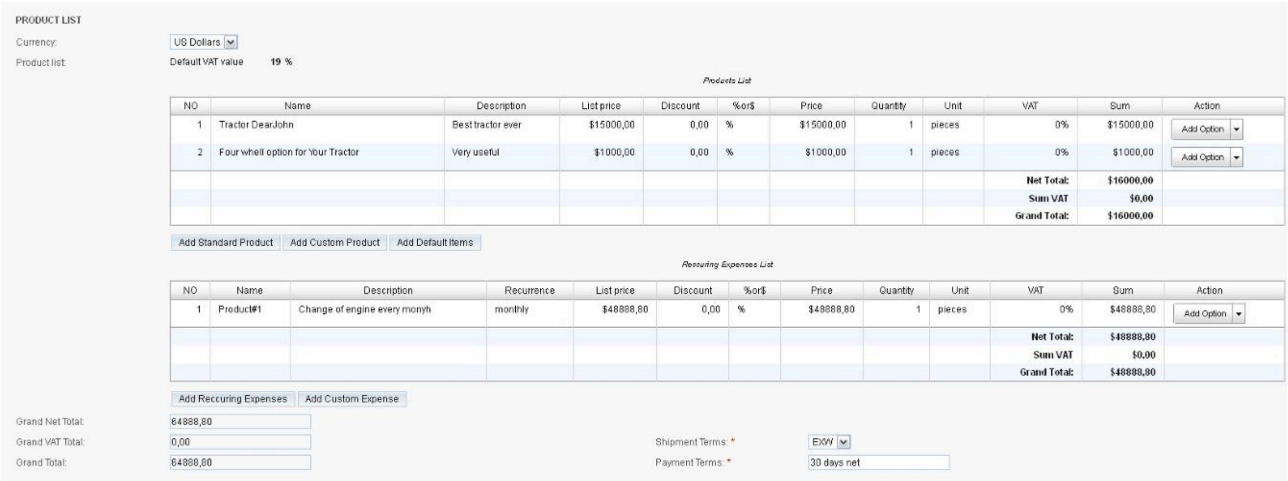


Figure 7a. EditView of Product list table.

The columns that are available are shown in Figure 7a. Button below the table allows adding standard Product that is already in the Products Module, or custom product.

“Add Default Items” button allows adding products that have their Status attribute set to “Default”. It is intended for use for items like shipment cost, installation cost and similar.

An Action button can have three values:

The Update value shows up only when Product version included in the quote is not the latest one and can be updated.

Add Option value allows adding product options that are associated with each product.

Delete removes product from the list.

The items in the table can be re-ordered by entering position number in the NO column. Enter desired row position number for the row that You want to move.

Figure 7a1. Entering row number.

Grand Net Total etc. field below the tables contains sums of both Product list and Recurring Expenses List tables.

The currency change can be done by changing value of Currency dropdown. The values in Quote or Contracts will be recalculated automatically according the exchange values entered in Admin section (Admin->System->Currencies).

- Lead time;
- Warranty terms;
- Shipment terms;

- Payment terms;
- Description- short description of the quote content.

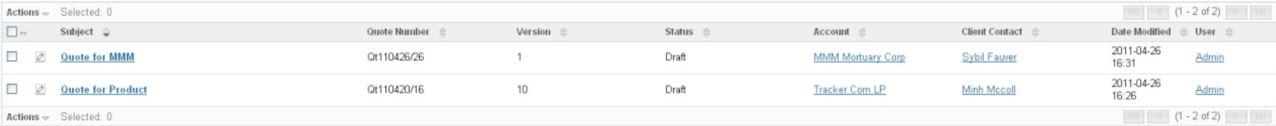


Figure 7b: List view of the Quote

You can easily associate existing documents with the product to store additional information about it in Attachment panel. For this purpose push “Add Attachment” button.

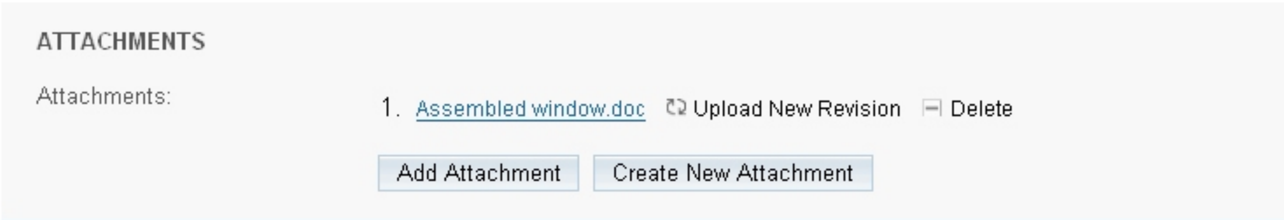


Figure 7c: EditView of Attachments panel.

It is also possible to create new documents from the Quote EditView by pushing “Create New Attachment” button.

The documents that are attached to the Quote will be automatically updated when user uploads new Revision from Documents module. The same task also can be done directly from Quote EditView by pushing “Upload New Revision” link. Popup window will open where file information can be entered.

The order of the entries in Attachment panel is possible to re-arrange by dragging and dropping them in new position.

Attachment panel is available also for Products, Contracts, Additions and ExternalContracts.

