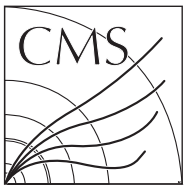

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

2009/02/20- Revision 1.48 - checkout 2009/02/20 13:30:04

TDR: the Technical Document Repository System
for the storage, concurrent access, and building of CMS reports, notes, and other LaTeX-based documents

George Alverson and Lucas Taylor
Northeastern University

Abstract

This note describes the TDR documentation system for LaTeX-based documents including CMS Technical Design Reports (TDRs), Expressions of Interest (EoIs), Letters of Intent (LoIs), CMS Notes, Internal Notes, and Analysis Notes. It describes the `TDR` cvs repository for the storage and concurrent multi-user access of documents and the use of the `tdr` build tool for compiling complete or partial documents from users' LaTeX source and graphics files. This system has been successfully used by hundreds of authors of the CMS Computing TDR, the Physics TDR, and a number of other documents. (See also: `http://cmsdoc.cern.ch/cms/cpt/tdr/`)

Contents

1 Overview

The CMS Technical Document Repository (TDR) system provides a straightforward environment for the preparation of reports and notes by large numbers of authors working concurrently. It comprises the following components:

1.1 TDR Document Repository

All files that are required for the assembly of completed documents are stored in a central CMS cvs repository (called `TDR`). The repository contains the common style files and build tools as well as all the user-generated text (LaTeX) files and figures. This system facilitates the sharing of documents, concurrent working, and means that users do not need to keep any files in their private area.

1.2 Document style files

Common LaTeX style files have been pre-defined for CMS Technical Design Reports (also used for EoIs, LoIs, and other large documents), CMS Notes, Internal Notes, Analysis Notes, and Conference Reports. Template examples are provided enabling the user to get started with minimal overhead.

1.3 Document build system

The philosophy of the TDR system is to keep the LaTeX document style commands distinct from the user-content. A `tdr` perl script is then provided that assembles on the fly a complete LaTeX document using pre-existing standard fragments and the users' LaTeX files. It then proceeds to build the document by processing the LaTeX, resolving cross-references and citations (using BibTeX), and creating a PDF (portable document format) file. The user selects the style of the document (CMS Note, Analysis Note, etc.) by specifying an option to the `tdr` command. It is therefore totally trivial to switch from one style to another.

1.4 External software

The system is designed to be independent of the CMS environment. All that is required is `cvs`, `perl`, and a standard installation of LaTeX. These are already part of the standard CERN Linux environment.

1.5 Getting started

To **create a new document** in the repository, for example a CMS Note, see section 2.

To **edit the document** once the template has been created, see section 3.

To **build a formatted manuscript** (PDF) for your document see section 4.

For **advice on using LaTeX**, for example to include figures, see section 5.

2 Creating a new document

All files reside in a standard CMS cvs repository (called `TDR`). On any machine with the CMS environment (e.g., `lxplus.cern.ch`) you can set the `CVSROOT` variable to point to the TDR repository by typing:

```
> project TDR
```

All this command does is set an environmental variable for the location of the cvs repository using Kerberos authentication: `CVSROOT=:gserver:isscvs.cern.ch:/local/reps/tdr`. Therefore, if you do not have the CMS-specific `project` command on your system, just set this variable yourself by hand (e.g., with `setenv` in csh). You also need to have a valid Kerberos ticket for the CERN.CH domain. SSH can also be used. (More information can be found on the central CERN cvs server page.)

2.1 Creating a new note or analysis summary

The first step for a new document is to create a template document in a new dedicated directory in the TDR cvs repository. To do this contact a manager of the TDR system – currently George.Alverson@cern.ch or Lucas.Taylor@cern.ch (this step may be more automated later). The directory will be created and you will be given access privileges. It will contain a template report or paper (LaTeX file) ready for you to start editing. Sub-directories are also created, e.g., for storing figures in PDF format. It is best to supply a list of the co-authors who are to have write access to the repository at the time of the request, although these can also be added later.

2.1.1 Naming convention for Analysis Notes and Physics Analysis Summaries

A new directory is created in the `TDR/notes` directory, named according to the convention chosen by the analysis group, e.g. `TOP-07-005`. Once created, this directory will contain a template note named according to the analysis name, e.g. `TOP-07-005.tex`. The `tdr` script will automatically generate the `cmsNoteHeader` from the directory name.

