WITA Manuscript Submission Guide

Neruthes   (WITA Editor)

2025-07-14

Abstract

Like most journals, Weight in the Attention (WITA) handles manuscript submissions. A rare characteristic of WITA is that the submission process is based on Git, GitHub, and pull request. This guide offers a comprehensive guide for authors in good faith of establishing an efficient manuscript acceptance workflow. Key takeaway — if you would like to submit an already published article, just open an issue and include URL to your article.

1   Directory Structure

Issues are grouped by year inside the "/issue" directory. Each issue consists of a "tex" file and a "tex.d" directory.

1.1   Issues Catalog

Sketch of directory structure "/issue":

```
/issue/
  |- 2025/              -> Year, format YYYY
    |- 202508.tex       -> IssueID, format YYYYMM
    |- 202508.tex.d/
      |- meta/
      |- entry/
```

1.2 Inside Single Issue

Sketch of directory structure "/issue/Year/IssueID.tex.d":

```
../202508.tex.d/        -> IssueDir
  |- meta/
  |- entry/
    |- 001/             -> EntryID
```

2 Organizing Your Entry

Sketch of directory structure "/issue/Year/IssueID.tex.d/entry/EntryID/":

```
../001/                 -> EntryDir
  |- info.toml
  |- main.tex
  |- cite.bib
```

You should have these 3 files. In addition, you may have subdirectories for images and code pieces.

2.1 info.toml

This is a TOML file that contains the metadata of the submitted manuscript.

Example:

```
[[article]]
id = "myrun"                # Same to EntryID
title = "Article Title Goes Here"
authors = ["John Appleseed (Reed College)", "John Doe (*)"]
authors_simple = "Appleseed, J., et al."
email = "user@example.com"
date = "2025-07-13"
license = "CC BY-ND 4.0"    # Omit NC to allow more
```

2.2 main.tex

This is where your article content goes.

Example:

```
\setentryid{myrun}
\stdarticle{Article Title Goes Here}{%
    \authorrow{John Appleseed}{(Reed College)}\\%
    \authorrow{John Doe}{}%
}{2025-07-13}
```

You should start the file like the example.

Here are commands you can use:

- `\setentryid`
  Declares article EntryID.

- `\stdarticle{Title}{Authors}{SubmissionDate}`
  Print article title, author, and date information.

- `\entrypath`
  Get path to entry. Useful for `\includegraphics{\entrypath/pic-1}`.

- `\authorrow{Name}{Institution}`
  Prints a row of author information. Must use inside argv1 of `\stdarticle`.

2.3 cite.bib

Unlike others, **cite.bib** may be omitted if your manuscript contains no citation at all.

3 Submission Workflow

Follow these steps to submit your article to WITA.

1. **Fork Repository**: Fork the repository and clone your fork to your machine.

2. **Elect EntryID**: Decide a short string for your article that is unlikely to collide with other authors of the same entry.

3. **Create Branch**: In your local repository, create a branch named after your username (e.g. "johndoe/entry256").

4. **Create Entry**: In your branch, create relevant directories and files.

5. **Local Build**: Run the issue building workflow locally and debug problems, if any.

6. **Create Pull Request**: Create a pull request that merges from your fork "johndoe:johndoe/entry256" to upstream master branch.

7. **Await Editor Review**: Editors will review your article and will very likely accept it.

The following list contains further clarification.

1. You are free to, and encouraged to, self-publish your article anywhere else such as your personal blog. It is *your* article, after all.

2. If you include non-empty "email" field in your "info.toml" file, an editor will mail you a letter of acceptance when your article is accepted.

3. Even if your article is accepted, there is no guarantee that it will be included in the next coming issue.

4. When creating pull request, use dummy IssueID "000000" (/issue/0000/000000.tex.d). An editor will move your article to an upcoming IssueID after acceptance.

5. To check whether your article is bug-free, you can run ./make.sh issue/0000/000000.tex. If you do not have a GNU/Linux machine, try WSL.

6. Your article should have a redistributable license such, e.g. CC BY-ND, CC BY-SA, GFDL. Alternatively, you can make it public domain. Open knowledge is great.

7. LLM-created articles are encouraged to be submitted as public domain work.

8. If the article is created by LLM, the LLM should be attributed as an author. Name can be online public service (e.g. Gemini, ChatGPT) or model identifiers (e.g. Qwen3-32B).