Table of Contents

CMS Paper Formatting Guide

The CMS tdr environment provides a wide variety of useful macros and tools for preparing your CMS notes and papers. Unfortunately, many people do not become aware of these until they are already well into the process, requiring additional effort to change the paper to use the CMS standards. This page is intended as a guide for people starting a new note or paper, so you can write things the right way the first time around and save yourself the trouble of having to change later.

This page is intended as a highlight of the most important things to be aware of; it is not a complete reference of all of the CMS style and formatting guidelines. For that, please consult the pages below:

- PubComm guidelines: PubComm style guide, figure guidelines, citation rules
- Paper submission rules: Prep checklist, LaTeX formatting checklist, references checklist, table guidelines (Note: many of the things on the prep checklist, such as combining into a single LaTeX file, should of course not be done until the very end of the preparation process when the paper is submitted, but the formatting rules are useful throughout the whole process.)

The formatting rules as described below are mandatory for papers; they are not mandatory for notes or internal documents, but it's generally a good habit to use these there as well, both to get into good practice and to make it easier to transfer text from a note to a final paper.

Standard macros

Many LaTeX macros are available to simplify and standardize your paper format. Use of these macros is **mandatory** for paper submission, so it is a good idea to start writing using these macros so you don't have to change over later. A full list of the available macros can be found in the template PDF file available here; the below summarizes the most important points:

Particle names

These macros should *always* be used for particle names, not only in expressions (`$\PZ \to \Pe\Pe$`) but also in text (`leptonic decays of \PZ bosons`). Note that there is no need to put these in math mode if they are in text (`leptonic decays of \PZ bosons`, not `$\PZ$ bosons`), except for the case when you have multiple particle names next to each other (`data collected in $\Pp\Pp$ collisions`). This is only a quick summary of the most common particles; a wide variety of more particles, including supersymmetric, fourth-generation, or other hypothesized particles, is available.

- Quarks: `\PQu`, `\PQd`, etc. for quarks; `\PAQu`, `\PAQd`, etc. for antiquarks; `\PQq` or `\PAQq` for a generic quark (q)
- Leptons: `\Pl` or `\Pell` for a generic lepton (in regular type or script, respectively); `\Pe`, `\PGm`, `\PGt` for electron, muon, tau; add `p`, `m`, `pm`, or `mp` for a sign; `\PGn` for a generic neutrino, or `\PGne`, `\PGnGm`, `\PGnGt` for a specific neutrino (`\PAGn`, etc. for antineutrinos)
- Bosons: `\PW`, `\PZ`, `\PH`, `\Pg` for gluon, `\PGg` for photon. You can add a sign as above, or `z` for a zero (`\PWp` for W^+, `\PZz` for Z^0, etc.)
- Baryons and mesons: `\Pp` and `\Pn` for protons and neutrons, `\PGp`, `\PGpp`, `\PGpz`, etc. for pions, `\PK`, `\PKp`, `\PKz`, `\PKL`, `\PKS`, etc. for kaons, `\PJGy` for J/ , etc.

Other common expressions

Many common CMS expressions are also available as standard macros: `\pt`, `\HT`, `\ptmiss`, `\CLs`, `\kt`, etc. Also, use `\abs{...}` rather than `|...|` for absolute value.

Units

When writing units, always use the appropriate macro **with no space** between the number and the unit macro. Most common units are available as special macros, so `7\TeV`, `20\mum`, `139\fbinv`, etc. If you need a unit for which there isn't a predefined macro, use the `\unit` macro: `40\unit{MHz}`. If you want to use the unit with no preceding space (for instance, in a table header), use the `ns` variant: `Lepton momentum (\GeVns)`. (If you use brackets instead, you'll also need to add `{}` afterwards so you don't get an extra space after: `Lepton momentum [\GeVns{}]`. See also note on spacing below.)

MC generators

Macros are also provided for common MC generators and other software packages: `\PYTHIA`, `\MGvATNLO`, `\FASTJET`, `\GEANTfour`, etc. Note that PDF sets don't have these macros and should just be written in all caps.

