

## Working remotely from an island: arara and other tools

Island of T<sub>E</sub>X (developers)

### Abstract

Over the last two years, the Island of T<sub>E</sub>X has complemented the T<sub>E</sub>X ecosystem with some auxiliary tools. This article is a short review of the last year’s (more or less) achievements and a preview of upcoming changes.

### 1 Providing a home for T<sub>E</sub>X-related projects

In 2019, the Island of T<sub>E</sub>X started as a small group of two friends trying to improve the T<sub>E</sub>X ecosystem (cf. fig. 1). What started as small steps towards a new arara release with some little side-projects became a more and more interesting place to stop by in 2020.



Figure 1: The Island of T<sub>E</sub>X logo.

The last year started off in full preparation of releasing arara version 5 in due time for the T<sub>E</sub>X Live 2020 pretest. A new arara version, new problems. Software engineering becomes “fun” as soon as users are concerned, yet we are determined to help our users. More about where this view got us and arara in a later section.

Apart from our flagship project, the island focused on new frontiers over the whole last year. Therefore, we

- stabilized T<sub>E</sub>Xplate,
- created an archive for stale T<sub>E</sub>X related projects (containing only a backup of the  $\epsilon\lambda$ T<sub>E</sub>X repository for now) to preserve history,
- migrated checkcites, a tool to check for missing or unused references, to the island,
- published the new albatross tool, and
- worked on publicity.

Of our efforts, three projects received a fair share of attention: our Docker images, the new T<sub>E</sub>Xdoc online tool and the shiny little albatross. Let us drop some words about each of these projects before looking at arara and the future.

### 2 New tools for a modern T<sub>E</sub>X ecosystem

In TUGboat 40:3,<sup>1</sup> we introduced the island’s Docker images for easily reproducible builds as well as a semi-official response to the need for continuous integration (CI). Our images were among the first using vanilla T<sub>E</sub>X Live, providing the required tools for running software included in T<sub>E</sub>X Live (Java Virtual Machine (JVM), Python, etc.). Additionally, we provide T<sub>E</sub>X Live releases from 2014 on and let the user decide whether they want to pull all the documentation and source files into their CI configuration.

With attention came the idea to more officially publish our images as texlive/texlive, which we gladly did. Now we are managing the Docker Hub releases at <https://hub.docker.com/r/texlive/texlive>. Although unnoticed at first, we even saw DANTE e.V., the German-speaking T<sub>E</sub>X user group, basing their Docker images on ours.

Apart from applications in CI, the Docker images have lured us into creating another tool based on them. T<sub>E</sub>Xdoc online, which we introduced in TUGboat’s previous issue,<sup>2</sup> is now the software running <https://texdoc.org>, the successor to <https://texdoc.net> (which now redirects), thanks to Stefan Kottwitz. We have incorporated a few improvements, among others HTTPS support. You can easily host your own instance.

After finishing T<sub>E</sub>Xdoc online to a production-ready degree, the island turned to the development of another handy tool, albatross. This command line tool, with a silly yet adorable name, solves a very common problem: finding (system) fonts that provide a certain glyph. Users may provide the glyphs themselves — e.g.  $\beta$  — or their corresponding Unicode code points in hexadecimal notation — e.g. 0xDF. Currently, it is a thin wrapper around fc-list but there are plans to make it even more useful. As it is in T<sub>E</sub>X Live, you should give it a shot.

### 3 arara — feeling at home on the island

In 2019, arara, the cool T<sub>E</sub>X automation tool, was one of the first new citizens of the Island of T<sub>E</sub>X. It moved just in time to work on a new version, version 5. This version was special in many ways, first and foremost as it followed its predecessor after such a short period of time.<sup>3</sup> Behind the scenes, we finished a major rewrite of arara, mainly working on features from user feedback, especially directory support and the processing of multiple files.

<sup>1</sup> [tug.org/TUGboat/tb40-3/tb126island-docker.pdf](https://tug.org/TUGboat/tb40-3/tb126island-docker.pdf)

<sup>2</sup> [tug.org/TUGboat/tb41-3/tb129island-texdoc.pdf](https://tug.org/TUGboat/tb41-3/tb129island-texdoc.pdf)

<sup>3</sup> Some members of the T<sub>E</sub>X community at StackExchange might remember that working on arara version 4 was one of Paulo’s major distractions while writing a never-ending thesis.