Pdf file can be generated from each version of the quote by pushing “Create Pdf” button in Quote Detail view. Pdf document name is constructed from prefix “QuotePdf” and Quote number. The created revision of document is the same as Quote version, for example the pdf file created from third version of quote will have document revision set to 3.

5.4 Contracts

Contracts have some additional attributes in comparison to quotes:

- Information about the schedule
    - start date
    - end date
    - deadline for cancellation
- The signed contract as a file.

The Contract Number is constructed in the similar way as for Quotes, however, separate Abbreviation list is used.

Versioning logic of the contracts depends on whether it was created from Quote (“linked” Contract) or as stand-alone record.

Stand-alone Contracts are versioned identical to the Quotes- new version is created every time something is modified.

Linked Contract version number is kept synchronized to the corresponding Quote version. For example, Contract created form third version of the Quote will have also third version number. When linked Contract is modified, the version number is not increased but kept the same.

Pdf file can be generated from each version of the Contract by pushing “Create Pdf” button in Contract Detail view. Pdf document name is constructed from prefix “ContractPdf” and Contract number. The created revision of document is the same as Contract version, for example the pdf file created from third version of Contract will have document revision set to 3.

When linked Contract is modified, the relationship between document and Contract is deleted since the pdf file is no longer corresponds to the content of the Contract.

Pdf file creation works only for the latest version of the Contract that is shown in List view. For previous versions the pdf file could not be created.

5.5 Contract additions

You can define additions to existing contracts. These extend or update the existing contract. You can only access additions from the contract referenced by them.

5.6 Textblocks

Textblocks are templates describing the matter of the contract. These templates should describe your contracts in general. You can insert and edit the textblocks when editing a contract. If you do any changes in the textblocks while editing the contract, they will affect only this particular contract.

The TextBlocks before included into pdf files, are processed with Smarty. All Quote/Contract, Contacts, Accounts related variables can be included into Quotes. For variables list please see pdf template creation manual.

The variables should be bracketed between {{ and }} for correct interpretation by Smarty. For example, {{$contract.userContact.first_name}} {{$contract.userContact.last_name}} will print current user first and last names when pdf file will be created.

Once inserted into Quotes or Contracts, the ordering of Textblocks can be re-arranged by simply dragging and dropping Textblock to the required position.

5.7 TeamTask

TeamTask module allows to assign task to several SugarCRM users. The TeamTask can be configured to be completed when all Users completes it or when first User completes it.

TeamTask can be created directly from Products/Quotes/Contracts/ProductCatalog DetailView- in this case relationship is created automatically- or as stand-alone TeamTask that is not associated with any record.

The Users that are associated with relevant Products/Quotes/Contracts are included by default. In addition, it is possible from Administration panel to configure the Users that will be added to each created TeamTask by default. Default users can be different for each OQC module.

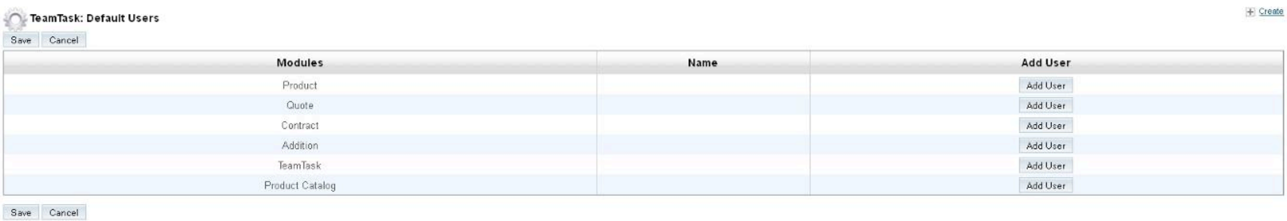


Figure 7d: Default TeamTask users configuration window.

Each User is able to edit only his own row in Participants panel. The row that is editable is highlighted with pink color. The User also can attach files to the row with comments or other information.

The EditView of TeamTask is shown below:

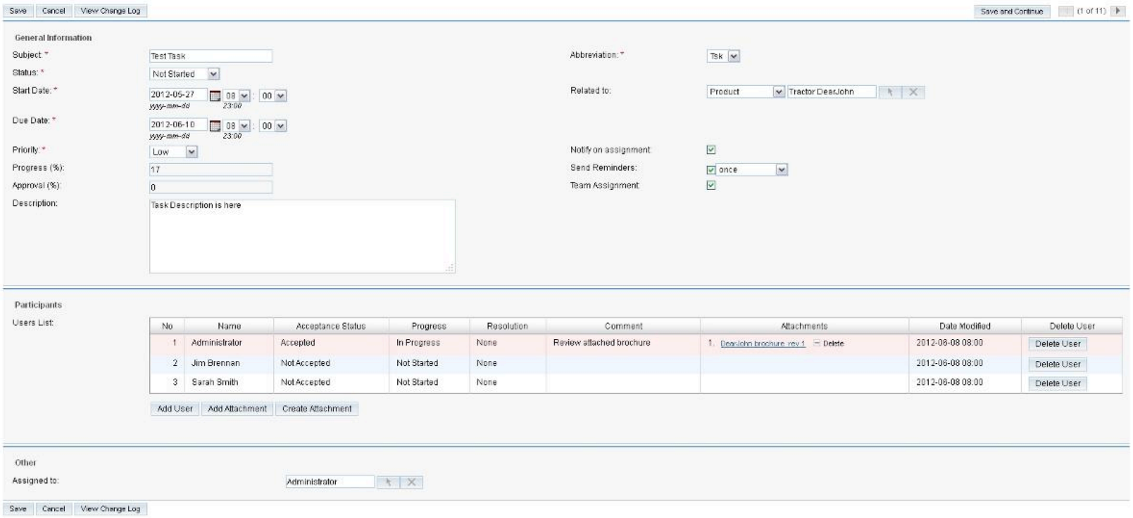


Figure 7e: EditView of TeamTask.

