The Island of TEX: Developing abroad, your next destination

Island of TEX (developers)

Abstract

The Island of TEX is a collaborative effort to provide a home to community-based TEX projects. This article discusses the Island's long-term goals and how the worldwide community can come aboard and help the organization enhance the TEX experience for everybody, from newbies to power users.

1 The beginning

Once upon a time, a group of friends decided to create an organization as a means to systematize and concentrate our efforts in developing the TEX ecosystem. Thus, the Island of TEX, a collaborative endeavour to provide a home to community-based open source TEX-related projects, was finally charted in the TEX world map. Island residents include Paulo Cereda, Ben Frank, Brent Longborough, Marco Daniel, Nicola Talbot, and Enrico Gregorio.

So far, the Island holds four main groups of projects: Docker images, programming libraries, assorted tools, and TEX editors. We plan more groups for the near future.

2 Docker images

The first group to be discussed refers to Docker images. Docker is an open platform for developing, shipping, and running applications, enabling a clear decoupling of applications from infrastructure.

A Docker image itself is similar to a snapshot of a lightweight virtual machine. When running an image, it shares the same kernel as the host system but provides a complete and independent infrastructure of operating system and software packages, as well as configuration files and environment variables.

The island provides Docker images for TEX Live repositories. It is important to note that we also provide the necessary tooling to execute common helper tools. We have two groups: *historic*, which, as the name implies, contains releases from 2014 on, and *latest*, which refers to the current stable release plus all updates available as weekly snapshots.

For every TEX Live release, from historic to latest, we provide four flavours: binaries only, binaries and documentation, binaries and sources, and the full pack, binaries, sources and documentation. When in doubt, choose binaries only and only pull the larger images if you have to. Keep in mind that sources and especially documentation files do add a significant payload. If you need an image for an older TEX Live feel free to file a feature request. We might consider adding older distributions if there is a user base.

The Island also provides Docker images for the ConTEXt distribution (full installation with all modules). It also provides the necessary tooling to execute common helper tools. We have three groups: MkIV current (updated monthly), MkIV beta and LMTX (updated weekly). If you need a different image for ConTEXt as well, feel free to file a feature request.

3 Programming libraries

The second group to be contemplated refers to programming libraries. The Island currently provides three libraries: a SyncTEX parser, a TEXdoc API and a TEX log analyzer (which will be omitted here due to its pre-alpha stage).

3.1 SyncTEX parser

SyncTEX is a synchronization technology supported by all the major engines. It maps the input to boxes and point positions in the output. Therefore, it can be used for forward and backward synchronization, that is, finding the appropriate point in the output PDF for the current input line, or finding the respective input line for the current point in the output.

While collecting ideas for other tools, we came to the conclusion that there is no comprehensible and up-to-date structure definition of a SyncTEX file, so we started to implement a parser, partially as an exercise to know the file format and partially to provide some structured representation to the community.

It is important to note that we have not looked at the official SyncTEX parser while implementing this component and we do not intend to. We read only the man page and started to look at real world examples. Hence this is neither official nor guaranteed to parse all SyncTEX files. That said, you are most welcome to open issues and merge requests, contributing test cases where it fails.

3.2 TEXdoc API

The next library in the Island is an API for the `texdoc` command line interface. Hence, it is only one layer of abstraction; we currently require the presence of `texdoc` on the target machine.

The API provides two classes: the `Entry` class which represents an entry in `texdoc`'s list of results and the `TeXdoc` class, presented here, which is a singleton, providing an interface to the actions of the command line utility. The most important methods and attributes of `TeXdoc` are:

`isAvailable` to check whether `texdoc` is in the system path,

`version` to get the version string of the `texdoc` installation,

`getEntries` to get a list containing all entries from the result set of a `texdoc` query given the provided keyword, and

`view` to view the resource specified by an entry; it returns a boolean to report whether the process exited successfully.

4 TEX-related tools

The third group in the Island refers to TEX-related tools. We currently hold four projects: TEXprinter, TEXplate, `checkcites` and `arara`.

4.1 TEXprinter

TEXprinter is an application designed for the purpose of printing threads from the TEX community at StackExchange. It can print threads in PDF and TEX formats. The PDF output is provided by the `iText` library. It is a quick option if you do not intend to customize the output. If the thread has images, they are embedded in the final result.

The TEX option is recommended if you want to format the code the way you like. It basically uses the `article` document class, the `listings` package and a fairly straightforward approach. If the thread has images, they are downloaded to the current directory and correctly referenced in the document. Of course, you need to compile it.

TEXprinter ships as a standalone JAR file with dependencies. There is no need to install it, simply download the JAR file from the project repository and execute it. Keep in mind that it needs the Java virtual machine installed.