2.1.2 Special Note on Physics Analysis Summaries

PAS documents are loaded into the CDS archives after approval. At this point, the title *as stored in the `hypersetup pdftitle` field* is passed to CDS as the document title. This allows for a fully formatted LaTeX title on the document and a natural language title for easy searching. The abstract, on the other hand, is taken from the `abstract` LaTeX version. Both are run through a simple-minded pre-processor for display. The preprocessor will not see any TeX macros, however, so those should should not be used.

2.1.3 Naming convention for CMS Notes, Internal Notes, and Conference Reports

A new directory is created in the `TDR/notes` directory, named according to the convention: `contactAuthor_serialNo`. `contactAuthor` is the CMS username (see CERN "phone" command) which is used for subsequent access control. `serialNo` is a simple serial number (001, 002,...) for the note generated at the time of the request; it is *not* anything to do with the final CMS note number which will be assigned independently during the review process. For example the first note requested by Paris Sphicas resides in the directory `TDR/notes/sphicas_001`. Once created, this directory will contain a template note called `contactAuthor_noteNo.tex` and a sub-directory called `fig` in which figures (PDF files) may be stored.

2.2 Creating a new Technical Design Report (or LoI, EoI, etc.)

For major reports, a new directory is created at the top-level, e.g., `TDR/ctdr` for the computing technical design report. This directory will contains the following sub-directories:

- `tex` - latex files and subdirectories (e.g., for different chapters);
- `fig` - figure files and subdirectories;
- `bib` - bibtex file(s) for references.

Note that for TDRs this sub-structure is assumed to exist by the `tdr` script (described below); if you change it things may fail.

3 Modifying a document and working with cvs

To start working you first need to checkout some directories and files from cvs. "Checkout" means asking cvs to make you a local copy of the directories and files and also to keep track of who is changing what as several people may be working on the same document concurrently.

3.1 Defining the working area and repository location

Go into a suitable local working directory, e.g.,

```
> mkdir work; cd work
```

and define the location of the cvs repository you are working with:

```
> project TDR
```

which just sets the environmental variable for the location of the cvs repository: `CVSROOT = :gserver:isscvs.cern.ch:/local/reps/tdr`.

3.2 Checking out desired files

Checkout the directory which contains the source files of the document you wish to work on. For example, to checkout a note that was created as described above:

```
> cvs co -l notes
> cd notes/notes
> cvs update -d noteName
> cd noteName
```

where `noteName` is something like `sphicas_001` or `TOP_07_005`. In addition to your specific note directory, you will see the following general files/directories:

- `tdr` - a script for building documents (described below);
- `general` - a R/O directory containing the style files.
- `tmp` - a temporary directory used for output PDF, etc.

For the major reports (e.g., a Technical Design Report), cvs modules have been defined which checkout all the required files and directories for a given document in a single command, e.g.:

```
> cvs co thisReport
> cd thisReport
```

where `thisReport` is `ctdr` for the computing TDR; `ptdr1` for Physics TDR volume I; `ptdr2` for Physics TDR volume II; `slhc-eoi` for the SLHC Expression of Interest; `diffractive` for the diffractive document, and so on.

If for some reason you want to look at the whole repository, type:

```
> cvs co TDR
```

and be prepared for a long wait as the complete repository is large.

3.3 Editing the document

Simply edit any of the LaTeX files with your favourite text editor. For example, for a new note, start with the file `contactAuthor_noteNo.tex`.

3.4 Committing your changes into the cvs repository

Before committing any changes always check your changes are valid LaTeX, otherwise you will break the document for all other authors.

Firstly, check the local file, e.g., `myfile.tex` by doing `tdr build myfile`.

If `myfile.tex` is included in a bigger document, e.g., `ctdr.tex`, then you must also check that this builds: `tdr build ctdr`. In both cases you should check that a valid PDF file is produced that looks as expected. LaTeXis rather verbose with its warnings, however it is imperative to look and verify that there are no **error** messages, and no **unresolved** references.

Changes to files are committed to (i.e. stored into) the repository using

```
> cvs commit -m''Comment explaining changes made''
```

The -m option should always be used to add a short informative message.

Finally: do not forget to `cvs add` any new files to the repository. It is not sufficient to just do a `cvs commit`. New files must be first added and then committed.