- TeamTask subject- can be entered when in EditView.
- Task Id- OQC generates number automatically by default. The number is constructed from Abbreviation, Date in yymmdd format and unique internal record number. The Task Id can be entered also manually, if corresponding field is not left empty (Task Id field is hidden by default, so You have to make it visible in Studio for manual Task Id configuration).
- Abbreviation – Used to generate unique Task Id.
- Status – indicates the status of the TeamTask. When “Completed” field is selected, the TeamTask will complete independent on values in Participants table. Once completed the TeamTask will be not possible to edit. The warning message is show in Participants panel.
- Start Date and Due Date – the dates when TeamTask starts and is intended to be finished.
- Notify on assignment – if checked, emails to each user will be send after entering TeamTask, and when the TeamTask is completed.
- Send Reminders – if checked, emails to users that did not completed the task will be send when deadline is reached.
- Team Assignment – if checked, TeamTask will be completed when all users reviews and completes it (sets Resolution column to other value than None). If not checked, then the first user that completes it, completes the TeamTask as well.

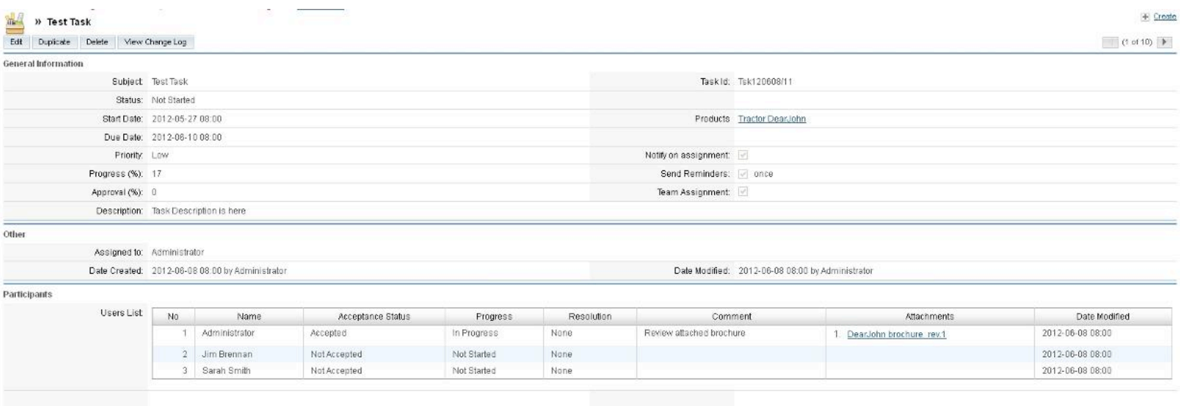


Figure 7f: DetailView of TeamTask.

5.8 Database maintenance

From version 2.2 RC1 OQC has administator tool for clean up of old versions of Products, Quotes, Contracts and External Contracts.

Figure 7g: CleanUp tool.

“Versioned modules” filed allows to select what modules user want to clean up. At least one module should be selected. Use Ctrl+Click to select several modules.

“Remove all records till” – all versions created or modified earlier that set date will be marked deleted. This setting can be overridden by “Remove all except latest?” checkbox. If this checkbox is activated, all versions except latest will be marked deleted.

“Remove from database permanently?” checkbox is a tricky one. If not checked, the records will be “soft” deleted. They will be not removed from database, but marked as deleted. Database size will not change. “Soft” deleted versions will be no longer accessible to the users. The advantage of this method is that if something was deleted accidentally, it can be relatively easy restored by modifying “deleted” field with database manipulation tools.

When “Remove from database permanently?” checkbox is checked, the records will be deleted from database and database size will decrease. Backup file of records deleted will be created in YourSugarInstance/cache/backups folder. Name of file is “oqc-backup-*date_time*.php. Using backup file it will be possible to restore deleted records.

In some situations permanent deletion of records might break relationships between items, especially in Products module. It is recommended first to try this function on development system.

6 Product Catalog

6.1 Introduction

2.x versions brings some changes to the product catalog user interface and the overall logic. It is possible to create several catalogs that might include different groups of the products. This allows better structuring and easier management of Product Catalog. The Products are considered global and can be included in any catalog by associating it with appropriate category of the catalog.

In the product catalog module you can create categories and structure them in hierarchies.

Product Catalog have the following main attributes that can be entered or modified in Edit view of the Product Catalog (see Figure 8):

- Product Catalog name- can be entered when in Edit view.
- Currency- allows selecting the currency of the Product Catalog. The prices of the products are calculated using currency conversion rates entered from Administration page. Note, the currency setting has effect only in created PDF file.
- Pdf Template – select which template to use for creation of PDF file. There are three pre-set templates, users can also create own templates.
- Global Discount/Markup – entered in percent. Positive numbers means discount while negative- markup in comparison to “regular” prices. Discount or Markup applied to all products included into Product Catalog. Note, this setting has effect only in created PDF file.

