Hour 23

Error and Warning Messages

Commitment of typographical or logical errors is unavoidable. Most of the errors and warnings in LaTeX can be debugged easily, while some could be a little bit complicated. Some commonly committed errors and subsequent error or warning messages generated by a command-line-based LaTeX compiler are discussed in this Hour.

During compilation of a LaTeX input file, a lot of process-related internal information are displayed, which are not so important particularly for beginners. If the input file, say named '`lsw.tex`', is free from any error that must be debugged, the compilation will be completed with the following two lines of final information:

```
Output written on lsw.dvi (12 pages, 50028 bytes).
Transcript written on lsw.log.
```

In the case of an erroneous file requiring mandatory debugging, however, the compilation will be paused showing the relevant error message immediately after arriving at the first error in the file. In most of the cases, the compilation will be paused with a '`?`' mark after the error message. By the `?` mark, the LaTeX compiler asks the user for an action, which can be responded as follows:

- ▷ If the error is not clear, '`h`' may be entered for help.
- ▷ Allow by pressing the `Enter` key to try to compile without debugging the error.
- ▷ If the error is clear, '`e`' may be entered to lead to the error in the input file. However, care must be taken with this option, since the compiler will open the input file in the terminal if it fails to communicate with the used LaTeX editor. In that case, editing the input file in the terminal, while the same is open in the LaTeX editor also, may even corrupt the input file.
- ▷ Alternatively, terminate the compilation with '`Ctrl+z`', and then edit the input file manually in the used LaTeX editor.

D. Datta, *LaTeX in 24 Hours: A Practical Guide for Scientific Writing*,
DOI 10.1007/978-3-319-47831-9_23

23.1 Error Message

Immediately after detecting the first error that must be debugged, the compilation of the LaTeX input file will either be paused without generating any output, or be terminated by generating output up to the previous page of that containing the first error. Then a user needs to take necessary steps to debug the input file according to the error message displayed by the LaTeX compiler. Some commonly generated error messages and their possible reasons are presented below in alphabetical order.

Error: `*` (no error message, just paused).

✓ `\end{document}` is missing at the end of the LaTeX input file.

Error: `! Display math should end with $$.`

✓ A mathematical symbol or expression is inserted in `$ $` in a math-mode, such as in the `equation` environment or in the `\( \)`-mode. Following the error message, the serial number of the line in which the error occurs will also be shown marking with `l.`, along with the erroneous expression at the end of the line, e.g.,

```
l.349    x+y = $\alpha
```

Error: `! Double subscript.`

✓ Subscript is inserted twice to a single character or symbol, e.g.,

```
l.54    \frac{x_i_j}{y}
```

Error: `! Double superscript.`

✓ Superscript is inserted twice to a single character or symbol, e.g.,

```
l.54    \frac{x}{y^t^2}
```

Error: `! LaTeX Error: Environment --- undefined.`

✓ A non-existing or mis-spelled environment is used, e.g.,

```
l.54    \begin{enumerating}
```

✓ Environment is correct, but its supporting package is not loaded. LaTeX will not say anything about the package, but the environment will be shown undefined, e.g.,

```
l.211    \begin{wrapfigure}
```

In the above example, `wrapfigure` is a correct environment, but the error is shown since its supporting package `wrapfig` is not loaded in the preamble.

Error: `! Extra }, or forgotten \endgroup.`

✓ Something is ended by '}' without starting it with '{', e.g.,

```
l.7    ... as follows for a file named \tt lsw.tex}
```

✓ An environment is used without starting it with `\begin{}`. Following the error message, the ending line of the environment will be shown.

Error: `! Extra alignment tab has been changed to \cr.`

✓ Extra entries with `&` are inserted in a row of a column-based environment, like `array`, `eqnarray`, `tabular`, or `tabularx`. This may also happen if two rows are not separated by `\\`, or if a column is generated with a wrong option, e.g., use of `X` in the `tabular` environment (`X` is permitted in the `tabularx` environment only). The erroneous row or the ending line of the environment will be shown after the error message.

✓ Non-existing column number is used in **\cline{}** in the **tabular** or **tabularx** environment.

Error: `! Extra \right.`

✓ Delimiter height adjustment command **\right** is used without preceding by its complementary command **\left**, e.g.,

```
l.35    \frac{x}{y} \right)
```

Error: `! File ended while scanning use of \---.`

✓ See `Runaway argument?`

Error: `! Illegal parameter number in definition of \---.`

✓ More number of arguments is used in **\newcommand{}{}** than defined, e.g.,

```
l.48    \newcommand{\sq}[1]{#1^{#2}}
```

Error: `! Illegal unit of measure (pt inserted).`

✓ Unrecognized length unit is used, e.g.,

```
l.24    \begin{minipage}[t]{5.0cn}
```

Error: `! Improper \prevdepth.`

✓ Previous environment is not closed by **\end{}**. The ending line of the next environment will be shown after the error message.

