Linux Foundation LaTeX Formatting and Style CheatSheet
Version 6.0

Jerry Cooperstein
Behan Webster

04/05/2021

Very Important

This document is not a substitute for thoroughly reading the Full Documentation in LF_COURSEWARE.pdf. It supplements that by giving an abbreviated set of examples.

Contents

1 Informational Boxes

All the following box-making macros are used in the form:

```
\begin{boxname}
   any useable cotent
\end{boxname}
```

The all have a default label and icon at the top left. You can override the label. So compare

```
\begin{info}
   Some useful information to take note of
\end{info}
```

Please Note

Some useful information to take note of

with

```
\begin{info}[Only Morons Would Fail to Note]
   Some useful information to take note of
\end{info}
```

Only Morons Would Fail to Note

Some useful information to take note of

The only exception is the `lfbox` which has no default title or icon, but a title is often added.

1.1 Information Boxes

lfbox

Kernel Version Note

kvnote

Please Note

info

Very Important

important

Don't do this

dontdothis

How it works

howitworks

Question

question

make menuconfig

configbox

You could also get this upstream from git

gitbox

Video Demonstration Resources

videobox

1.2 Location Boxes

On the Web

webbox

On the command line

cli

On Your Laptop

onlaptop

On Your Server

onserver

On Your Virtual Machine

onvm

On Gentoo

gentoobox

On Linux Host Machine

onhost

On Linux Mint

linuxmintbox

On Embedded Target

ontarget

On RedHat

redhatbox

On the Serial Console

serialbox

On RedHat, Centos, or Fedora

redhatfamilybox

On Container

oncontainer

On RedHat, Centos, Fedora, or openSUSE

rpmfamilybox

1.3 Distro boxes

On Arch Linux

archlinuxbox

On openSUSE

opensusebox

On CentOS

centosbox

On Ubuntu

ubuntubox

On Debian

debianbox

1.4 Boxes that show file content

filebox

On Debian, Ubuntu, or Linux Mint

debianfamilybox

Shell code

shbox

On Fedora

fedorabox

1.5 Distros

\archlinux	**Arch Linux**
\centos	**CentOS**
\debian	**Debian**
\debianfamily	**Debian**, **Ubuntu**, or **Linux Mint**
\fedora	**Fedora**
\gentoo	**Gentoo**
\linuxmint	**Linux Mint**
\redhat	**RedHat**
\redhatfamily	**RedHat**, **CentOS**, or **Fedora**
\rpmfamily	**RedHat**, **CentOS**, **Fedora** or **openSUSE**
\opensuse	**openSUSE**
\ubuntu	**Ubuntu**

2 File Listing Boxes

2.1 File Type Specific Listings:

For each type of file for which there is specific support, there are two related macros that can be used. As an example you can do:

```
\cfile{lab_foo.c}
```

where `lab_foo.c` is an included file, which might give:

```c
#include <stdio.h>
#include <stlib.h>
#include <unitsd.h>

int main(int argc, char **argv)
{
        printf("Hello LF Course Maintainer\n");
        exit(EXIT_SUCCESS);
}
```

As a very useful variation you can put a title in as in

```
\cfile[\texttt{lab\_foo.c}]{lab_foo.c}
```

lab_foo.c

```c
#include <stdio.h>
#include <stlib.h>
#include <unitsd.h>

int main(int argc, char **argv)
{
        printf("Hello LF Course Maintainer\n");
        exit(EXIT_SUCCESS);
}
```

where you should note the optional title comes **before** the file name, and we have used verb?_? for the special character in the file title, but do not to do for the name.

Very Important

These macros do syntax highlighting and parse the code encapsulated. If you give incorrect or incomplete code, you may get LaTeX errors that can be hard to fix. In that case you would be better off just using a simpler verbatim environment, such as `\kcode`

If you want to place the file in line instead of in a separate code you would use the related `clst` enviroment as in:

```
\begin{clst}
int main(int argc, char **argv)
{ }
\end{clst}
```

```c
int main(int argc, char **argv)
{ }
```

```
\begin{clst}[\texttt{lab\_foo.c}]
int main(int argc, char **argv)
{ }
\end{clst}
```

lab_foo.c

```c
int main(int argc, char **argv)
{ }
```

To save repetitive space, we will just give the list here without showing examples. You should experiment and see how you like them and if they are usefull to you.