Creating your own macros

For expressions that you will use frequently, it is strongly recommended to create your own macros. In general, you want to wrap your macro in `\ensuremath{...}` and follow it with `\xspace`, so that it can easily be used in either text or math mode: `\newcommand{\BbH}{\ensuremath{\PB \to \PQb\PH}\xspace}`

Many authors like to create macros for their final result numbers, so that if these should change during the process, they can easily be updated in all places that they appear. In this case you can skip the `\ensuremath` since it doesn't matter whether these are in math mode or not: `\newcommand{\masslimit}{1050\GeV\xspace}` (You may or may not want to include the units in the macro, depending on how exactly you plan to use it. See also the note on spacing below.)

Usage of math mode

The full guidelines for using math mode can be found in the LaTeX formatting checklist, but the important points can be summarized as follows:

- Don't use math mode when not necessary, so `We require that the lepton \pt be greater than 50\GeV`, not `$\pt$` or `$50\GeV$`.
- If there is an expression containing =, <, etc. with material on both sides of the operator, put the whole expression, including both sides, in math mode: `We require $\pt > 50\GeV$`, not `\pt $>$ 50\GeV`.
- However, if the expression only has material on one side of the operator (this also applies to operators such as `\approx` or `\sim`), then *only* the operator should be in math mode: `We require a relative isolation $<$0.4`, not `$<0.4$`. (If the expression on the right needs to be in math mode, then you can achieve the same result by using `{<}` inside math mode: `The value should be ${<} \alpha^2$`.)

A note on spacing

The `\xspace` macro tries to be smart about when to insert a space and when not, but it is not always successful. As mentioned above, if you have multiple particle names in a row, they should be set in math mode to avoid extra spaces. If you get an undesired space elsewhere, you can also suppress it by adding `{}` after the macro name. You can also fix this permanently by adding the appropriate symbol to the `\xspace` exceptions; see "Advanced tricks" below.

Standard text

The PubDetector twiki page provides many bits of standardized text for use in your note or paper. For the detector description, you should use the text given on this page (selecting the parts of the detector description that are most appropriate to your analysis). There are also standard descriptions of various algorithms (such as particle flow, soft drop, PUPPI, etc.) that you can also use. It is strongly recommended that you at least start with the text on this page, although of course you should feel free to customize it to your own needs.

References

Correct formatting of references can be quite time consuming. To save yourself some time, there are many resources to help you. First, there is standard BibTeX provided for many common references:

- General references: PubGuidelines
- Statistics references: StatisticsReferences
- MC generator references: CitationsForGenerators

In addition, many other references are also available in the template .bib file provided in the tdr repository here.

Of course, many references you will need to add yourself. Unfortunately, while many sites offer BibTeX output, often this output is not very compatible with the CMS style. In general, inSPIRE produces output which is reasonably close to CMS style, so I would recommend starting there. However, you will need to make the following changes:

- If there is a `number` field, delete it.
- The `pages` field will often contain a range (e.g. `274-277`). In this case, delete the rest of the range and just leave the starting number (`274`).
- For journals that contain a letter in the name, like Eur. Phys. J. C or Phys. Rev. D, inSPIRE will attach this letter to the volume number, but CMS prefers to attach this to the journal name, as below:

inSPIRE format (needs to be fixed)	**CMS format** (correct)

`journal = "Eur. Phys. J.",`
`volume = "C79",`
	`journal = "Eur. Phys. J. C",`
	`volume = "79",`

Note that newer inSPIRE references appear to have this issue fixed, thankfully, but be sure to keep an eye out for it.

In cases where the document is only available on CDS (for example, PASes), you can start with the BibTeX available on CDS, but you will need to make the following changes:

- Delete the `institution`, `address`, `reportNumber`, and `month` (if present) fields.
- Change the `collaboration` field to simply `"CMS"` and add `author = "{CMS Collaboration}",`

You might find it easier to start with the entry for an existing PAS and modifying the title, number, year, and URL to the new reference.

Titles

The formatting of the paper title should also be checked. Oftentimes, the entire title will be enclosed in braces, which may mean that the title will be incorrectly capitalized, depending on the bibliography style. Instead, you should format the title so that *only* things that should be uppercase are enclosed in braces, as below:

- *wrong*: `title = "{The CMS Experiment at the CERN LHC}",`
- *right*: `title = "The {CMS} Experiment at the {CERN} {LHC}",`

Outside braces, it (usually) doesn't matter if a word is capitalized or not, since it will be automatically capitalized or lowercased as appropriate for the bibliography style, so you don't need to change "Experiment" in the example above (although of course you can if you want to).