- Pdf Document Name- Enter PDF document name that will shown up in the Documents subpanel in Product Catalog Detail View.
- Valid From/Valid to – enter Product Catalog validity period that will be printed in PDF file.
- Appearance- instead of using default front and end pages of created PDF file user can upload own PDF files that will be used as front and end pages.
- Product Categories- create Product Catalog categories and hierarchy structure.

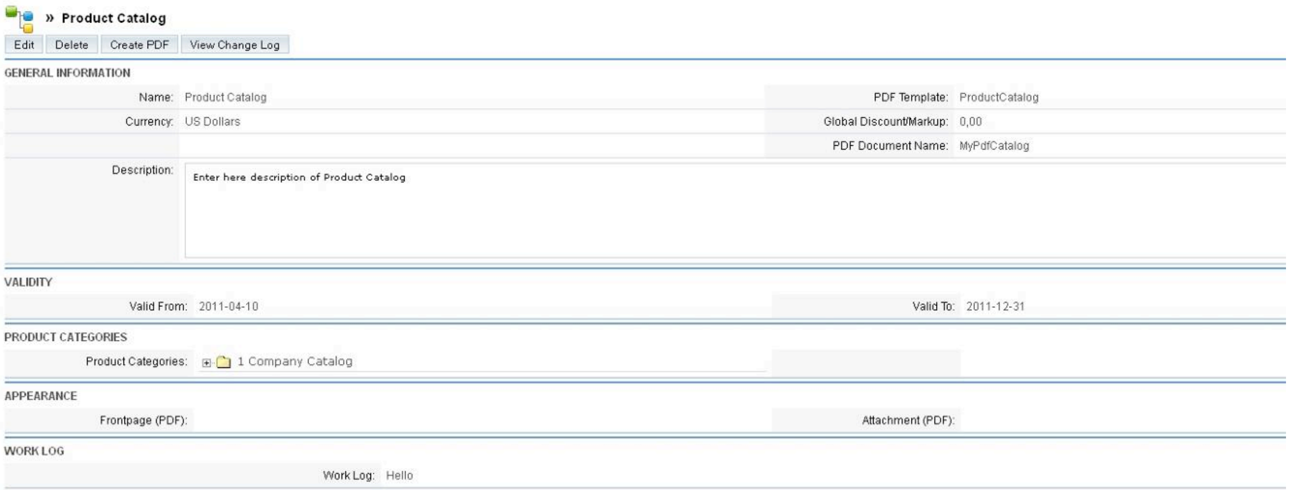


Figure 8: Detail view of a product catalog

Including/editing Products into Categories

The Product Catalog categories should be populated with the Products. After catalog creation one can assign newly created or existing product to one of categories. The assigned products can be seen in ProductCatalog DetailView. Only Products that have status “active” and “published” can be seen in tree (see Figure 8a, products will have alphabetic numbering like 1.1.a, while Categories have numerical indexes). Product Options will appear in the tree only when assigned to “master” product. The icon of option is different to “master” products.

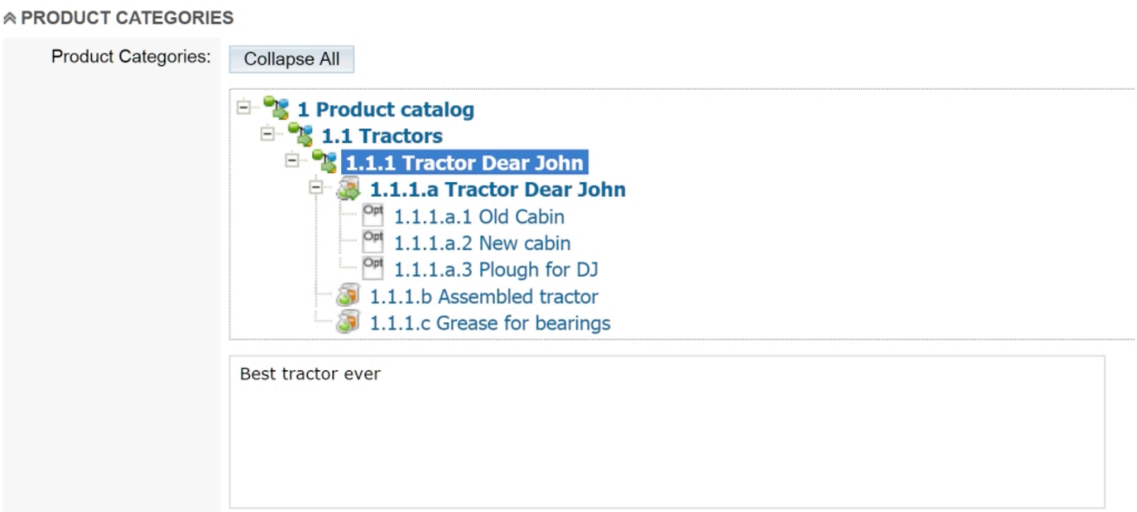


Figure 8a: Detail view of a product catalog with products assigned to the category.

The products are visible in EditView as well. The category assignment of products can be edited in EditView using drag-and-drop or from Products Module.