3.4.1 Checking everything is OK with cvs

If you want to see the status of your local files compared to the repository type:

```
cvs -n -q update
```

This does not *DO* anything. It just tells you the status.

The first character of each line tells you the status of the file:

- U means the file needs updating from the repository (your version of the file is stale). To update use the `cvs update` command.
- M means you need to merge your changes with some changes somebody else has made to the repository version. Try to avoid doing this (messy) step by committing frequently.
- ? means you have a file locally that `cvs` knows nothing about. Maybe it's meant to be local (e.g., is temporary). If you want it to be in the repository then you must use `cvs add` and the `cvs commit`.

3.5 Creating a standalone paper, e.g., for submission to a journal

If you wish to export your paper (for publication, local work or for security), you can produce a tarball with all the necessary files with

```
> tdr --style=note --export b mynote.
```

This will function on Unix or Windows systems which have recent copies of LaTeX (including $\mathcal{AMS}$-LaTeX) and `perl` installed.

Please see also section 4.8 on formatting for journals.

4 Building a formatted manuscript

The LaTeX file(s) must be processed to produce a fully typeset and formatted manuscript in PDF (Portable Document Format). A `tdr` perl script is provided for building the whole or parts of your document, as described below. There is no need use any of the following commands yourself: `latex`, `pdflatex`, `pdftex`, `bibtex`, `dvips`, or `dvi2pdf`. They are all replaced by the `tdr` script.

4.1 Initializing your environment

Set up the runtime environment by typing:

```
> eval `./tdr runtime -sh`     // if you use Bourne-shell or Korn shell
> eval `./tdr runtime -csh`    // if you use c-shell or tc-shell
```

This must be done from the top-level directory of the checked out area, i.e. the location of the `tdr` script. Note also that the syntax uses single *back* quotation marks.

The `tdr` command has a simple scram-like syntax with `runtime`, `build`, `clean`, and `veryclean` commands, support for one-letter abbreviations and so on. For details on `tdr` options type:

```
> tdr help
```

4.2 Building a PDF file from a LaTeX file

To create a PDF file from a LaTeX file `myPaper.tex`, simply type:

```
> tdr build myPaper      (or simply:  tdr b myPaper)
```

Assuming the LaTeX files have no errors in them, the last line of the screen output will tell you the location of the output PDF file. It is stored in the top-level `tmp` directory together with various log files.

If the build fails, check the printout on the screen for LaTeX errors and resolve them; typically these are trivial syntax errors. Then run the build again.

4.3 Choosing the document style

You can choose to format the paper according to various pre-defined styles using the `style` option, for example:

```
> tdr  --style=note  build  myPaper
```

will format the paper as a CMS Note. Valid styles are

- `tdr` for large reports (the default),
- `note` for CMS Notes,
- `an` for Analysis Notes,
- `pas` for Physics Analysis Summaries,
- `cr` for Conference Reports, and
- `in` for Internal Notes.

Note that PAS documents can be in either draft mode (the default), or non-draft, as set by the `--nodraft` switch.

4.4 What your LaTeX files should (not) contain

The `tdr` script makes a copy of your simple LaTeX file and automatically inserts all the required LaTeX boilerplate commands to produce a fully consistent LaTeX document in the `tmp` directory, in accordance with the CMS document style requested in the command line options (see above). It then processes the document using PdfLaTeX with several passes to resolve cross references; citations are handled using BibTeX.

Therefore, it should be stressed that the file `myPaper.tex` should *not* contain any document definition commands (e.g., `\documentclass`, `\begin{document}` and so on).

4.5 Making partial builds

To speed things up, especially for large documents, the `tdr` command can build single chapters, sections, or indeed any arbitrary LaTeX files. For example, if your main file is called `myPaper.tex` and looked like:

```
\input{titlepage.tex}
\input{introduction.tex}
\input{data-analysis.tex}
\input{results.tex}
```

then you could use the following commands

```
> tdr build myPaper   // build everything as a single PDF paper
> tdr b results       // build just the results section as PDF
```

In general you should be in the directory in which the LaTeX file resides.

4.6 Setting the default file to build

To save specifying your preferred build target (e.g., `myPaper.tex`) each time, just set the Unix environmental variable `TDR_TARGET` to `myPaper`. Then you can just type

```
> tdr b
```

If `TDR_TARGET` has not been set, then `tdr` builds this document.

