Instructions for Authors

Phillip Helbig

2025 November 29

Abstract

This document should be sufficient to enable authors to write items for *The Observatory* with minimal effort, in a form which allows for efficient processing by the Editors. While we aim to preserve backward compatibility, we can't guarantee that. In any case, make sure that you follow the latest version of this document (see the date above), available at `http://www.obsmag.org/templates/instructions.pdf`.

1 Introduction

Obviously, since the first issue in 1877, both the format of submissions and the methods of production for the *The Observatory* have changed several times. All the same, the appearance of the final product hasn't changed that much. Recent changes are becoming on-line-only with the 2025 February issue and producing the *Magazine* in LaTeX as of the 2026 February issue. Recent input formats include `.tex`, `.txt`, `.doc`, `.docx`, and `.odt`, as well as scans of paper printed with a typewriter. While we will continue to support, at least for a while, formats which have been accepted in the last few years (though hopefully no more typewriter submissions), it would be easier for us *and for authors* if all future submissions are in LaTeX. Such submissions should also be extremely easy even for authors with no experience in LaTeX. Those with more experience should be able to put that to good use.

2 Concept

TeX is a typesetting language. LaTeX is a set of TeX macros, the idea being that the author concentrates on content rather than presentation. So, the author types `\fract{1}{2}` rather than commands to position one number above another, separated by a horizontal line of given length and thickness, and with commands to position that line at a certain distance from the two numbers and at a certain height above the baseline of the text. Similarly, `\section` indicates a new section, a blank line a new paragraph, and so on. The author shouldn't have to worry about how such elements are formatted.

Of course, someone has to take care of the formatting, and that is done by the person who writes the corresponding *document class* (`.cls`) for the main formatting and/or a package (`.sty`) to provide additional style elements. In practice, such packages are sometimes somewhat incomplete and the author is expected to handle some of the formatting, following some sort of style guide. Our goal is to have everything formatted automatically which can be formatted automatically. Of course, that can be done only for those parts of the text which are indicated to be special in some way, such as a section title, caption, or heading. In such cases, the author should just write the corresponding text; there is no need to try to emulate the font shape or size used in the *Magazine*, nor such things as text width, justification, and so on. The only formatting the author needs to do is *within* some text if necessary, *e.g.*, indicating that something is to be set as maths rather than normal text, or to indicate special formatting to follow the style of the *Magazine*, *e.g.*, italicizing the names of space missions. In the former case, standard LaTeX commands should be used. In the latter case, provided macros should be used.

3 Minimal version

Choose the appropriate template from `http://www.obsmag.org/templates/`: `article.tex`, `thesisabstract.tex`, *etc.* and make a copy of that (the copy will be referred to as the 'item file' below); you can call it whatever you want. Each template corresponds to a LaTeX 'environment' and consists of three parts:

- the environment with empty fields (`{}`) to be filled in by the author
- the same with keywords naming the fields (commented out with `%`)
- an example (commented out)

Simply fill in the fields in the item file with the corresponding information, which should be obvious. *There is no need to format anything!*

Strictly speaking, that is sufficient. Just email the item file to the Editors at `submissions@obs.mag.org`. (It's probably safer to email it as an attachment, but since it is just a text file, it could be included in the body of the email (which need not contain anything else), at least if it contains only printable ASCII characters and the lines are not too long. (There are other reasons to avoid long lines; see below.))

While we will continue to accept legacy formats (see above), this minimal version of the submission process should be easy enough even for those with no LaTeX experience and/or without access to a LaTeX installation. But if it is more convenient for you, you can send your contribution in one of those formats and/or as a plain text file. But the minimal version makes it easy for the Editors, and is probably less work than any legacy format.

4 Basic steps with LaTeX

In practice, many authors would like to process their contribution with LaTeX. That also provides a check that it is formatted correctly. The resulting PDF file could also be included in the submission, though strictly speaking that shouldn't be necessary. (If there are technical problems, we would request it so that the author's intention is clear.)