Only Products that have set status to “active” are displayed in Select Product Pop-ups and can be included into Quotes or Contracts. Active flag can be set or disabled from Product Editview.

Editing category sequence

Since 2.2RC1 release editing capabilities thanks to dynatree plugin are significantly improved.

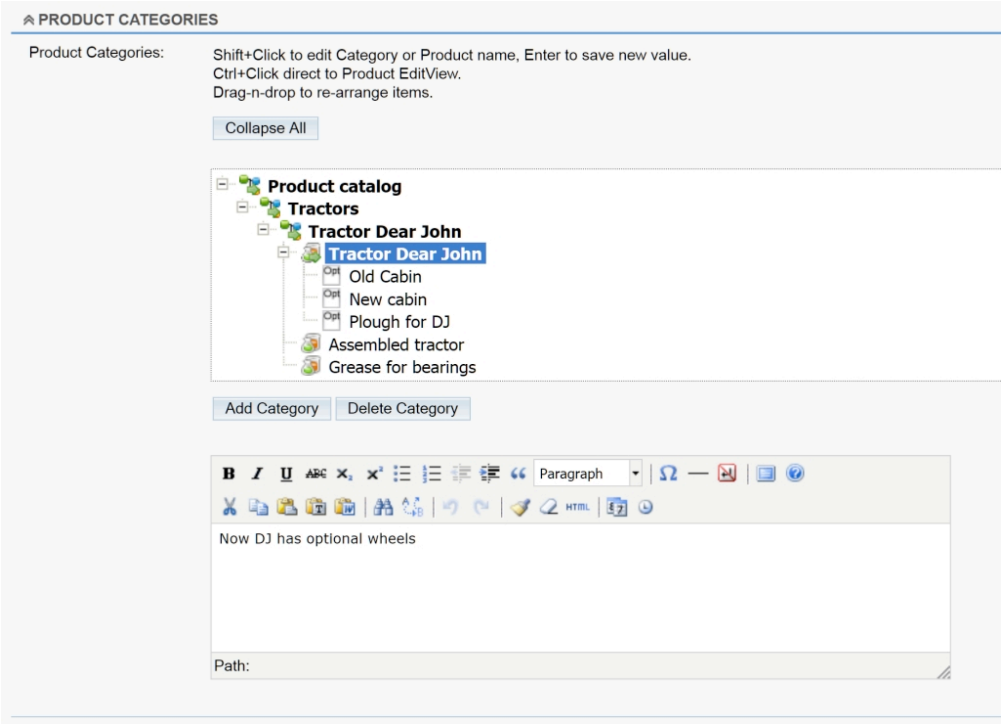


Figure 8b: Edit view of a product catalog with products assigned to the category.

Category sequence and hierarchy can be changed by simple drag-n-drop.

Product sequence can be changed or products can be assigned to new category. Product name can be edited the same way as Category, by Shift+Click. It is possible to edit also Product description in html text editor below category tree. During the save of Product Catalog, Product Name, Catalog Position, Assigned Category and Description will be updated (however, new version of Product will not be created).

Creation of Product Catalog Pdf file

By pushing “Create Pdf” button the information stored in category tree and product descriptions can be organized into Product Catalog. The templates used for Pdf file creation are described in more detail in Chapter 8.2.

The Products are included into pdf file if:

- active attribute is set to true;
- publish state attribute is set to “published”.

Each Product might have one or several file attachments (documents). These attachments can be included into pdf file as well, if Document Purpose field is set to “For Product Catalog”. **NOTE:** Document Purpose field in Documents module is hidden after install, so please configure it to be visible trough Studio.

Each time the button “Create Pdf” is pushed new revision of catalog document is created. The Product Catalog document name is constructed from string $app_list_strings['oqc']['pdf']['common']['filenamePrefixCatalog'] (can be found in SugarModules\language\application\en_us.lang.php of installation zip file) and current year.

The ProductCatalog revision file name in addition has information about creation date in “Ymd” format and revision number.

6.2 Step By Step: Create a Product Catalog Categories

1. Open the ProductCatalog module in SugarCRM and push “Create Product Catalog”.
2. You will see the window like in Figure 9. There will be two dummy categories created by default. Shift+Click to edit name of category.

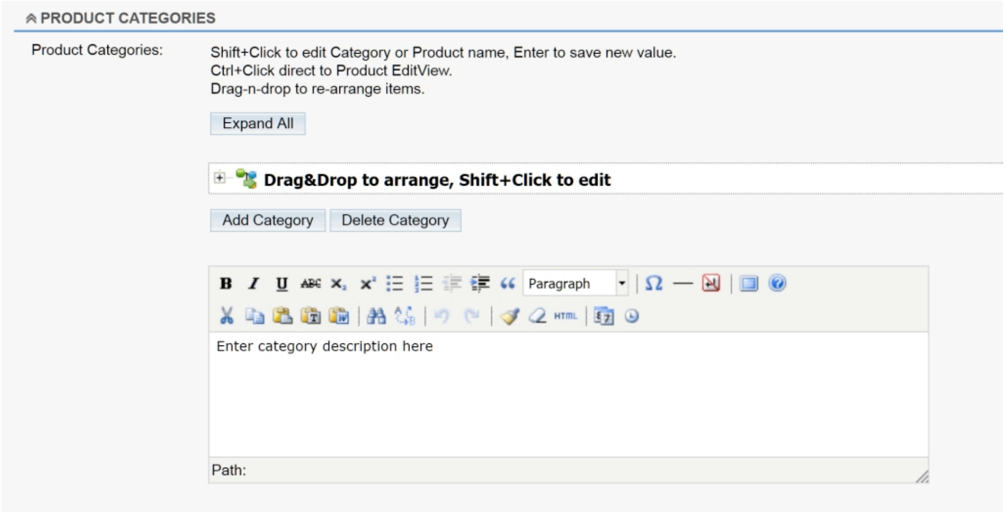


Figure 9: New Product catalog window

3. Click on the 'Add Category' button. This will add node to the root of the tree. Shift+Click the node to update its name. If you select a node by clicking on it you will see its description in the editor at the bottom (see figure 10).