4.7 Cleaning up

To clean up temporary files (i.e the locally-created `tmp` directory):

```
> tdr clean
```

To clean up temporary files and emacs and nedit backup files:

```
> tdr veryclean
```

4.8 Submitting to Journals

Different journals have different formatting requirements and we have only tested against a limited set, so you will have to be responsible for knowing the specific requirements for your journal. What follows is only an example based on formatting the example note using RevTeX.

1. No multicol is allowed (in general, figure formatting does have to be re-examined to match journal requirements). Must be removed if used if not allowed by the journal (RevTeXsays no).

2. Use the abstract environment rather than the command:

```
\abstract{my abstract} -> \begin{abstract}my abstract \end{abstract}
```

3. Use the journal-required macros for author/title, too. Quite often they are the same as ours.

4. Replace the style file with journal style+key packages normally supplied by `cms-tdr`:

```
\documentclass[11pt,twoside,a4paper,pdftex,pas]{cms-tdr} ->

\documentclass[twocolumn,amsmath,amssymb]{revtex4}
\usepackage{xspace}
\usepackage[bookmarksnumbered,bookmarksopen,bookmarksopenlevel=1,
colorlinks=false,pdfborder=0,plainpages=false,
pdfpagelabels]{hyperref}
\usepackage{hypernat}
\usepackage{graphicx,graphics}
\providecommand{\cmsNoteHeader}[1]{\preprint{#1}}
\providecommand{\cmsNoteContact}[1]{\relax}
```

5. Comment out the `\RCS` commands. That way the file still has the `cvs` tag info, even if it is not displayed.

6. Delete the `editorOnSwitch`, editor and contributor.

Note also that for papers being submitted to arXiv, the instructions are located at `http://arxiv.org/help/submit_tex#pdflatex`. In particular, `\pdfoutput=1` should be located within the first 5 lines of the text so that arXiv can distinguish between PdfLaTeX and LaTeX.

5 Advice on using LaTeX

5.1 LaTeX macros for commonly used constructs

Provisions are made to implement macros across TDR volumes, within a volume, or even locally in a particular section. However, in order to establish a standard look and feel for the text symbols in the TDR volumes (such as for E_{T} and p_{T}), we encourage use of the generally defined macros and strongly discourage local use unless you are certain a similar symbol would not be used by another editor.

At the top-most level, definitions defined in `TDR/general/ptdr-definitions.tex` are available to all TDR volumes. An extensive set of macros have been defined there and should be used whenever possible. They include, for example, `\ET`, `\fbinv`, `\sTop`, etc. At the top-level of each TDR (e.g., in `TDR/ptdr1/tex/definitions.tex`, there is another file `definitions.tex` for volume-specific definitions. Macros should be suggested and implemented for frequently used constructs or common symbols or names, e.g., `\etc` could be defined to produce "*etc.*" and so on. The macros in the `definitions.tex` files are usable in tex files at all levels of the particular TDR.

Use `\newcommand` to define a new command that does not exist, `\renewcommand` to re-define a new command that already exists, or `\providecommand` to define a new command but accept the old definition without complaint if it has already been defined.

To override a general definition in `TDR/general/ptdr-definitions.tex` simply (re-)define it in the local `definitions.tex`. But please consult with the appropriate TDR editor.

5.2 Fonts

Do not override the default fonts. They are currently set to be Palatino and Helvetica. The math fonts have also been changed to Palatino so that they do not clash with the body text, particularly in regards to numbers and units. This means the authors should use `\text` commands to put text in subscripts and superscripts, and most importantly *do not use* `\rm` in formulas, otherwise you will end up with formulae looking like the second one below.

$$\phi = \text{a Greek letter} \tag{1}$$

$$\mathrm{\OE} = \text{a mistake} \tag{2}$$

It is also advisable to use the `\textrm{Some text}` form rather than `{\rm Some text}`. The same is true for the other short-form holdovers from plain TeX, `\tt` and `\it`, particularly if you would like to submit your paper to a journal with minimal re-editing.

5.3 Editorial macros

In addition to the extensive measurement and physics symbols, some editorial macros are defined in `TDR/general/definitions.tex` as well. For example, the following tex fragment:

```
\editor{Jane Doe} \\
\contributor{Tom Cobbley} \\
\fixme{check this number!} \\
```

produces the following.

