CGD-QP-001

内部

公司文档管理办法

（试行）

苏州珂晶达电子有限公司

标　　题	公司文档管理办法:（试行）		
编　　号	CGD-QP-001	密　　级	内部
类　　别	质量流程	状　　态	试行
关联文档			
责 任 人	赵军	客户单位	
联系方式	zhaoj@cn.cogenda.com	联系方式	
签　　署		会　　签	
日　　期		日　　期	

摘要

第一部分为公司文档的管理和排版的要求。第二部分为技术内容和对要求的解释，包括排版软件和排版模板的使用介绍、表格、图片以及代码的排版方式。

修订记录

版本	日期	负责人	备注
0.1	2013.12.25	赵军	draft
0.2	2014.01.04	沈忱	translate .tex to .cls

目录

代码目录

第一章　要求

为规范公司文档的管理，特制定本办法。目前为征求意见阶段，欢迎意见和建议。

1.1　适用范围

本管理办法适用于所有权归公司的、密级为“内部”及“公开”的正式文档。包括但不限于：

- 公司对外的正式报告，包括技术方案、技术总结以及操作手册等。
- 公司对外的书面交流文档。
- 公司对内的质量流程、开发文档以及有存档价值的过程文档等。

不属于此办法的管理范围：

- 密级为“秘密”及以上的文档。
- 无存档价值的交流文档。

1.2　排版要求

公司要求以后的正式文档尽可能用 LaTeX 排版，并使用公司统一的模板。具体要求和实现方法见本文档后面的部分。

1.3　文件存档

1.3.1　应有文件

每一份文档对应的不是一个文件，而是一个**目录**。这个目录中应有两个文件：

1. 最终的 `pdf` 文件。
2. 一个 `zip` 或 `gzip` 包，其中包含生成这个 `pdf` 文件所需要的所有 `tex` 文件、图片文件及其他文件如插入代码、原始数据等。

注意事项：

1. 如果这个文档是用 lyx 软件编辑的，则应在 `zip/gzip` 压缩包中包括 `lyx` 文件。
2. 如果文档中的插图是用 LibreOffice 等软件制作的，应在 `zip/gzip` 压缩包中包括 `odt/odp` 等文件。

3. 如果生成 pdf 文档需要特殊的操作步骤，或有其他需要注意的事项，应当在 README 文件中加以说明。

1.3.2 文件位置

存在服务器上，全公司可读，总路径为：/home/public/document。以本文档为例，其完整的目录为：/home/public/document/procedure/CGD-QP-001/V01/。其三层路径名：

- procedure 表示文档类型为质量流程，
- CGD-QP-001 为本文档的编号，
- V01 表示版本。

这个路径里面有两个文件，一个 pdf 文件，一个原始文件的 gzip 包。

1.3.3 文档编号规则

文档编号为 CGD-XX-xxx，其中 CGD 表示 Cogenda，XX 为字母，表示文档类型。xxx 为数字，为本文档的顺序号。编号 XX 对应的类型及其存放路径为：

XX	文档类型	存放路径
TP	技术方案	proposal
TR	技术总结	report
MN	操作手册	manual
QP	质量流程	procedure
MS	其他文档	miscellaneous

1.3.4 目录名和文件名

每个文档对应的目录名是文档的完整编号，不可以用其他内容。文档内的文件名可以自由命名，但只能由英文字母、数字、下划线（_）或英文句点（.）组成，不可以出现其他字符。

1.3.5 入档

文件入档，如果还没有文档编号，请先跟公司文档管理员申请编号，填入文档后在 helium 下测试是否顺利生成 pdf 文档，然后将文件拷给公司文档管理人员。

第二章　排版软件及模板的使用方法

公司在试用了多个排版软件之后，决定选用 LaTeX，并使用统一的模板。LaTeX 的工作方式为：用户编辑的文档为纯文本文件，配上用户需要的图片文件（或者原始数据文件），用 LaTeX 软件处理成为 ps 或 pdf 文件。如图 2.1 所示。用户对版式的要求全部写在文本文件中，也就是用户需要了解一些 LaTeX 的排版语法。