Error: `! LaTeX Error: \begin{nenv} on input line -- ended by \end{document}.`

✓ Environment 'nenv' is opened with **\begin{**nenv**}** in line `--`, but the compilation reached the end of the input file, i.e., **\end{document}** command, before it is closed with **\end{**nenv**}**.

Error: `! LaTeX Error: \begin{nenvs} on input line -- ended by \end{nenvf}.`

✓ Environment `nenvs` is opened with **\begin{**`nenvs`**}** in line `--`, but forgotten to close it by **\end{**`nenvs`**}** (or mistakenly closed by **\end{**`nenvf`**}**). The ending line of the next environment (or the erroneous closing line) will be shown after the error message.

Error: `! LaTeX Error: Can be used only in preamble.`

✓ Something is inserted after **\begin{document}**, which is permitted in the preamble only, e.g., **\usepackage{}**.

Error: `! LaTeX Error: \caption outside float.`

✓ The **\caption{}** command is inserted outside a float environment, like **table** or **figure**.

Error: `! LaTeX Error: Command \--- already defined.`

✓ Attempt is made to define an existing command as a new command, which is not permitted, e.g.,

```
l.29    \newcommand{\alpha}{A Greek letter}
```

Error: `! LaTeX Error: File '---' not found.`

✓ The figure file '`---`' inserted through **\epsfig{}** or **includegraphics[]{}** is either not available or not in proper format, e.g.,

```
l.246    ...ludegraphics{file=pot.eps,width=2cm}
```

In the above example, the figure file might be available, but it is read as '`file=pot.eps,width=2cm`', not as '`pot.eps`'. This is due to the fact that the figure is inserted using the syntax of **\epsfig{}** in **\includegraphics[]{}**.

Error: `! LaTeX Error: File '---.cls' not found.`

✓ Either an unavailable or a mis-spelled document-class file is inserted in **\documentclass{}** (`cls` extension means a document-class file).

Error: `! LaTeX Error: File '---.sty' not found.`

✓ Either an unavailable or a mis-spelled package file is loaded through **\usepackage{}** in the preamble (`sty` extension means a style file, which is used as a package).

Error: `LaTeX Error:Lonely \item--perhaps a missing list environment.`

✓ The **\item** command is inserted outside a listing environment, like **enumerate**, **itemize**, or **description**. The erroneous line, or the next line if the **\item** command has no content, will be shown after the error message.

Error: `! LaTeX Error: Missing \begin{document}`

✓ Either some contents of a document are inserted before **\begin{document}** or **\begin{document}** is missing.

Error: `! LaTeX Error: No \title given.`

✓ The **\maketitle** command is not preceded by **\title{}**, or it is missing.

Error: `! LaTeX Error: Not in outer par mode.`

✓ Two non-permitted environments are nested, like **figure** in **minipage** or **table**. Following the error message, the first line in the inner environment will be shown.

Error: `! LaTeX Error: Option clash for package ---.`

✓ The same package is loaded multiple times using **\usepackage[]{}** with different options. The next line after the last erroneous package loading will be shown after the error message.

Error: `! LaTeX Error: Something's wrong--perhaps a missing \item.`

✓ No **\item** is inserted in a listing environment, like **enumerate**, **itemize**, or **description**. The ending line of the environment will be shown after the error message.

✓ A non-permitted environment is nested inside a secondary environment, like **verbatim** in **tabular**.

Error: `! LaTeX Error: There's no line here to end`

✓ The **\vspace{}** command is preceded by a blank line, or followed by a new line or a line break command, like \\, **\newline** or **\linebreak**. Following the error message, the next line that of **\vspace{}** will be shown.

Error: `! LaTeX Error: Too deeply nested.`

✓ Excess number of environments are nested. The starting line of the first excess nested environment will be shown after the error message.

Error: `! LaTeX Error: Too many columns in eqnarray environment.`

✓ More than three columns, i.e., more than two **&**, are used in a single row of the **eqnarray** environment. The ending line of the environment will be shown after the error message.

Error: `! LaTeX Error: Undefined tab position.`

✓ A tab is used in a row of the tabbing environment, which is not defined earlier, e.g.,

```
l.253    Volume (V)    \> = bdh    \> High
```

Error: `! LaTeX Error: Unknown option ‘---’ for package ‘---’.`

✓ An unknown option is inserted in \usepackage[]{} while loading a particular package. The erroneous line, shown after the error message, may be somewhat strange, e.g.,

```
l.40    \RequirePackage
```

Error: `! LaTeX Error: \verb illegal in command argument.`

✓ The \verb" " command is inserted in the argument of another command, e.g.,

```
l.54    \textcolor{red}{\verb"This is in red color"}
```

Error: `! Misplaced alignment tab character &.`

✓ Alignment tab & is inserted in a wrong place (it is used in column-based environments, like array, eqnarray, tabular, or tabularx).