Editor(s): Jane Doe

Contributor(s): Tom Cobbley

FIXME: check this number!

Notes use `author`, `address`, and `abstract` commands.

5.4 Inclusion of Figures

Figures should reside in the `fig` directory of the corresponding TDR (volume). A figure may be included as follows:

```
Figure~\ref{fig:test} shows a figure prepared with the TDR
template and illustrates how to include a picture in a document
and refer to it using a symbolic label.
\begin{figure}[!Hhtb]
   \centering
   \includegraphics{width=0.55\textwidth}{c1_BlackAndWhite}
   \caption[Caption for TOC]{Test of graphics inclusion.\label{fig:test}}
\end{figure}
```

The result of the above is roughly as follows:

> Figure 1 shows a figure prepared with the TDR template and illustrates how to include a picture in a document and refer to it using a symbolic label.

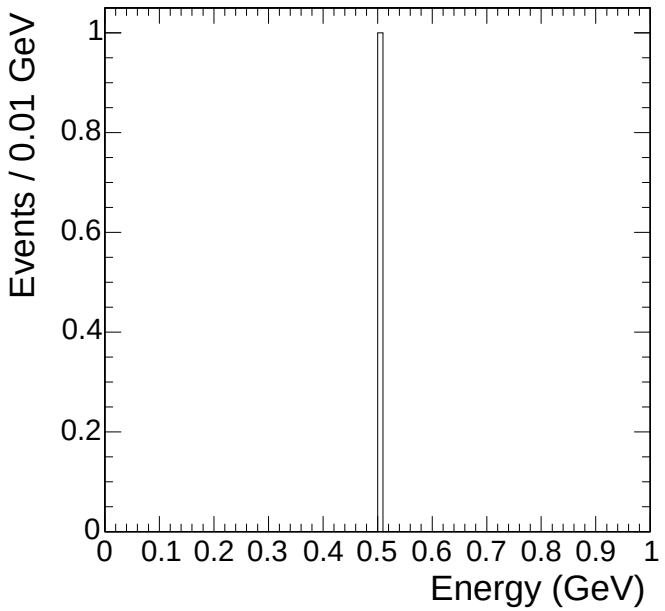

Figure 1: Test of graphics inclusion.

Note that the file extension (type) for the filename (e.g., `c1_BlackAndWhite.pdf` above) is not explicitly specified. Also note that authors should use an alternate short caption within the

first set of brackets when the complete caption is unduly long for including in the list of figures in the Table of Contents.

Also note that the current recommended size for figures is `0.55\textwidth` for square plots, and `0.7\textwidth` for ones with a standard (i.e. produced using the root template described in Section 5.4.5) rectangular aspect ratio.

Finally, note that correct results for the labeling occur only if you place the `label` command within the caption environment.

5.4.1 Colour Figures

Figures will generally be printed in black and white for paper versions of the final document. We have found that the automatic conversion of colour figures to black and white often results in a lack of legibility, so we recommend that all authors provide a black and white version for each figure which they have checked for legibility on an actual paper copy.

Colour versions of figures can by provided for PDF output using the `combinedfigure` macro in place of the `\includegraphics` command. This takes two arguments corresponding respectively to the black and white and the coloured versions of the same picture, for example:

```
Figure~\ref{fig:test} shows a figure prepared with the TDR
template and illustrates how to include a picture in a document
and refer to it using a symbolic label.
\begin{figure}[!Hhtb]
   \centering
   \combinedfigure{width=0.4\textwidth}{c1_BlackAndWhite}{c1_Colour}
   \caption[Caption for TOC]{Test of graphics inclusion.\label{fig:test}}
\end{figure}
```

Both figures should have the same size or the pagination may be affected.

5.4.2 How to include multiple figures

If you need to include multiple figures into the figure environment (i.e., you need only one common caption), the recommended procedure is to use multiple instances of the `\includegraphics` command, combined with the `tabular` environment if needed. Please do not use the `subfigure` environment just to get "(a)" and "(b)" labels, it is a waste of white space and does not look as nice as putting the labels directly on the plot. Moreover, do not use the `picture` environment to draw the labels, because the coordinate system is absolute on the page and not relative to where the figure will be placed (i.e. this only works for the very final version). In short, to label multiple figures, it is best to embed the label into the plot.