Automatic reference checking

Building your note or paper in "preview" mode will also run a script which checks your .bib file for many common issues (including most of the above). See the "Tools" section below.

Tools

There are a few scripts to help automate some of the formatting tasks. They should be in the `utils/` directory of your note. (If they're not there, you may need to update your utils -- follow the directions in the "Updating utils" section of TdrProcessing.)

- **Preview mode**: Building your note or paper in preview mode will run an automated tool to check your references. To do this, simply add the `--preview` option to your tdr build command. For documentation on the various issues detected by this tool and how to resolve them, see PaperSubmissionPrepPreview.
- **find_8bit.py**: This is a simple script to find non-ASCII characters in your .tex files, which can cause LaTeX compilation errors (see below). Run it with `python3 utils/general/find_8bit.py mytex.tex`. Further documentation is available at PaperSubmissionPrepNonAscii.
- **matchTeXcommands.py**: This script checks for any unused LaTeX macros in your file. For directions, see PaperSubmissionPrepUnusedCommands.

LaTeX compilation errors and warnings

If there are any LaTeX errors in your paper, they will need to be fixed prior to submission, and in general, it's good practice to minimize errors so you can spot things that are actually problems. Here's a quick guide to some common warnings and errors and how to resolve them.

- `Warning--empty institution in CMS-PAS-LUM-18-002`: If you correctly format your PASes according to CMS style, they will not have an institution included, which will cause this warning to be produced. Newer versions of the tdr utils have gotten rid of this warning, so the simplest way to deal with it is simply to update your utils directory following the directions above.
- `PDF inclusion: found PDF version <1.7>, but at most version <1.5> allowed`: This is also a relatively harmless error, but it's easy to fix. Just use ghostscript to convert your PDF file back down to a version 1.5 file as follows: `gs -sDEVICE=pdfwrite -dCompatibilityLevel=1.5 -dNOPAUSE -dQUIET -dBATCH -sOutputFile=NewFile.pdf FileName.pdf` (and then just overwrite the old file with the new file)
- `Missing character: There is no   in font pplr7t!`: This is caused by having non-ASCII characters in your input. Use the script `find_8bit.py` described above to find these and change them to proper LaTeX accented characters. This can also be caused by having a less-than or greater-than sign outside of math mode, which doesn't work in LaTeX.
- `There were multiply-defined labels`: The brief output that the tdr script gives you doesn't actually tell you which labels are multiply defined, but if you go to your output directory and look at your XXX-YY-NNN_temp.log and look for "multiply", you can find the actual labels that have the problem and fix them.
- `PDF inclusion: multiple pdfs with page group included in a single page`: This problem is caused by PDFs which have multiple different groups defined. In general, it's simplest to fix this by fixing it in whatever program that was used to generate them. You can also use a ghostscript command like the above but with version 1.3; be sure to check your output after doing this to make sure that it is correct. However, if these solutions are not possible, you can also add the command `\pdfsuppresswarningpagegroup=1` at the top of your file, where you define commands, to suppress these warnings.

Advanced tricks

- Fixing issues with `\xspace`: Most CMS macros end with `\xspace` so that they can be used in text mode without the following space disappearing, and it's recommended that you do the same with the own macros that you define. Unfortunately, the algorithm that `\xspace` uses to determine whether or not there should be a space afterwards is not very sophisticated, and in some cases it will insert a following space where you don't want one. For individual instances, you can fix this by adding `{}` after the macro name (e.g. `[\GeVns{}]` to get a properly formatted "[GeV]"), but you can fix this permanently by adding the appropriate characters or macros to not insert space before to an `\xspaceaddexceptions` command at the top of your file (with the rest of your macro definitions). For example, `\xspaceaddexceptions{]\Pp}` will tell `\xspace` to not add a space before a close bracket (so you can just write `[\GeVns]` in the above example) or `\Pp`, so you can write `\Pp\Pp` without it having to be in math mode. (Be careful with things like the latter case, though, since now if you write something like `14\TeV \Pp\Pp collisions`, there won't be a space after the "TeV".)
