

CGD-QP-001

内部

公司文档管理办法

（征求意见稿）



苏州珂晶达电子有限公司

2013.12.31

| | | | |
|-------|----------------------|------|-----|
| 标 题 | 公司文档管理办法 | | |
| 编 号 | CGD-QP-001 | 密 级 | 内部 |
| 类 别 | 质量流程 | 状 态 | 未定稿 |
| 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 | 2013.12.31 | 赵军 | still a draft |
| 0.3 | 2014.01.04 | 沈忱 | translate .tex to .cls |

目录

| | |
|-------------------------------------|---|
| 第一章 要求 | 1 |
| 1.1 适用范围 | 1 |
| 1.2 排版要求 | 1 |
| 1.3 文件存档 | 1 |
| 1.3.1 应有文件 | 1 |
| 1.3.2 文件位置 | 2 |
| 1.3.3 文档编号规则 | 2 |
| 1.3.4 目录名和文件名 | 2 |
| 1.3.5 入档和查询方法 | 2 |
| 第二章 排版软件及模板的使用方法 | 3 |
| 2.1 软件和模板使用方法 | 3 |
| 2.1.1 在服务器上使用 \LaTeX | 4 |
| 2.1.2 在自己电脑上安装 <code>texlive</code> | 4 |
| 2.2 关于 <code>lyx</code> | 5 |
| 第三章 表格、图片和代码的格式 | 6 |
| 3.1 表格 | 6 |
| 3.2 图片 | 7 |
| 3.3 代码 | 7 |

代码目录

| | | |
|------|--|---|
| 2.1 | 直接使用 Helium 上的 L ^A T _E X | 4 |
| 2.2 | 安装 L ^A T _E X 模板 | 4 |
| 2.3 | 直接使用 Helium 上的 L ^A T _E X | 4 |
| 2.4 | 安装字体 | 4 |
| 2.5 | 安装 Texlive | 4 |
| 3.1 | ctable 语法 | 6 |
| 3.2 | ctable 示例 | 6 |
| 3.3 | 三线表的一般表格语法示例 | 6 |
| 3.4 | 插入图片语法示例 | 7 |
| 3.5 | Pascal 代码示例 | 7 |
| 3.6 | 引用代码的语法 | 7 |
| 3.7 | 引用代码示例 | 7 |
| 3.8 | 引入代码文件示例 | 8 |
| 3.9 | C 代码示例 | 8 |
| 3.10 | Python 代码示例 | 8 |
| 3.11 | bash 代码示例 | 9 |

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：

1. 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。
2. 公司对外的书面交流文档。
3. 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：

1. 密级为“秘密”及以上的文档。
2. 无存档价值的内部交流文档。

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第二部分。

1.3 文件存档

1.3.1 应有文件

要求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. 生成这个 pdf 文档所需要的所有 tex 文件、图片文件及其他文件如插入代码、原始数据等，并压缩成 zip 或 gzip 包。
3. 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。
4. 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。
5. 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。

用于生成 \TeX 文件的过程文件，比如 lyx 文件，不需要存放。

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/QP/C-QP-001/V01/`。其三层路径名：

- QP 表示文档类型为质量流程，
- C-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 |
| DI | 开发文档 | DI |
| QP | 质量流程 | QP |
| 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，但只能由字母、数字或下划线（_）组成，不可以出现其他字符。

1.3.5 入档和查询方法

文件入档，请先在 helium 下测试是否能生成 pdf 文档，然后将文件拷给公司服务器管理人员，并给出如下信息：作者、文档类型、题目、关联的项目，如果是已有文档的版本更新，还要给出文档编号。

文件查询，即想找到文档对应的编号，可以打开文件 `/home/public/document/index.html` 来查询。

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。

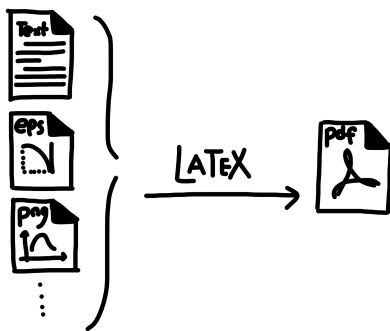


图 2.1: \LaTeX 工作方式

公司选用 \LaTeX 并采用统一的模板，好处为：

1. 统一公司文档的风格。
2. 方便不同的人在不同平台下编辑和修改同一个文档。
3. 相比其他排版软件， \LaTeX 崩溃的概率小；文本文档损坏的概率也小。
4. \LaTeX 排出的版面更漂亮
5. \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。（跟 Linux 一样， \LaTeX 常用的语法也就那么几页纸的事 [1]。）
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 软件和模板使用方法