4.2 TEXplate

The next project in the Island is TEXplate, a tool for creating document structures based on templates. The application name is a word play on *TEX* and *template*, so the purpose may be clear: we want to provide an easy and straightforward framework for reducing the typical code boilerplate when writing TEX documents. Also note that one can easily extrapolate use beyond articles and theses.

The application is powerful enough to generate any text-based structure given that a corresponding template exists. TEXplate can also be used for package writers in generating automated tests. The tool is already available in your favourite TEX distribution.

4.3 checkcites

The third project in the Island is `checkcites`, a Lua script written for the purpose of detecting unused or undefined references from both auxiliary or bibliography files. We use the term *unused reference* to mean a reference present in the bibliography file but not cited in the TEX file. The term *undefined reference* is exactly the opposite, that is, the item cited in the TEX file, but not present in the bibliography file. The script supports two backends, BibTEX and `biber`, and can detect files from your TEX tree.

4.4 arara

Finally, the fourth project in the Island is `arara`, the cool TEX automation tool. This is probably our most well-known and widespread project, as well as the oldest one.

`arara` is a TEX automation tool based on rules and directives. It gives you a way to enhance your TEX experience. The tool is an effort to provide a concise way to automate the daily TEX workflow for users and also package writers. Users might write their own rules when the provided ones do not suffice. `arara` is currently at version 5.1.3; some noteworthy features of version 5 include:

- Support for working directory: users may now specify their working directory by specifying a command line option, so commands will run from a different directory than the directory `arara` launched in. This is especially useful when calling a TEX engine as they resolve files against the working directory.
- Support for processing multiple files: `arara` is able to compile multiple files at once by providing multiple files as arguments. Please note that they should reside in the same working directory. Every other kind of compilation of multiple files is restricted by the mechanisms of the running programs.
- Updated rule pack: `arara` features more than 60 rules ready for use, including TEX engines (with support for both stable and development branches) and assorted tools for graphics, bibliography, indexing, cleaning and conversion between file formats. A typical user might rely just on this pack, without the need of writing a custom rule.

Version 5.1 of `arara` is already available in TEX Live 2020. Simply execute `arara` in your terminal and have fun. The Island has an interesting workflow for `arara`. When a new version is ready to be released, we simply tag it and GitLab will build a CTAN release. Our build script compiles and packages the

application binary, typesets the documentation using our Docker images, assembles the CTAN tree, as well as the accompanying TDS ZIP, and provides a full CTAN ZIP file artifact ready for download. Then we simply send this file to our friends at CTAN.

We are discussing some new features and development paths for the next major version of `arara`.

- We plan to split our tool into an API, a core implementation (that is, a library) and the implementation of the executable (that is, a command line interface). Projects relying on code in the `arara` JAR distributions have to be updated (which might be a potentially breaking change).
- Languages will have to be passed as IETF BCP 47 codes. The old system will be removed.
- Localization will be provided by classes as a library instead of property files in `arara`'s resources. We want to decouple localization from the core implementation and make it easy for users to contribute their own messages.
- The log file shall be specified as path anywhere on the file system. This behaviour, however, may be altered for a future safe mode.
- We are working on a Kotlin DSL (domain specific language) to gradually supersede the YAML-based rule format. The new format aims at being significantly more expressive, easier to write and debug, and less error-prone.
- Elements marked as deprecated in version 5.0 are effectively removed, such as the `<arara>` shorthand notation. This will be a breaking change.
- We are working on a project configuration file format based on a Kotlin DSL as a means to provide resource dependency management, enhanced automation and several additional features. As a consequence, the `preamble` command line option and corresponding configuration map entry will be replaced.

You can follow the progress of version 6.0 in our repository, under the development branch. You can also join our Gitter chat room to learn more about our plans for this major milestone. And of course you are invited to contribute.

5 Editors

The fourth and last group in the Island refers to TEX editors. We currently hold one project which is an application named ArTEXmis.

ArTEXmis aims at being an opinionated, powerful TEX editor designed for all users, especially for package writers and kernel developers. We want to provide an elaborate, immersive user experience when writing TEX code, with a comprehensive and extensive set of features and actions. This project is still under development and we have not provided any public releases so far. We are working on it, so stay tuned for updates.

6 Final remarks

The Island of TEX is hosted at GitLab. Some projects originally hosted at GitHub, such as `arara` and `checkcites`, are now simply mirrored, so the development is consolidated under our organization. Issues and merge requests are welcome.

The TEX ecosystem is a very challenging yet exciting place. We have plans to migrate more tools to the organization in an organic way. At the moment, we have a new project being discussed with the core team. You can learn more about us from our official GitLab repository as well as *TUGboat* articles and electronic mail.

◇ Island of TEX (developers)
`https://gitlab.com/islandoftex`