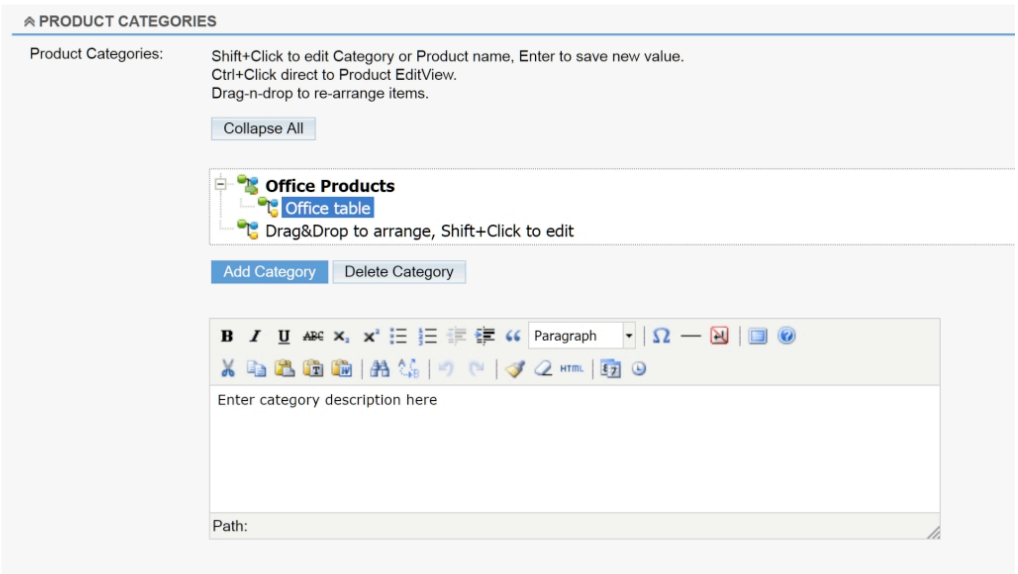


Figure 10: Editing of product categories

4. Create other categories as well. New categories will be inserted into root category. You can change the hierarchy with drag and drop.
   You should avoid creating hierarchies deeper than five levels because this will not look very good in the pdf file. It is possible but will result in a layout that is not very clear.

7 External Contracts

7.1 Create positions

New external contracts do not have a table enumerating the cost positions (see figure 11).

Figure 11: Instead of a cost positions table a new external contract only has a button for adding position rows.

If you click the add position button new rows in the table are created (see figure 12). The positions table has no effect on the finalcosts fields at the bottom of the page. It is only there to explain the costs in more detail.

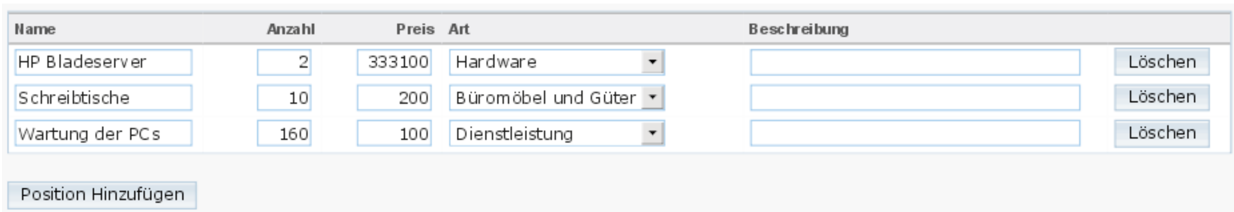


Figure 12: Add some cost positions.

7.2 Create Costs

To create new costs select some of the contract matter values (see figure 13). Hold the CTRL key to select several contract matter values.

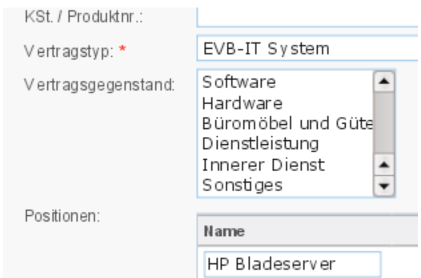


Figure 13: Select several contract matter values to create cost tables

The creation of cost tables is influenced by the begin and end dates of an external contract as well. All necessary cost tables are created automatically. If you change the contract matter values or the lifetime of the external contract, the tables are updated correctly (see figure 14).