5.4.3 How to handle figures in PDF, jpeg, and PS formats

Files with extensions of `.pdf` (recommended) and `.jpg` are automatically picked up. Direct import of `.eps` files is not supported by the `pdftex` driver which is used to convert LaTeX to PDF. You are advised to convert your `.eps` file to a `.pdf` file using Adobe Acrobat (best results), the `epstopdf` command or `ps2pdf -dEPSCrop`, and commit that to `cvs`.[1] Try to

[1] An alternative approach would be to use LaTeX plus `pstopdf`. However, this often fails to produce correct `.ps` and hence `.pdf` output files; nor does it support the inclusion of `.pdf` or `.jpg` pictures which are generally much more compact than the corresponding `.eps` files.

avoid converting figures through an intermediate program, such as Powerpoint, and instead convert the natively produced Postscript. If you do convert an EPS file, you are encouraged to also commit the original EPS version as well in case of conversion problems found later. The editors may re-convert if necessary.

Also, keep in mind that some later versions of PDF (e.g., 1.5) will conflict with the PdfLaTeX machinery on many systems, including `lxplus` so please save figures (e.g., from Distiller) with version 1.3 or 1.4.

5.4.4 Where to store figures

In general the figures should reside in the `fig` directory or one of its subdirectories. A `fig` directory exists for each major document, e.g., `TDR/ptdr1/fig/` or `TDR/ctdr/fig/`. Small papers with only a few figures do not require the use of a subdirectory.

Do *not* refer to any figures which reside outside the TDR repository; instead, `cvs add` the file in the `fig` directory and check it in.

By default figures are looked for in the `fig` directory.

If a figure file resides in a subdirectory, e.g., `fig/muon`, of the `fig` directory, then simply prepend the directory name when referring to the figure in the `\includegraphics` command (i.e. `muon/c1` in the above example).

5.4.5 Standard macro for figures produced with ROOT

To maintain a standard look and feel for the figures in the Physics TDRs, a Root macro was contributed by Thomas Speer. Figure 1 shows an example plot made using it. In the TDR repository check out: `TDR/ptdr1/fig/scripts/tdrstyle.C`. To use it:

```
.L tdrstyle.C
setTDRStyle()
```

5.5 Convention for figure and table captions

Figure captions should be located below each figure, as shown in the example above. Table captions, however, should reside *above* the table. For example:

```
\begin{table}[h]
  \begin{center}
    \caption{Table captions are above the table whereas figure
    captions are below.}
    \label{tab:mytab}
    \begin{tabular}{lcc} \hline
       Parameter & Value 1 & Value 2 \\ \hline
       $s$ & 10.0 & 20.0 \\
       $t$ & 20.0 & 30.0 \\
       $u$ & 30.0 & 40.0 \\ \hline
    \end{tabular}
  \end{center}
\end{table}
```

which produces the following:

Table 1: Table captions are above the table whereas figure captions are below.

Parameter	Value 1	Value 2
s	10.0	20.0
t	20.0	30.0
u	30.0	40.0

5.6 Chapters, Sections and Other Sectioning Commands

For all notes and conference reports use the following section heading commands: `\section`, `\subsection`, `\subsubsection`, and `\paragraph`. For Technical Design Reports the top-level sectioning command is `\chapter` followed by all the above sectioning commands.

The PDF bookmarks produced from PdfLaTeX will choke on TeXsymbols, e.g., "2.6 This is a "026E30Fsection" for "2.6 This is a \section" since TeX uses 026E30F to represent the backslash. Use the `\texorpdfstring` macro:

```
\section{Finding the split \texorpdfstring{$A_2$}{A2}}
```

And this is what it should look like:

5.7 This is a `\subsection`

This is some text.

5.7.1 This is a `\subsubsection`

This is some text.

5.7.1.1 This is a `\paragraph` This is some text.

5.8 Cross-references and bibliographic citations

5.8.1 Referring to Sections, Figures, Tables, etc.

LaTeX provides powerful, robust, and scalable facilities for cross-referencing based on symbolic labels. Please use them!