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 L^AT_EX

模板及示例文件在服务器的 `/home/public/document/template` 路径下。

首先，我们需要设置使用 TexLive 软件所需的环境。

```
1 # Set the path environment of texlive
2 source /usr/local/texlive/setenv.sh
3 # Or to make it permanent, you can do:
4 # cat /usr/local/texlive/setenv.sh >> $HOME/.bash_profile
```

代码 2.1: 直接使用 Helium 上的 L^AT_EX

其次，每个用户需要将最新的模板文件安装到 `$HOME/texmf/tex/latex/cgdrep` 目录下，其步骤如下：

```
1 # Unpack the template
2 tar xf /home/public/document/template/cgdrep.tar.gz
3 cd cgdrep
4 # Install the template files
5 ./install.sh
```

代码 2.2: 安装 L^AT_EX 模板

现在，我们可以使用 L^AT_EX 模板的示例了：

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

代码 2.3: 直接使用 Helium 上的 L^AT_EX

2.1.2 在自己电脑上安装 texlive

目前我们使用的 texlive 是个未经简化的庞大的包，iso 镜像文件 2.4G，安装后需要 3.5G 的硬盘空间。镜像文件下载路径为：`/home/public/software/tex/texlive2013`。对于 Linux 系统，可能需要几个字体文件，文件在 `/home/public/software/tex/texlive2013/win_fonts.tar.gz`。

假设你准备把字体安装在 `/usr/share/fonts/TTF/`。安装为：

```
1 # copy the font files
2 tar -xzf win_fonts.tar.gz && sudo cp -r win_fonts /usr/share/fonts/TTF/
3 # active the fonts
4 sudo fc-cache -fv
```

代码 2.4: 安装字体

假设你准备把 texlive 安装在 `/usr/local/tex`。方法为：

```
1 # mount the iso file
2 sudo mount -o loop texlive2013-20130530.iso /mnt/dvd && cd /mnt/dvd
3 # install it
4 export TEXLIVE_INSTALL_PREFIX=/usr/local/tex
```

```
5 perl install-tl
6 # then follow the instructions to set the path environment
7 export PATH=/usr/local/texlive/2013/bin/x86_64-linux:$PATH
```

代码 2.5: 安装 Texlive

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

2.2 关于 lyx

lyx 是一个直观的编辑 \TeX 文件的编辑器。

不需要 lyx 的人可以跳过这用节¹。

¹脚注示例。

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，我们要求表格采用居中三线表，样式如表 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：

```
1 \begin{ctable}{Label}{Caption}{Alignments}
2  第一行（表格项目）  \\ \hline
3  第二行 \\
4  .....
5 \end{ctable}
```

代码 3.1: `ctable` 语法

需要注意的是这里的 `Alignments` 不是常见的 `{llll}`。（表示此表格有四列，都居左。）而是用：

`{+l^l^l^l}`

来表示四列居左，这个怪样子是为了实现第一行的黑体字。表 3.1 所用的语法为：

```
1 \begin{ctable}{triline}{三线表}{+l^l^l^l}
2  版本 & 日期      & 负责人      & 备注 \\ \hline
3  1.0  & 2013.12.24 & 沈忱、纪冬梅 & 初稿 \\
4  1.1  & 2013.12.25 & 赵军        & 二稿  \\
5 \end{ctable}
```

代码 3.2: `ctable` 示例

如果你不想局限于 `ctable` 的功能，也可以用一般的表格语法，表 3.1 的一般语法为：

```
1 \begin{table}[htbp]\caption{\label{triline} 三线表}
2 \centering\small\begin{tabular}{+l^l^l^l}\thickhline\rowstyle{\bfseries}\bf
3  版本 & 日期      & 负责人      & 备注 \\ \hline \rm
```

```

4 1.0 & 2013.12.24 & 沈忱、纪冬梅 & 初稿 \\
5 1.1 & 2013.12.25 & 赵军 & 二稿 \\
6 \thickhline\end{tabular}\end{table}

```

代码 3.3: 三线表的一般表格语法示例

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：

```

1 \begin{figure}[htbp]\centering
2 \includegraphics{file.jpg}
3 \caption{\label{Label}Caption}
4 \end{figure}

```

代码 3.4: 插入图片语法示例

3.3 代码

先给出一段文档中插入 Pascal 代码的样子：

```

1 for i:=maxint to 0 do
2 begin
3 { do nothing }
4 end;
5 Write ( ' Case insensitive ' );
6 Write ( ' Pascal keywords . ' );