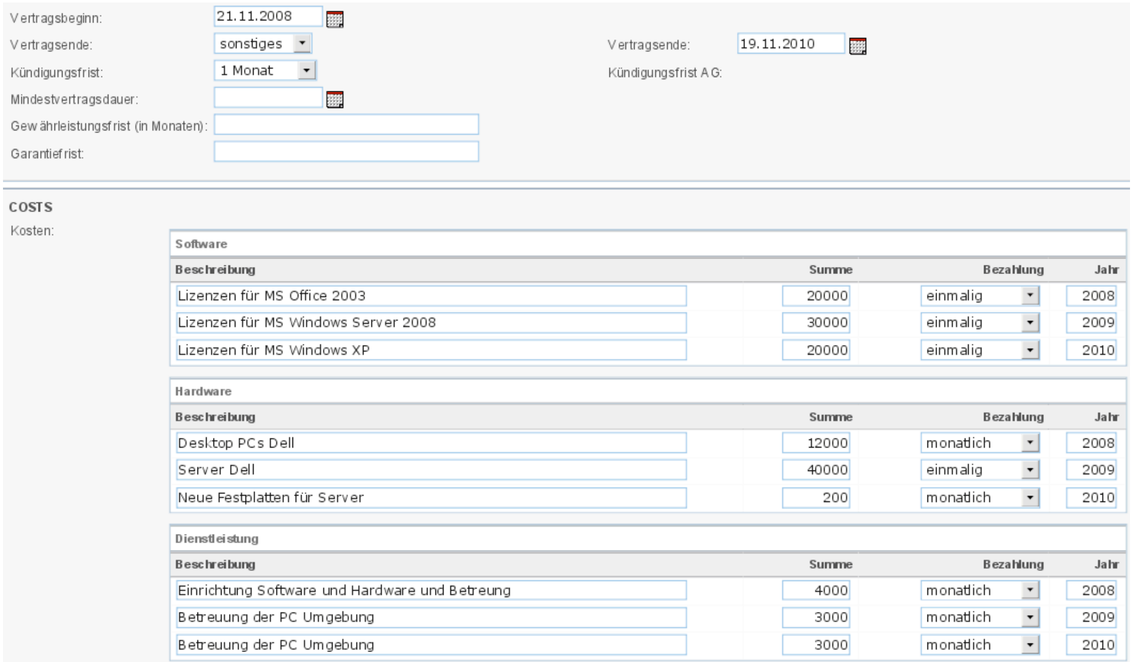


Figure 14: Edit the description, prices and payment fields in the cost tables.

The system caches your input as you type. You can update the lifetimes or modify the contract matter values without losing your already entered data. This means you can enlarge the lifetime, fill-out the fields of the cost tables, shrink the lifetime and enlarge it again, you will see your previously entered data again. Just try it!

7.3 Contract Numbers

In the contract number section you can enter the numbers of existing contracts or those that have not been registered in the system yet. This is useful if you want to associate the external contract with other contracts. You can use the quicksearch feature for entering the contract numbers (see figure 15).

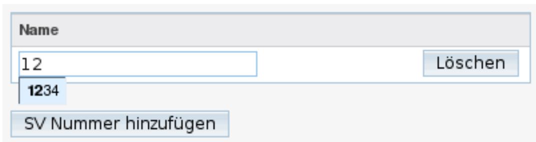


Figure 15: Usage of the Quicksearch feature during the selection of contract numbers

However, it is also possible to enter numbers of contracts not registered in your sugarcrm installation (see figure 16).

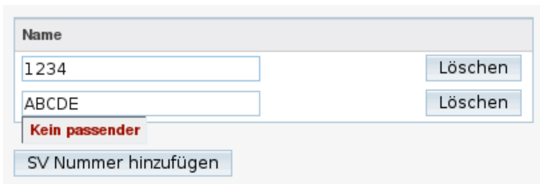


Figure 16: Enter the contract numbers of existing and not existing contracts

7.4 PDF creation

Like in other modules (product catalog, quotes, contracts) you can create a pdf file from every external contract summarizing it into a printer-friendly version (see figure 17).

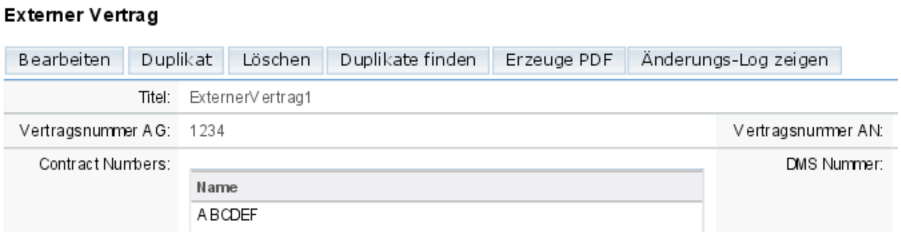


Figure 17: The create pdf button in the detail view of an external contract

7.5 Export to CSV

You can export lists of external contracts to work with them in Excel. To do so you have to sort the contracts in SugarCRM by a criteria like accounts. In the CSV file you will see sums of the finalcosts for the contracts with a row for sub-totals after each group. If you sort by accounts the finalcosts of all contracts will be added to a subtotal for each account. The sub-totals will be listed after the contracts.

The following steps are required to for the CSV export:

- Decide on which field you want to group the contracts and *sort the list by this field*. Please note that you can use the results of a search as a basis for your selection as well.
- Select the contracts from the list that should appear in the CSV export. Please note the shortcuts for selecting all contracts from the list.
- Click on Export. A CSV file will be generated that you can download and open with Excel or OpenOffice.org for example.