Asking someone familiar with LaTeX (and if used, BibTeX and the NATBIB package) for help should be enough to get a submission ready as described in this section and the next section. I have intentionally included no links to LaTeX documentation, as there is much information easily available on the World Wide Web, including descriptions of excellent traditional paper-book manuals (buying them is a very good investment) and various descriptions, tutorials, forums, suggestions, and so on, at a wide range of levels of difficult and in several languages. There is something for everyone.

Formatting is handled by document class `obsmag.cls`. Put that somewhere where LaTeX can find it. (The current working directory should always work; those familiar with LaTeX can install it in the appropriate place, which would also make it available to all users on the machine.) There are two files you need to edit. One uses the same template for all contributions. For that, make a copy of our template `contribution.tex` (`http://www.obsmag.org/templates/contribution.tex`) (the copy will be referred to as the 'contribution file' below). The other is the 'item file' mentioned above. You can name those copies whatever you like.

Make the following changes to the contribution file

- Fill in the appropriate fields for the `\issueinfo{}{}{}{}` command, *e.g.*, `\issueinfo{146}{2026}{February}{1310}` (it will work with empty arguments but the corresponding header information will be missing).
- Replace `item.tex` with the name of your item file in the argument of the `\input{item.tex}` command (not necessary if it is called `item.tex`).
- Process the file with LaTeX.

Send the item file and perhaps the PDF file to `submissions@obsmag.org` (see above).

For articles, the author's name appears on odd-numbered pages. In the case of an extremely long name and/or more than one author, the corresponding text (between the year and month and the page number in the header) might be too long. In that case, uncomment the line which begins with the command `\renewcommand*{\shortauthorlist}` and include a short author list as the argument. (The maximum length depends on the letters *etc.* but is approximately 55 characters.) Also a good idea if, in exceptional cases, there is a footnote in the author list. (Note that footnotes are not needed to specify afffiliations in the case of more than one affiliation; see the corresponding template.)

5 References

Some items in the *Magazine* sometimes contain citations and references. *The Observatory* uses superscripted numbers for citations. The corresponding references appear in the order cited in the reference list (repeat citations re-use the previous number), *e.g.*,

(17) M. Breger, *ASP Conference Series*, **210**, 3, 2000.

as produced by code similar to

```
(17) M.~Breger, \textit{ASP Conference Series}, \textbf{210}, 3, 2000.
```

(See the inside back cover for examples, *e.g.*, `http://www.obsmag.org/issues/published/2025/10/1308_inside_back_cover.pdf`.

5.1 Using LaTeX for citations and references

While it is possible to write literal superscripts in LaTeX (*e.g.*, `$^{1}$` or, more verbose but more correct, `\textsuperscript{1}`), and construct the reference list to match by hand, that presents several problems:

- There is no guarantee that the citations and references correspond.
- The references must be formatted by hand.
- Manual renumbering is necessary if citations are added or deleted.
- Without additional work by the editors, citations are not linked to the reference list *via* HTML links in the text.

We thus strongly suggest that authors use the standard LaTeX mechanism for citations and references. The minimal version of that is to have commands such as `\cite{key-2}` (where '`key-2`' is a more or less arbitrary text) in the main text and after the main text a bibliography environment with the corresponding entries, *e.g.*,

```
\beginthebibliography{10}
\bibitem{key-2} M. Breger, \textitASP Conference Series, \textbf210,3, 2000.
\bibitem{key-13} E. Poretti, \textitA&A, \textbf409, 1031, 2003.
\end{thebibliography}
```

(the argument, here '10', simply corresponds to the length of the number of the last entry, in this case two digits). Authors could either have such an environment at the end of the main item file (preferably *after* the `\end` command ending the *Magazine* environment (article, book review, *etc.*) or have it as a separate file and uncomment the line `%\input(bblname.tex)` in the contribution file, replacing `bblname` with the name of the corresponding file. (Convention is to have the extension `.tex`; in that case, `.tex` is not necessary in the `\input` command, but there is no harm in leaving it in and it is perhaps clearer. If another extension is used, then the full file name, including extension, must appear in the arbument of the `\input` command.) The file must be processed twice with LaTeX.