For example, to create symbolic links to a chapter and a section:

```
\chapter{Mass Storage Systems\label{ch:mss}}
\section{Requirements\label{sec:mss-requirements}}
```

Note that the `label` command is contained *within* the curly braces of the appropriate sectioning command so that the value can be resolved correctly. For figures and tables, the `label` command should be similarly enclosed within the associated `caption` command.

To then refer to the chapter and section:

```
The CMS hierarchical mass storage systems, described in
Chapter~\ref{ch:mss} will be of a size unprecedented in
HEP, as described in Section~\ref{sec:mss-requirements}.
```

This will result in output something like:

> The CMS hierarchical mass storage systems, described in Chapter 9 will be of a size unprecedented in HEP, as described in Section 9.1.

Note that the numbers (9 and 9.1) are automatically generated according to the placement of the `label` commands in the overall context of the document. The number of digits (levels) is determined automatically from the level of the sectioning command used (chapter, section, subsection, etc.).

Always – *repeat always* – use symbolic labels (e.g., `sec:mss-requirements`) for references and not hardwired numbers (e.g., `9.1`) as the latter will invariably become wrong very quickly.

5.8.2 Bibliographic References

All bibliographic entries are defined in a BibTeX file (i.e. files with `.bib` extension in the `bib` directory of the TDR (volume) of interest. This enables a standard format to be ensured and helps avoid duplicated entries. Before defining a new bibliographic item, please check in the `.bib` files whether it has already been defined, and if so then use it as it is. When creating new BibTeX entries, the format of the bibliographic entries is mostly self-evident and one can cut-and-paste from an existing entry (well, check that it produces reasonable output) and then change the text.

Keep in mind that for listing authors, the BibTeX implementation uses "Last Name, First Name" (and it automatically abbreviates the first name). Concatenate authors using "and", and instead of writing "*et al.*" use "and others." BibTeX will handle the substitution, although it does not trim the author list automatically. For complicated names, you can place them in braces , but do this sparingly.

We strongly recommend the use of the SPIRES[2] BibTeX labels when such an article can be found there, because a unique label is created and LaTeX can spot multiply-defined references. It also saves you the time of creating the entry yourself. Such an entry looks like:

```
@Article{Agostinelli:2002hh,
     author    = "Agostinelli, S. and others",
     collaboration = "GEANT4",
     title     = "GEANT4: A simulation toolkit",
     journal   = nim,
     volume    = "A506",
     year      = "2003",
     pages     = "250-303",
     SLACcitation  = "%%CITATION = NUIMA,A506,250;%%",
     DOI       = "10.1016/S0168-9002(03)01368-8"
}
```

However, in the above instance and for many other *commonly* cited references, we will use a more conventional name (e.g., GEANT4 instead of Agostinelli:2002hh). So please check the other bibliography files to see if yours is already defined.

In addition, we recommend setting the "DOI" field that was added to the Article BibTeX format in the TDR framework (and is illustrated above). This field represents the Digital Object

[2] `http://www.slac.stanford.edu/spires/hep/`

Identifier for your reference.[3] When you prepend this number with `http://dx.doi.org/`, your browser automatically is directed to the electronic version of the article (provided your institution has paid for this access). Currently you need to manually determine and enter this field after examining the publication.

To refer to an item in the bibliography using its symbolic label in your text, use one of the following forms:

```
Either: the CMS detector is described elsewhere~\cite{CMSTP};
or: the CMS detector is described in reference~\citen{CMSTP}.
```

This will result in output something like:

> Either: the CMS detector is described elsewhere [34]; or: the CMS detector is described in reference 34.

Note the omission of the square brackets, in the second form, when the reference is explicitly (rather than parenthetically) referred to.

The list of references will be placed at the end of the TDR. It is suggested that each group maintain a separate `.bib` file in the `bib` directory for the chapter specific references. Common references for the entire TDR will be kept in a common file (e.g., ptdr1.bib). Common software references will be kept in software.bib.

5.8.3 Web References

Please use the `\href` and `\url` commands to embed links into your document.

Example:
`\url{http://cms.cern.ch/iCMS/}` gives `http://cms.cern.ch/iCMS/`,
`\href{http://cms.cern.ch/iCMS/}{The CMS web site}` gives The CMS web site.

5.9 Glossary

Please add a short entry to `glossary.tex` whenever introducing any new acronym or abbreviation. Even plain English terms with specific technical meaning should be included (e.g., Python).

[3] `http://www.doi.org/`