7.5.1 Setting up the CSV export

To be able to open the generated CSV files in Excel please go to the admin area in SugarCRM and do the following:

- Set the default export delimiter character to `;`
- Set the default export character set to `ISO-8859-15`.

Perhaps you will need other settings for different office software or different client operating systems. Please note, that you can use OpenOffice.org to import the CSV files as well.

7.6 E-Mail notifications

With the e-mail notification feature, all employees that are in a certain role are informed about external contracts that will end in the next weeks. For each contract only one notification will be send.

A notification e-mail could look like the following:

```
Hello Ingo Jaeckel,

the following external contracts will run out soon:

On 30.06.2009 ends "Contract ABC", see link.
On 30.06.2008 ends "Contract 123", see link.

Best regards, SugarCRM :-)
```

7.6.1 Customization

You can easily customize the e-mail notification feature to your needs. Have a look on the `ChangeNotifier` class in the `include/oqc/common/ChangeNotifier.php` file. For sending e-mails an instance of the `ChangeNotifier` class is created and the `notifyUsersOfChange` method is called on it.

7.6.2 Configuration

Warning deadline

For each external contract you can set the number of months before the contract enddate you should be notified of the ending of the contract. This number defaults to two months. But you can set in the edit view of the contract.

Role

The name of the notification role can be configured as well. It defaults to `ExternalContractsNotification` and you can change it by setting the variable `notificationRole` in the configuration file `include/oqc/conf/documents.properties`.

Sender address

Change the address of the sender by updating the variable `changeNotifierEmailAddress` in the configuration file.

Text template

The text templates used in the email can be changed in the language files of the SugarCRM installation[1] or in the openqc installation zip file[2]. The templates are stored in the variable `$app_list_strings["oqc"]["Email"]`. The meaning of the variables keys are discussed now:

- `subject`: The subject of the notification email.
- `hello`: The “hello” to the employee receiving the email.
- `greetings`: The greetings at the end of the email.
- `introduction`: The introduction sentence before the list of contracts that will end in the near future.
- `notificationLine`: The text that describes a single ending external contract. It contains the contract name, the enddate and a link to the contract in your SugarCRM installation.

Cronjob

To automatically send the notification emails the file `oqc/notifyUsers.php` has to be executed periodically. This can be done as a Cronjob directly in SugarCRM or as a cronjob of a nonprivileged user on the SugarCRM webserver. We recommend the daily execution of the script.

8 PDF Generation

The generation of PDF documents is fundamentally composed of the following steps:

- Preprocess data originating from the webinterface using *PHP code*.
- Convert formatted HTML texts to latex code by running *oqc_HtmlToLatexConverter*.
- Convert attached files to PDF using *DocumentConverter* and *OpenOffice*. This step is optional, if one attach the files that are already in PDF format, then this step will be skipped.
- Merge preprocessed and converted data into the static template by the use of S*marty*.
- Generate PDF document out of filled in template using *pdflatex*.
- Attach attachments to PDF document using *pdftk*.

If the generated PDF document is to be redesigned, essentially the static template, which consists of LaTeX and smarty code, has to be changed. New data has to be preprocessed in the PHP code and passed to smarty before processing the template.

---

[1] See `custom/application/Ext/Language`.
[2] See `SugarModules/language/application`.

8.1 Quotes and Contracts

Data-preprocessing PHP code is in `modules/oqc_Contract/oqc_CreatePdf.php`, which depends on `include/oqc/Pdf/Common.php`.

The static template, which consists of LaTeX and smarty code, is structured into the following parts under `include/oqc/Pdf/Contract/`:

Template frame (`Contract.tpl`): Definition of document properties, pdf informations and listing of parts.

Title page (`Titlepage.tpl`): Logos, contract data, addresses.

Contract arrangements (`TextBlocksSkeleton.tpl`): Text blocks.

Service tables (`Services.tpl`): Tables containing services.

8.2 Product Catalog

Data-preprocessing PHP code is in `modules/oqc_ProductCatalog/ProductCatalogPdfCreator.php`, which depends on `include/oqc/Pdf/Common.php`.

The static template, which consists of LaTeX and smarty code, is structured into the following parts under `include/oqc/Pdf/ProductCatalog/`:

Template frame (`ProductCatalog.tpl`): Definition of document properties, pdf informations, static text, formatting definitions and listing of parts.

Introduction, Contact Information and Validity parts (`Intro.tpl, Contact.tpl, Validity.tpl`): The files containing Introduction, Contact Information and Validity descriptions of the Product Catalog.

Categories, service part (`Categories.tpl`): Description of categories and detailed descriptions of products.

Products, pricelist part (`Products.tpl` and `ProductTables.tpl`): Listing of categories and their products including price and unit.

References

[1] The OpenQuotesAndContracts project, `http://www.sugarforge.org/projects/openqc/`

[2] LDS, Landesbetrieb fur Datenverarbeitung und IT-Serviceaufgaben des Landes Brandenburg, `http://www.lds-bb.de`

[3] Hasso-Plattner-Institut, `http://www.hpi.uni-potsdam.de`

[4] DocumentConverter, `http://artofsolving.com/opensource/pyodconverter`

[5] Deutsche Sprachdateien fur SugarCRM 5.2, `http://www.sugarforge.org/projects/germanlangpack/`