```

代码 3.5: Pascal 代码示例

引用代码的 \LaTeX 语法为：

```

1 \begin{lstlisting}[language=Language,caption={Caption},label=Label]
2 .....
3 ( 代码内容 )
4 .....
5 \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：

```

1 \begin{lstlisting}[language=Pascal,caption={Pascal 示例代码},label=PascalExample]
2 for i:=maxint to 0 do
3 begin
4 { do nothing }
5 end;
6 Write ( ' Case insensitive ' );
7 Write ( ' Pascal keywords . ' );
8 \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：

```
1 \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。

```
1 #include <stdio.h>
2 #define N 10
3 /* Block
4  * comment */
5
6 int main()
7 {
8     int i;
9
10    // Line comment.
11    puts("Hello world!");
12
13    for (i = 0; i < N; i++)
14    {
15        puts("LaTeX is also great for programmers!");
16    }
17
18    return 0;
19 }
```

代码 3.9: C 代码示例

```
1 class BankAccount(object):
2     def __init__(self, initial_balance=0):
3         self.balance = initial_balance
4     def deposit(self, amount):
5         self.balance += amount
6     def withdraw(self, amount):
7         self.balance -= amount
8     def overdrawn(self):
9         return self.balance < 0
10 my_account = BankAccount(15)
11 my_account.withdraw(5)
12 print my_account.balance
13 import unittest
14 def median(pool):
15     copy = sorted(pool)
16     size = len(copy)
17     if size % 2 == 1:
18         return copy[(size - 1) / 2]
19     else:
20         return (copy[size/2 - 1] + copy[size/2]) / 2
21 class TestMedian(unittest.TestCase):
```

```

22     def testMedian(self):
23         self.failUnlessEqual(median([2, 9, 9, 7, 9, 2, 4, 5, 8]), 7)
24 if __name__ == '__main__':
25     unittest.main()

```

代码 3.10: Python 代码示例

```

1  #!/bin/sh
2  # renna: rename multiple files according to several rules
3  # written by felix hudson Jan - 2000
4
5  #first check for the various 'modes' that this program has
6  #if the first ($1) condition matches then we execute that portion of the
7  #program and then exit
8
9  # check for the prefix condition
10 if [ $1 = p ]; then
11
12 #we now get rid of the mode ($1) variable and prefix ($2)
13     prefix=$2 ; shift ; shift
14
15 # a quick check to see if any files were given
16 # if none then its better not to do anything than rename some non-existent
17 # files!!
18
19     if [ $1 = ]; then
20         echo "no files given"
21         exit 0
22     fi
23
24 # this for loop iterates through all of the files that we gave the program
25 # it does one rename per file given
26     for file in $*
27     do
28         mv ${file} $prefix$file
29     done
30
31 #we now exit the program
32     exit 0
33 fi
34
35 # check for a suffix rename
36 # the rest of this part is virtually identical to the previous section
37 # please see those notes
38 if [ $1 = s ]; then
39     suffix=$2 ; shift ; shift
40
41     if [ $1 = ]; then
42         echo "no files given"

```

```
43     exit 0
44     fi
45
46 for file in $*
47 do
48     mv ${file} $file$suffix
49 done
50
51 exit 0
52 fi
53
54 # check for the replacement rename
55 if [ $1 = r ]; then
56
57     shift
58
59 # i included this bit as to not damage any files if the user does not specify
60 # anything to be done
61 # just a safety measure
62
63 if [ $# -lt 3 ] ; then
64     echo "usage: renna r [expression] [replacement] files... "
65     exit 0
66 fi
67
68 # remove other information
69 OLD=$1 ; NEW=$2 ; shift ; shift
70
71 # this for loop iterates through all of the files that we give the program
72 # it does one rename per file given using the program 'sed'
73 # this is a single command line program that parses standard input and
74 # replaces a set expression with a give string
75 # here we pass it the file name ( as standard input) and replace the nessesary
76 # text
77
78 for file in $*
79 do
80     new='echo ${file} | sed s/${OLD}/${NEW}/g'
81     mv ${file} $new
82 done
83 exit 0
84 fi
85
86 # if we have reached here then nothing proper was passed to the program
87 # so we tell the user how to use it
88 echo "usage;"
89 echo " renna p [prefix] files.."
90 echo " renna s [suffix] files.."
```



```
91 echo "renna r [expression] [replacement] files.."
92 exit 0
93
94 # done!
```

代码 3.11: bash 代码示例

参考文献

- [1] T. Oetiker, H. Partl, I. Hyna, and E. Schlegl, “The not so short introduction to LATEX2 ϵ .” available online: <http://www.tex.ac.uk/tex-archive/info/lshort>, 1995.