Error: `! Misplaced \noalign.`

✓ Previous row prior to a \hline command in the tabular or tabularx environment is not ended by the line-break command \\. The ending line of the environment will be shown after the error message.

Error: `! Missing } inserted.`

✓ Something is started with ‘{’ in a paragraph within an environment other than document, but it is not closed by ‘}’. The ending line of the environment will be shown after the error message. Since the exact location of the error is not shown, it may take some time to manually locate the error inside the environment.

Error: `! Missing $ inserted.`

✓ Either the symbol or the environment shown after the error message is to be inserted in a math-mode (say in $ $, \(\), \[\], or equation environment), e.g.,

```
l.304    Greek letter \alpha
```

or,

```
l.364    \begin{array}{ll}
```

Error: `! Missing control sequence inserted.`

✓ The first argument of \newcommand{}{} or \renewcommand{}{} is not a command, i.e., not preceded by a \, e.g.,

```
l.42    \newcommand{sq}[2]{#1^{#2}}
```

Error: `! Missing \endcsname inserted.`

✓ The name of an environment is preceded by a \ in \begin{} or \end{}, e.g.,

```
l.24    \begin{\tabular}
```

✓ The first argument of \newcommand{}{} is missing. The next line of the erroneous one will be shown after the error message.

Error: `! Missing \endgroup inserted.`

✓ In a group of nested environments, an inner environment is not ended by \end{}, e.g., while using tabular in table and closing table by \end{table} without closing tabular by \end{tabular}.

Error: `! Missing number, treated as zero.`

✓ A numerical-valued argument of a command or environment is missing, or a nonnumeric value is assigned to it. The erroneous line, or the next line if the argument was the last term of the erroneous line, will be shown after the error message, e.g.,

```
l.34    \hspace c
```

or,

```
l.25    \begin{tabularx}{}{|l|c|c|}
```

Error: `! Missing \right. inserted.`

✓ Delimiter height adjustment command **\left** is not closed by the complementary command **\right**. Following the error message, the erroneous line or the ending line of the environment in which the error occurs will be shown, e.g.,

```
l.116    \[f\left(\frac{x+1}{2}+2)\]
```

Error: `! Package varioref Error: \vref at page boundary -- - -- (may loop).`

✓ The effect of the **\vref{}** command falls in the last line of a page of the output file, which will be displayed as, e.g.,

```
l.630    ...ed in \S\vref{sec:colwidth}
```

This error is something like a warning only. Hence, the compilation may be allowed to continue without bothering about it.

Error: `! Package xcolor Error: Undefined color '---'.`

✓ The inserted color is not defined through **\definecolor{}{}{}** supported by the **color** package. Following the error message, the erroneous line will be shown, e.g.,

```
l.263    ...r{LightBlue}{This is in light blue color}
```

Error: `! Paragraph ended before \--- was complete.`

✓ See `Runaway argument?`

Error: `Runaway argument?`

✓ Something is started with '**{**', but not closed by '**}**'. The line number shown below the error message may point to a broader location of the error. However, the exact location of '**{**' can be found easily by checking the immediate next line of the error message and then it can be closed by '**}**' in the proper location, e.g.,

```
{enumerate \section {Unnumbered listing} \ETC.
```

or,

```
Column {\bf X is used in tabular environment \ETC.
```

The above two examples indicate that '**{**' prior to '`enumerate`' and '`\bf`', respectively, are not closed by '**}**'.

Error: `! TeX capacity exceeded, sorry [input stack size=---].`

✓ The first argument of **\renewcommand{}{}** is empty, or a fragile command is inserted as an argument of another command, say **\footnote{}** in **\section{}**. Compilation will not be paused in any of the cases, but the output file will be generated up to the previous page of that containing such error. Following the error message, the erroneous line will be shown, e.g.,

```
l.32    \renewcommand{}
```

or,

```
l.213    ...{\footnote{} is a fragile command.}
```

Error: `! Too many }'s`

✓ See `! Extra }, or forgotten \endgroup.`

Error: `! Undefined control sequence.`

✓ An unavailable or mis-spelled command is inserted, e.g.,

```
l.60  ...ommand is inserted. The Greek letter \alfa
```

✓ Command is correct, but its supporting package is not loaded. LATEX will not say anything about the package, but the command will be shown as an undefined control sequence, e.g.,

```
l.78    This is in \textcolor
```

In the above example, **\textcolor** is a correct command, but the error is shown since its supporting package **color** is not loaded in the preamble.