5.2 BibTEX

Even better would be to use BibTEX, which is part of every standard LaTeX distribution. Citations are as above, but the reference list is constructed automatically from a `.bib` file. To use BibTEX, uncomment the line beginning with `\bibliographybibname` in the contribution file (and make sure that the line beginning with `\input(bblname.tex)` is commented out), replacing `bibname` with the name of your `.bib` file. Then process the contribution file once with LaTeX, once with BibTEX, then, as above, twice with LaTeX.

A `.bib` file consists of entries of different type containing the citation key and various fields for the corresponding information, *e.g.*,

```
@ARTICLE      { Rstabell68a ,
                AUTHOR       = "Rolf Stabell",
                TITLE        = "Cosmological models with pressure",
                JOURNAL      = MNRAS,
                YEAR         = "1968",
                  VOLUME       = "128",
                  NUMBER       = "3",
                  PAGES        = "313",
                  MONTH        = feb,
                  NOTE         = "",
                    ANNOTE       = "",
                    KEY          = "",
                      DOI          = "10.1093/mnras/138.3.313"
              }

DOI works but is "wrong"
@ARTICLE      { RStabellSRefsdal66a ,
                AUTHOR       = "Rolf Stabell and Sjur Refsdal",
                TITLE        = "Classification of general relativistic
                                world models",
                JOURNAL      = MNRAS,
                YEAR         = "1966",
                  VOLUME       = "132",
                  NUMBER       = "3",
                  PAGES        = "379--388",
                  MONTH        = "apr",
                  NOTE         = "",
                    ANNOTE       = "",
                    KEY          = "",
                      DOI          = "10.1093/mnras/132.2.379"
              }
```

Comments on the above:

- The key, *e.g.*, `Rstabell68a`, is essentially arbitrary.

- Fields can also be delimited *via* `{}`, though since `{}` is also used for many other purposes (such as to surround all or part of a field which should be processed as is and not subjected to formatting such as converting text to lowercase, particularly useful for uppercase acronyms, German proper nouns, and so on).
- Only the fields needed by the *Magazine* need to be included, *e.g.*, for articles author, journal, (first) page number, and year; other fields can be present but are ignored if not needed. It is recommended to include the `DOI` field since we hope to use `DOI`s in the future
- Fields can be empty.
- Text outside the group of the corresponding item is ignored and can be used for comments.
- Multiple authors separated by `and`; the bibliography style will automatically use only initials if desired.
- All necessary formatting is done by the bibliography style.
- Values of keywords not in `""` are macros defined either previously in the same `.bib` file and/or in another `.bib` file earlier in the argument of `bibliography` (multiple files are separated by commas). Macros can be defined more than once, in which case previous definitions are overwritten and only the last used. One can thus have a master list of macros—particularly useful for journal names and abbreviations—in one file, then overwrite them if necessary if the journal requires something else, preferably in a journal-specific `.bib` file containing such macros. That allows the `.bib` to be used for any journal without having to make journal-specific changes. (It is also possible to (re)define a LaTeXcommand for journal names. The advantage is that defining it *via* BibTeX (*e.g.*, `@STRING{PNASUSA = {Proc. Natl. Acad. Sci. USA}}`, allows it to be tested with just BibTeX which requires very little processing time (as it reads only the `.aux` file), with no need to repeat LaTeX processing for such tests. Note that there are pre-defined macros for months, *e.g.*, `jan`, `apr`.
- Items can be concatenated *via* `#`, *e.g.*, `jul # "10"`.

To use BibTeX put `obsmag.sty` (`http://www.obsmag.org/templates/obsmag.sty` somewhere where LaTeX can find it.

The most useful thing I have done in my life in order to avoid wasting time is learning to type properly, closely followed by learning to use LaTeX and BibTeX.

5.3 The natbib package

The natbib package was designed for NATural-science BIBliographies. While it really comes into its own when used with author/year-style citations, it also

offers many features which are valuable with numerical citations, such as automatically compressing multiple references in one citation, and it's useful `\citet` and `\citep` functionality always works. For example, `It has been shown by \citet{Jones}` produces something like 'It has been shown by Jones (1985)', while `Previous work\citep{Jones}` leads to something like 'Previous work (Jones 1985). In numerical style, the former (note that the exact same code is used) would produce 'It has been shown by Jones[18]' and the latter 'Previous work[18]'. It is definitely worth using, especially if one also has occasion to use author/year-style citations.