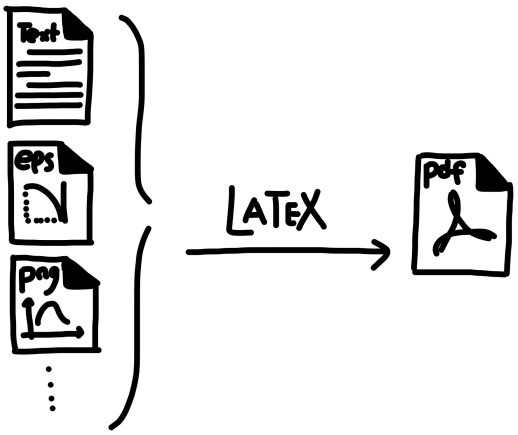

图 2.1: LaTeX 工作方式

公司选用 LaTeX 并采用统一的模板，好处为：

- 统一公司文档的风格。
- 方便不同的人在不同平台下编辑和修改同一个文档。
- 对于大型文档，LaTeX 崩溃的概率小于其他常见排版软件；文本格式的文件损坏的概率也小。
- LaTeX 排出的版面更漂亮。
- LaTeX 免费。

坏处是用户编辑 TeX 文本文档时不够直观。对此公司的建议顺序是：

1. 使用纯文本编辑器编辑 TeX 文档，找一个 LaTeX 手册放在手边。（LaTeX 常用的语法也就那么几页纸的事 [?]。）
2. 用 lyx 编辑。
3. 其他可输出 TeX 文档的软件。要求生成的 TeX 文档适于人工阅读。

2.1 软件的安装和使用

目前我们公司使用的 LaTeX 发行包为 TexLive，你可以直接使用公司 Helium 服务器上的，也可以在自己的电脑上安装一份来用。以下分别介绍。

2.1.1 在服务器上使用

只需要设置使用 TexLive 软件所需的环境。

```
# Set the path environment of TexLive
  source /usr/local/texlive/setenv.sh
# Or to make it permanent, you can simply do:
#    cat /usr/local/texlive/setenv.sh >> $HOME/.bash_profile
# test it
  which xelatex
```

代码 2.1: 在服务器上设置使用 TexLive 的环境

2.1.2 在自己电脑上安装

目前我们使用的 TexLive 是个完整的庞大的包，其 iso 镜像文件 2.4G，安装后需要 3.5G 的硬盘空间。安装前请在你自己的硬盘上准备充分的空间。镜像文件下载路径为：/home-/public/software/tex/texlive2013。对于 Linux 系统，可能需要先几个字体文件。字体文件在 /home/public/sofware/tex/texlive2013/win_fonts.tar.gz。

假设你准备把字体安装在 /usr/share/fonts/TTF/。安装方法为：

```
# copy the font files
  tar -xzvf win_fonts.tar.gz && sudo cp -r win_fonts /usr/share/fonts/TTF/
# activate the fonts
  sudo fc-chache -fv
```

代码 2.2: 安装字体

假设你准备把 TexLive 安装在 /usr/local/tex 。安装方法为：

```
# mount the iso file
  sudo mount -o loop texlive2013-20130530.iso /mnt/dvd  && cd /mnt/dvd
# install it
  export TEXLIVE_INSTALL_PREFIX=/usr/local/tex
  perl install-tl
# then follow the instructions to set the path environment
  export PATH=/usr/local/texlive/2013/bin/x86_64-linux:$PATH
```

代码 2.3: 安装 TexLive

对于 Windows 用户，以上镜像文件也支持 Windows。请阅读镜像文件里的说明文件。

2.1.3 Lyx

Lyx 是一个直观的编辑 TEX 文件的编辑器。不需要 lyx 的人可以跳过这一节。服务器上已经安装了 lyx，你可以直接使用，也可以在自己电脑上安装。

在安装模板之前，需要先启动 lyx 一次以生成 `$HOME/.lyx`。安装模板之后，启动 lyx 并点击 Tools ▶ Reconfigure，然后重新启动 lyx。