23.2 Warning Message

In the case of not having any error that must be debugged, the compilation of a LATEX input file, say named '`lsw.tex`', by the `latex` command will generate the output with the following two lines of final information:

```
Output written on lsw.dvi (12 pages, 50028 bytes).
Transcript written on lsw.log.
```

If the input file contains any labeling and cross-referencing, i.e., labeling numbered items with **\label{}** and referring them with **\ref{}**, or citing a bibliographic item with **\cite{}**, two or three lines of warning messages may also be seen somewhere above the two lines of final information mentioned above. Such warning messages will be like those as shown below:

```
... Warning: There were undefined citations.
LaTeX Warning: There were undefined references.
LaTeX Warning: There were multiply-defined labels.
```

The first two warning messages may disappear upon completing the compilation of the input file as discussed in §14.3, 15.4 or 16.2.3. However, it is a matter of concern if any of such warning messages still remains.

Warning: `... Warning: There were undefined citations.`

✓ This is a warning from the bibliographic reference generator, e.g., '`Package natbib Warning`' if bibliographic references are generated through the **natbib** package. The warning message means that some citation keys used in the **\cite{}** command are unknown, may be undefined or mis-spelled in the bibliographic list. Checking the list of information generated by the LaTeX compiler, such errors can be found very easily, e.g.,

```
Package natbib Warning: Citation ‘lko’
   on page 8 undefined on input line 271.
```

✓ Alternatively, the output file (i.e., the `.dvi` file) may be checked for ‘`?`’ sign, where each of such errors will be marked by a ‘`?`’ sign. Once the location of an error is found in the `.dvi` file, the required correction can be made in the input file.

Warning: `LaTeX Warning: There were undefined references.`

✓ This warning message says that some reference keys used in the **\ref{}** command (or other similar commands, such as **\vref{}**, **\pageref{}**, **\vpageref{}** and **\eqref{}**) are either mis-spelled or not defined through the **\label{}** command. As in above, these errors can also be found very easily by checking the list of information generated by the LaTeX compiler, e.g.,

```
LaTeX Warning: Reference ‘pichart’
  on page 8 undefined on input line 269.
```

✓ Alternatively, the output file (i.e., the `.dvi` file) may be checked for ‘`??`’ sign, where each of such errors will be marked by ‘`??`’ sign. Once the location of an error is found in the `.dvi` file, the required correction can be made in the input file.

Warning: `LaTeX Warning: There were multiply-defined labels.`

✓ This is the warning message if the same reference key is defined multiple times through the **\label{}** command, i.e., multiple items are labeled by the same reference key. Unlike in the above two warnings, LaTeX will not show the location of the error in this case. However, checking the list of information generated by the LaTeX compiler, it can be found which reference key is defined multiple times, e.g.,

```
LaTeX Warning: Label ‘vcomp’ multiply defined.
```

Once the multiply defined reference key is found, the input file can be checked for the locations where it is defined.

23.3 Error Without Any Message

There might be some visual errors, which LaTeX cannot catch and hence the output is generated without any error or warning message. Some of such cases are addressed below:

▷ A table may be printed in a wrong location if wrong syntax is used in **\begin{table}[ ]** for vertical alignment, say **l**, **c** or **r** in place of **h**, **b**, **t** or **H**.

- ▷ Strange sectioning numbers will be generated if various sectioning commands are not used in a proper order, e.g., **\subsection{}** without preceded by **\section{}**, or **\subsubsection{}** without preceded by **\subsection{}**.
- ▷ If a font type command shown in Table 2.1 on page 10 is used for changing fonts of a particular portion, but forgotten to put it in **{}** or to change it by another applicable font type command, the fonts of the entire remaining contents of a document will be changed. For example, **\tt** not in **{}**, or **\scshape** is not followed by **\upshape**.

23.4 Tips for Debugging

It may be difficult to debug a LATEX input file if it contains a huge number of errors. Sometimes a single error committed in one place may also cause many more errors in the remaining portion of a document, thus making the debugging of the LATEX input file complicated and cumbersome. Hence, it would be a good practice to proceed as follows:

1. Periodically compile the input file, i.e., without waiting to finish the typing of the entire input file, compile it upon completion of the typing of each segment, say a paragraph or a table.

2. If it becomes difficult to identify or debug an error, first comment the entire doubtful segment with % sign. Then compile the input file repeatedly, each time uncommenting a line or a subsegment in order, until the erroneous line (or subsegment) is detected for debugging.