After releasing version 5 for the T<sub>E</sub>X Live 2020 pretest, we got hooked by the idea of aligning release schedules of `arara` with T<sub>E</sub>X Live releases. So we had big plans and started to work on version 6 right after releasing version 5. We might have been a bit too ambitious, though, as we received some complaints about a non-working version 5 from our users. Somehow, they caught our failure to test the new release on some versions of Windows. Well, everything has been fixed and we were able to move on.

Approximately three quarters of 2020 remained and we tried to make version 6 shine through an enhanced feature set and optimized workflows. Some new features we want to highlight:<sup>4</sup>

- Preambles (think of that as the commands `arara` will execute on that file) have received new options. Among others, you may now define your workflows using preambles and set a global default preamble. That way, you may now even call `arara` on files without those special `arara` comments (directives) and make our tool (auto)magically execute your default preamble. Users wanted to be able to switch their editor's default compiler to `arara`, even for files without explicit statements, and now they can.
- You may pass parameters from the outside into your build flow. Call `arara` as

```
arara -P jobname=thesis file.tex
```

and you can receive that parameter within your `arara` rules and directives like this (line breaks added for *TUGboat* formatting; this directive should be on one line):

```
% arara: pdflatex: { options: [
  "-jobname=@{getSession()
    .get("arg:jobname")}" ] }
```

Multiple users have requested being able to parametrize their directives, so we finally managed to implement it.

- We added eight new rules and improved the rule format. The most frequent requests we receive are about supporting new tools in our rule set. We gladly add more T<sub>E</sub>X tools to `arara`, so if you are missing something feel free to contact us.
- For quite some time, `arara` has been a very powerful tool and has been criticized for being so powerful. We now implemented a first draft of a safe mode that restricts `arara` in some of the more harmful execution steps. Do not expect real safety, though. This is going to take more work to prevent obvious malicious behaviour.

<sup>4</sup> A detailed discussion of new features is in our blog post about `arara`'s new release on its website at <https://islandoftex.gitlab.io/arara/>.

- One of the most prominent problems with `arara` has been the lack of good introductory material, especially as the manual has grown. Hence, we now provide a quick start guide for new users. If you do not use `arara` yet, maybe this guide is for you. Interested? Simply run `texdoc arara-quickstart` on your local system.

The above is only a short excerpt of the change log but probably the most important changes for users. So to come back to the initial statement: version 6 is a major milestone in many ways but most importantly because the technical improvements of version 5 allowed us to implement so many features our users have been waiting for (sometimes for years).

## 4 Perspectives

The Island of T<sub>E</sub>X will continue to work on improving the T<sub>E</sub>X ecosystem in 2021. We hope to be as productive as we were in 2020. So let us try to outline our near-term goals.

First of all, we want to improve `albatross` and `checkcites` to make them even more useful. For the latter that will most probably include a rewrite. Anyway, these small helpers are what makes the daily T<sub>E</sub>X workflow a bit more fluent so we will try to gather new ideas for little new helpers. We have welcomed a new member to the island, who will reveal his first IoT tool soon.

However, the major plans and projects are settled around `arara`. In the long run, we want the tool to provide far more than just a CLI tool for compiling T<sub>E</sub>X. We have plans for a documented API for defining T<sub>E</sub>X flows that is not as `latexmk`-centred as some of the existing APIs floating around and some kind of `arara` daemon that will be able to translate multiple documents in parallel.

Furthermore, most of our tools are written in Kotlin and therefore rely on the JVM. With the so-called Kotlin multiplatform projects (MPP), we will try to get closer to native performance. Among others, this could also be useful in scenarios where a computer (or cloud service for that matter) with a T<sub>E</sub>X Live installation may not run a JVM.

To wrap this up, we have many plans and hope to realize as much as possible. If you are interested in helping us develop ideas or even implementing some code: visas for the island are free and easy to get, so feel free to reach out.

◇ Island of T<sub>E</sub>X (developers)  
<https://gitlab.com/islandoftex>