2.2 模板的安装和使用

公司的技术报告模板是在用户自己目录下的，所以即使用户是在服务器上运行 TexLive，也需要安装模板。模板路径：`/home/public/document/template/cgdrep.tar.gz`。安装方法：

```
# Unpack the template
  tar xf /home/public/document/template/cgdrep.tar.gz
  cd cgdrep
# Install the template files
  ./install.sh
```

代码 2.4: 安装模板

没有报错的话，可以使用 LATEX 模板的示例了：

```
# test the LaTeX example
  cd example/latex
  make
# There should be several new files and one of them is main.pdf
  evince main.pdf &
```

代码 2.5: 模板示例

第三章　表格、图片和代码的格式

我们暂定了表格、图片和代码的格式。这并不是最终版，欢迎提出意见。

3.1　表格

一般，我们要求表格采用三线表，首行为其 header，用黑体。字号比正文半号，位置居中。如表 3.1：

表 3.1: 三线表

版本	日期	负责人	备注
1.0	2013.12.24	沈忱、纪冬梅	初稿
1.1	2013.12.25	赵军	二稿

实现这样的表及其浮动位置，可以用 Cogenda 自己制作的宏 ctable，其语法为：

```
\begin{ctable}{Label}{Caption}{Alignments}
 第一行 （表格项目）  \\ \hline
 第二行 \\
.....
\end{ctable}
```

代码 3.1: ctable 语法

则表 3.1 的完整语法为：

```
\begin{ctable}{triline}{三线表}{llll}
 版本 & 日期       & 负责人       & 备注 \\ \hline
 1.0  & 2013.12.24 & 沈忱、纪冬梅 & 初稿\\
 1.1  & 2013.12.25 & 赵军         & 二稿  \\
\end{ctable}
```

代码 3.2: ctable 示例

如果你不想局限于 ctable 的功能，也可以用一般的 LaTeX 表格语法（用了 tabu 宏包），表 3.1 的一般 LaTeX 语法为：

```
\begin{table}[htbp]\caption{\label{triline} 三线表}
\centering\small\begin{tabu}{llll}\thickhline\rowfont{\bfseries}
```

```
 版本  & 日期       & 负责人       & 备注 \\ \hline
 1.0  & 2013.12.24 & 沈忱、纪冬梅 & 初稿 \\
 1.1  & 2013.12.25 & 赵军         & 二稿 \\
\thickhline\end{tabu}\end{table}
```

代码 3.3: 三线表的一般 LaTeX 语法示例

3.2 图片

插入图片语法简单，这里没有设置自己的宏，而是用的是普通语法：

```
\begin{figure}[htbp]\centering
\includegraphics[width=0.5\textwidth]{file.jpg}
\caption{\label{Label}Caption}
\end{figure}
```

代码 3.4: 插入图片语法示例

对于图片的大小控制，当用户没有特别需求的时候，建议采用 `textwidth` 的倍数方式。

3.3 代码

先给出一段文档中插入 Pascal 代码的样子：

```
for i:=maxint to 0 do
begin
{ do nothing }
end;
Write('Case insensitive ');
Write('Pascal keywords.');
```

代码 3.5: Pascal 代码示例

引用代码的 LaTeX 语法为：

```
\begin{lstlisting}[language=Language,caption={Caption},label=Label]
.......
  （代码内容）
.......
\end{lstlisting }
```

代码 3.6: 引用代码的语法

则如上代码 3.5 的引用方法为：

```
\begin{lstlisting}[language=Pascal,caption={Pascal 示例代码},label=PascalExample]
for i:=maxint to 0 do
begin
{ do nothing }
```

```
end;
Write('Case insensitive ');
Write('Pascal keywords.');
\end{lstlisting}
```

代码 3.7: 引用代码示例

如果代码比较长，在某个原始代码文件里，可以用如下语法，将整个文件的代码引入：

```
\lstinputlisting[language=Language,label=Label,caption={Caption}]{Filename}
```