6 Which LaTeX?

For more than 30 years, I personally used the tried-and-true command sequence `latex`, `dvips`, more recently supplemented by `ps2pdf`. That was sufficient for my purposes as my own figures are produced as PostScript files via a FORTRAN program. And in the old days that was all there was. It also made it easier in those rare cases when editing PostScript files by hand was necessary. If that's sufficient for you, fine. However, the traditional `latex` command can include only PostScript figures. Such figures can also be quite large.

At least for the foreseeable future, the *Magazine* is produced via `pdflatex`. That has several advantages:

- Other types of files, such as `jpg` and `png`, can be inlcuded.
- `.pdf` files can be included. Some figures these days might exist only as PDF, rather than having been converted from PostScript. They are also usually much smaller in terms of file size, making it easier to send them via email, less disc space is used, etc.
- Only one command is needed instead of three.

The only disadvantage is that PostScript figures have to be converted to PDF, but doing that once usually takes less time than that required to read the big PostScript file during processing and to execute the additional commands. (Note that some installations might be set up so that `pdflatex` calls some conversion on the fly, allowing PostScript files to be included. But it is usually more efficient to convert the files and alter the corresponding commands to include them. Note that it is usually not necessary to include the file extension in the command (usually `\includegraphics`) to include the file, though it can be clearer.)

So the preferred format for figures such as plots is PDF rather than PostScript, though the latter are acceptable; JPEG is good for images; PNG for most other things.

Note that in practice it is usually the case that the `latex` and `pdflatex` commands run the same executable file, so if your LaTeX installation is reasonably new, you can probably just type `pdflatex` in order to use it.

Since `pdflatex` is sufficient for our purposes and widely available, we plan to stick with that for now. Should we move to a different engine in the future, then probably `lualatex` rather than `xelatex`. There are still a few things which `pdflatex` can do but `lualatex` cannot, and we are more likely to run into them than to need features supported only by `lualatex`. At least for now, probably all three engines will work with all submissions for the *Magazine*, and even the traditional `latex` will in many cases.

7 Macros

LaTeX macros are useful for many reasons: they can save keystrokes by being shorter than their expansion, they ensure consistency (provided that one always uses the macro and doesn't type its expansion literally), they prevent almost all typos for the corresponding text (a typo in a macro name is unlikely to result in another valid macro), they provide context. With regard to the last item, while macros specifying space missions, book titles, and foreign words might all ultimately result in the text being set in italics, using the corresponding macro provides additional context. Such macros also allow one to write the same LaTeX code for different purposes; it is necessary only to redefine the macro appropriately. Note that `\(re)newcommand*` is preferred to `\(re)newcommand` if there is never a line break (*i.e.*, start of new paragraph) within the argument. In the following, I use `\(re)newcommand` generically for both.

`\newcommand` fails if the command is already defined. (That is one reason one should use it as opposed to the TeX command `\(long)def`, which would silently overwrite the definition.) `\renewcommand` fails if the command is not defined, which is also desired and useful behaviour. A solution is `\providecommand`, which uses the existing definition if there is one but defines it as requested if not. Of course, there is no guarantee that the two definitions be identical, or even supposed to represent the same expansion. `\renewcommand` is useful to redefine one's own commands, *e.g.*, to be able to write the same text wherever it ends up being submitted, matching the redefinition to the journal style. (`\newcommand` would also work, but since most commands—especially most of one's own commands—are not journal specific, it is better to have a master list with `\newcommand` and redefine those need with `\newcommand`.) If one of one's own commands it later to be found to be defined by some LaTeX class or package, then if the meaning is the same one can replace one's own `\newcommand` with `\providecommand`. If the meaning is not the same, then to avoid confusion one should rename one's own `\newcommand`.

Most authors who have been using LaTeX for a while have their own suite of commands. A scheme for retaining those while at the same time using them together with LaTeX classes and packages which define their own commands will be presented in a future version of this document. For now, we offer `obscommands.tex`. If unwanted, it can be commented out at the beginning of the contribution file. However, we encourage authors to make use of the commands in it, taking the steps described above in case of any definition conflicts.