- **Any File**
  `fromfile()` and `filelst`
- **C, C++**
  `cfile{}` and `clst`
- **Device Tree**
  `fdtfile{}` and `fdtlst`
- **LaTeX**
  `latexfile{}` and `latexlst`
- **Makefile**
  `frommakefile{}` and `makelst`
- **Bash Shell Script**
  `shfile{}` and `shlst`
- **YAML**
  `yamlfile{}` and `yamllst`
- **JSON**
  `jsonfile{}` and `jsonlst`
- **Python**
  `pytonfile{}` and `pythonlst`

2.2 Location-Specific (and other) Listings:

There are some other variations which look like location boxes, but differ in that they are verbatim environmets. Compare:

```
\begin{onlaptop}
some code with  \LaTeX\ macros and commands
\end{onlaptop}
```

On Your Laptop

some code with LaTeX macros and commands

with

```
\begin{onlaptoplst}
some code with  \LaTeX\ macros and commands
\end{onlaptoplst}
```

On Your Laptop

```
some code with  \LaTeX\ macros and commands
```

2.3 Variations

- **Laptop**
  `onlaptopfile{}` and `laptoplst`
- **Server**
  `onserverfile{}` and `serverlst`
- **VM**
  `onvmfile{}` and `vmlst`
- **Target**
  `ontargetfile{}` and `targetlst`
- **Host**
  `onhostfile{}` and `hostlst`
- **Serial**
  `onserialfile{}` and `seriallst`
- **CLI**
  `clifile{}` and `clilst`
- **Git**
  `gitfile{}` and `gitlst`

3 Verbatim Enviroments

3.1 Environments that Do Not Interpret commands: `cmd`, `response`, `kcode`

There are many cases where you want to insert text which is not interpreted by LaTeX, and where you want to use a monospace font. This is always useful for commands and output etc

This is also used for short or syntatically incomplete listings to display literal code and text without boxing and titles etc as we have shown up to now.

The basic ones to use are:

- `cmd`
  Used for commands typed at terminal
- `response`
  Used for output on terminal in response to commands
- `kcode`
  Used for anything else that you want literal

These differ in colors, which may change at some point in the future. **Examples:**

```
  \begin{cmd}
$ wc /etc/passwd
  \end{cmd}
  \begin{response}
52  122 2952 /etc/passwd
  \end{response}
```

rendering

```
$ wc /etc/passwd
```

```
52  122 2952 /etc/passwd
```

Notice the change in color between `cmd` and `response`.

If you do not want to have a change in color or use for other things kcode is just fine and was used above to show the `cmd` and `response` dialogue.

```
\begin{kcode}
anything you want
\end{kcode}
```

renders as

```
anything you want
```

3.2 Environments That Interpret Commands: `cmdtt`, `responsett`

Sometimes you want a command interpreted inside a verbatim-like environment. For that we use some custom environments similar to the `alltt` one in standard LaTeX. Compare

```
  \begin{cmd}
$ echo This is \mycourse
  \end{cmd}
```

rendering

```
$ echo This is \mycourse
```

with

```
  \begin{cmdtt}
$ echo This is \mycourse
  \end{cmdtt}
```

rendering

```
$ echo This is LFD999
```

Note that in the `cmdtt` environment there can be subtle differences. For example, if you just use \ instead of `\textbackslash` it will probably put everything on one line and drop the backslash! So you have to inspect output and may have to deal with special characters.

One also has a `responsett` environent which is not used anywhere near as much.

Indentation & Vertical Spacing

These environments can be sensitive to horizontal indentation. If `\begin{..}` or `\end{..}` for `cmd`, `cmdtt`, `response`, or `responsett`, begin at the beginning a line, sometimes the vertical spacing afterwards gets compressed. This is probably a bug not a feature we hope to fix at some point.

3.3 Including Entire Verbatim files

These macros do not interpret LaTeX coding. interpretaton of any commands within. Most common will be `rawoutfile{}` macro to show the output of a command or precess.

`\rawcmdfile{somefile}`
gives a sequence of command line input in blue

`\respfile{somefile}`
give a series of output lines in purple

The only difference in these forms here is font color.

You can always use one of the other previously discussed macros if you want to put the output in a box or use a title for it