代码 3.8: 引入代码文件示例

下面给出 C、Python 和 Bash 代码示例，欢迎提出排版意见。

```c
#include <stdio.h>
#define N 10
/* Block
 * comment */

int main()
{
    int i;

    // Line comment.
    puts("Hello world!");

    for (i = 0; i < N; i++)
    {
        puts("LaTeX is also great for programmers!");
    }

    return 0;
}
```

代码 3.9: C 代码示例

```python
class BankAccount(object):
    def __init__(self, initial_balance=0):
        self.balance = initial_balance
    def deposit(self, amount):
        self.balance += amount
    def withdraw(self, amount):
        self.balance -= amount
    def overdrawn(self):
        return self.balance < 0
my_account = BankAccount(15)
my_account.withdraw(5)
print my_account.balance
import unittest
def median(pool):
```

```
    copy = sorted(pool)
    size = len(copy)
    if size % 2 == 1:
        return copy[(size - 1) / 2]
    else:
        return (copy[size/2 - 1] + copy[size/2]) / 2
class TestMedian(unittest.TestCase):
    def testMedian(self):
        self.failUnlessEqual(median([2, 9, 9, 7, 9, 2, 4, 5, 8]), 7)
if __name__ == '__main__':
    unittest.main()
```

代码 3.10: Python 代码示例

```
#!/bin/sh
# renna: rename multiple files according to several rules
# written by felix hudson Jan - 2000

#first check for the various 'modes' that this program has
#if the first ($1) condition matches then we execute that portion of the
#program and then exit

# check for the prefix condition
if [ $1 = p ]; then

#we now get rid of the mode ($1) variable and prefix ($2)
  prefix=$2 ; shift ; shift

# a quick check to see if any files were given
# if none then its better not to do anything than rename some non-existent
# files!!

  if [$1 = ]; then
     echo "no files given"
     exit 0
  fi

# this for loop iterates through all of the files that we gave the program
# it does one rename per file given
  for file in $*
    do
    mv ${file} $prefix$file
  done

#we now exit the program
  exit 0
fi

# check for a suffix rename
```

```
# the rest of this part is virtually identical to the previous section
# please see those notes
if [ $1 = s ]; then
  suffix=$2 ; shift ; shift

   if [$1 = ]; then
    echo ”no files given”
   exit 0
   fi

 for file in $*
  do
   mv ${file} $file$suffix
 done

 exit 0
fi

# check for the replacement rename
if [ $1 = r ]; then

  shift

# i included this bit as to not damage any files if the user does not specify
# anything to be done
# just a safety measure

  if [ $# -lt 3 ] ; then
    echo ”usage: renna r [expression] [replacement] files... ”
    exit 0
  fi

# remove other information
  OLD=$1 ; NEW=$2 ; shift ; shift

# this for loop iterates through all of the files that we give the program
# it does one rename per file given using the program ’sed’
# this is a sinple command line program that parses standard input and
# replaces a set expression with a give string
# here we pass it the file name ( as standard input) and replace the nessesary
# text

  for file in $*
  do
    new=‘echo ${file} | sed s/${OLD}/${NEW}/g‘
    mv ${file} $new
  done
exit 0
```

```
fi

# if we have reached here then nothing proper was passed to the program
# so we tell the user how to use it
echo "usage;"
echo " renna p [prefix] files.."
echo " renna s [suffix] files.."
echo " renna r [expression] [replacement] files.."
exit 0

# done!
```

代码 3.11: bash 代码示例

3.4 几类特别对象

有几类特别对象，要在文档里用不同的字体，我们定义了如下字符：

对象	语法	效果示例
文件名	\filename{}	directory/file.name
命令名	\command{}	which xelatex
参数名	\parameter{}	Energy
用户输入	\userinput{}	*runApp*
图形界面菜单	\guimenu{>{}}	Edit ▶ Copy

3.5 list 列表

对于三个 lists（itemize、enumerate 和 description），我们都改了间距。如果用户将 itemize 用于段落，想保留段落间距的话，用新的环境变量 paraitem。语法为：

```
\begin{paraitem}
\item 第一段。
\item 第二段。
.....
\end{paraitem}
```

代码 3.12: 保留段落间距的环境变量