CI Scripts

This directory contains scripts for each build step in each build stage.

Running a Stage Locally

Be aware that the tests will be built and run in-place, so please run at your own risk. If the repository is not a fresh git clone, you might have to clean files from previous builds or test runs first.

The ci needs to perform various sysadmin tasks such as installing packages or writing to the user's home directory. While most of the actions are done inside a docker container, this is not possible for all. Thus, cache directories, such as the depends cache, previous release binaries, or ccache, are mounted as read-write into the docker container. While it should be fine to run the ci system locally on you development box, the ci scripts can generally be assumed to have received less review and testing compared to other parts of the codebase. If you want to keep the work tree clean, you might want to run the ci system in a virtual machine with a Linux operating system of your choice.

To allow for a wide range of tested environments, but also ensure reproducibility to some extent, the test stage requires `docker` to be installed. To install all requirements on Ubuntu, run

```
sudo apt install docker.io bash
```

To run the default test stage,

```
./ci/test_run_all.sh
```

To run the test stage with a specific configuration,

```
FILE_ENV="./ci/test/00_setup_env_arm.sh" ./ci/test_run_all.sh
```

Configurations

The test files (`FILE_ENV`) are constructed to test a wide range of configurations, rather than a single pass/fail. This helps to catch build failures and logic errors that present on platforms other than the ones the author has tested.

Some builders use the dependency-generator in `./depends`, rather than using the system package manager to install build dependencies. This guarantees that the tester is using the same versions as the release builds, which also use `./depends`.

If no `FILE_ENV` has been specified or values are left out, `00_setup_env.sh` is used as the default configuration with fallback values.

It is also possible to force a specific configuration without modifying the file. For example,

```
MAKEJOBS="-j1" FILE_ENV="./ci/test/00_setup_env_arm.sh" ./ci/test_run_all.sh
```

The files starting with `On` (`n` greater than 0) are the scripts that are run in order.

Cache

In order to avoid rebuilding all dependencies for each build, the binaries are cached and re-used when possible. Changes in the dependency-generator will trigger cache-invalidation and rebuilds as necessary.