The commands have rather obvious names describing their functionality. Please don't define any other command with the same name. Some of the commands ensure the correct house style for the journal (and can of course be used with the same names but different definitions elsewhere, which has all the advantages described above). The list will be expanded in the future (please check that you have the latest version from `http://www.obsmag.org/templates/obscommands.tex`.)

Note that macros (always introduced by a backslash (\)) consist of either one or more letters or a single non-letter. In the former case, they are terminated by the next non-letter. *If that non-letter is a space then it essentially disappears*, so if a space is needed after the command it must be provided explicitly, *e.g.*, with `\ ` (backslash followed by a space) for a normal intra-word space (or `~` for a non-breaking space. When TEX was originally developed, more commands were used in math mode than outside it. In math mode, it makes sense to use spacing to separate commands (several of which might follow one another with nothing in between) with spaces for legibility but at the same time have the spaces ignored when the code is processed, the spacing being determined by the rules for setting mathematical text. (FORTRAN adopted a similar policy which, although the original motivation was different, was used for similar purposes.) It would probably break little if any existing code if TEX were changed so that spaces ending commands are not ignored outside of math mode, but that probably won't happen. (There is the XSPACE package to provide that functionality with personal macros, but it doesn't affect built-in macros.) If a space is needed, the usual method is to type `\ ` (backslash followed by a space) after the command, or `{}` (an empty group, which can always be used to end the command even if it is not needed, which might appeal to consistency curmudgeons.

8 Packages

The huge variety of LATEX packages have vastly improved LATEX which, like the TEX it is built on, was already very good. (Their quality is obvious from the fact that, 40 or 50 years after they were developed—and originally intended only as personal shortcuts for the respective authors—, they are still state-of-the-art.) If you find yourself trying to do something complicated in LATEX, there is probably a package for it. Spending a few minutes to learn about the package is probably more efficient than rolling your own, and will be compatible with the code from other users. While many packages offer a wide range of options, many probably use them for the most common tasks, which are usually easy to find in the documentation.

`obsmag.cls` itself loads several packages, as it needs them itself; feel free to use the features provided by them. Of course, they should be used only for formatting within the main text, not for changing the layout (which is what `obsmag.cls` often uses them for). Since packages need to be loaded before the `\begin{document}` command, they need to be loaded in the contribution file, not the item file. In such a case, submit the contribution file as well, or include

a note saying which packages you used. In the future, we will include commonly used packages in `obsmag.cls` and encourage all users to use them.

9 Finding the right method

Depending on your experience with LaTeX and the features you need, choose one of the following scenarios. The list is ordered by increasing familiarity with LaTeX in general, you should use the highest item on the list which meets your needs, though we discourage the first two options.

1. Submit your contribution in a legacy format such as `.doc`, `.docx`, or `.odt`. (That is reasonably easy for us to deal with for simple text, but in such cases option 2 or, better, 3 is preferred. More complicated texts should be sumitted in LaTeXif at all possible.)

2. Submit your contribution as a simple plain-text file.

3. Go with the minimal version above.

4. Go with the minimal version in the sense that no additional commands are needed, but use LaTeX commands in the text where appropriate, *e.g.*, $v = HD$ for equations.

5. As above, but process the file with LaTeX and perhaps include the resulting PDF file with your submission.

6. Use `obscommands.tex`.

7. Use additional packages.

10 Common LaTeX mistakes

- Font-changing commands such as `\bfseries`, `\itseries`, `\normalfont` are *not* commands with arguments, but rather change the appearance of the text until it is changed again, or until the corresponding group ends. Thus, write `{\itshape text}`, not `\itshape{text}`. The second 'works' if that group is immediately ended by another one, *e.g.*, `\caption{Some normal text \itshape{some italic text}}`. It will *not* have the expected effect in this case: `\caption{Some normal text \itshape{some italic text} some normal text.}` (While it might be useful to change the type of text for *part of* a caption, please *do not* use such code to change the standard caption format; as described above, that is done by the `obsmag.cls` class.)

- Don't use `\emph{}` as shorthand for `texttt` or `\itshape`. If the surrounding text is in italics, then that will revert to the normal font. That is fine if it really is used for emphasis, but often italicized items (such

as book titles) remain italicized if the surrounding text is in italics (such as a figure caption); that is also the style for the *Magazine*. Best is to use `\emph` for emphasis but instead of `\textit` the corresponding macro (e.g.,`\device` for satellites, space missions, telescopes, instruments such as spectrographs, etc.)

11 Doing it better

- Commands such as `\emph`, `\texftbf`, and `\textic` *are* commands which take an argument; they should be used for short pieces of text; otherwise it is better to use the commands which change the appearance of text within a group.

- Don't use `$-$` as shorthand for some sort of dash in text mode; it is a minus sign and a minus sign only. LaTeX supplies three different lengths of dashes, `-`, `--`, and `---`, and those are sufficient. What does differ from journal to journal is which type is used in what cases, whether or not they are surrounded by spaces, whether they are allowed at the beginning and/or end of a line, and so on. Easy solution: use our 'dash' macros with no explicit space after them, *e.g.*, `Herbig\ndash Haro` or `Herbig\ndash{}Haro`. The 'endash' is quite common, used for joining names (`\ndash`), ranges (`5\rdash7`), to indicate mathematical relationships (`magnitude\jdash redshift relation`), and so on. Even if they all resolve to `--` in the *Magazine*, using them not only provides context, but also allows the same code to be used elsewhere, redefining the commands if necessary. Similarly, several commands resolve to `---`. `-` is used only for hyphenation, which is automatic, and for joining two-word adjectives, which should be written explicitly (because there is absolutely no justification ever for not joining the components of a two-word adjective except in cases where both are capitalized).

- Similarly, in math mode, `$--$` is two minus signs. For things like ranges, use something like `$3\times10^{3}$--$8\times10^{4}$` or, better, 3×10^3--8×10^4.

- `[htbp]` doesn't tell LaTeX where it *must* put floats (tables and figures), but where it *can* put them. If not specified, the default is `tbp`. If, *e.g.*, `[ht]` is specified, then LaTeXstill might put it on another page if there are reasons to do so. Sensible for the *Magazine* is `[htbp]`.

- If a superscript is needed outside of math mode, `\textsuperscript{text}` is preferred to `$^{text}$`. (If you use it often, define something like `\newcommand*\tsp[1]\textsuperscript#1` then write `\tsp{text}`. or whatever.)

- Avoid long lines in the `.tex` file. Since error messages contain the line number, they are easier to find if the line is short. 72 characters is a

sensible maximum length in almost all cases.

12 When formatting *is* necessary for style

Although the goal is that the author thinks about only content, not about style, sometimes style commands are necessary because it is difficult to determine automatically the function of the text. For example, text set in other than normal text must be indicated as such; for such cases, use macros without arguments such as `\eg`, `\etc`, and so on, and macros with arguments such as `\code`, `\mission`, `\foreign` (for a foreign word which should be italicized).

The *Magazine*has a rule for tables: no rules! Tables should constain no horizontal lines. There is already space between the caption (which is above the table, so put it at the top in the `\table` environment) and the top of the table, and between the bottom of the table and the main text (if there is any; the table might be at the bottom of a page); also, the table has an obviously different font from the main text. If separation is needed between rows (*e.g.*, after the header or to separate rows into groups), instead of `\hline` use `//[3pt]` instead of just `\\` at the end of the previous row.

Tables and figures should be provided with captions. The style, layout, and numbering are automatic.

Within the main text, use, *e.g.*, `1/2` rather than `\frac{1}{2}`.

Use ‘‘’’ for direct quotations (except inside other direct quotations); use ‘’ everywhere else.

13 Plans for the future

This document will be updated as needed, as well the templates. Make sure that you have the latest version of each. Please report any problems, suggestions, *etc*. to `manager@obsmag.org`. The first goal is to get everything working so that the *Magazine* can be can be produced in LaTeX as of the 2026 February issue. The same templates as used by the Editors should also be usable by authors, so the next goal is this documentation, which can incorporate feedback from authors. The third item is to embed the functionality into a more standard template for the LaTeX world, which should make it easier to become part of third-party installations such as Overleaf. Priority will be given to features requested by authors. In the near future a more conventional style guide will appear, accessible from `http://www.obsmag.org/templates/`.