

Contributing Code: Best Practices

While not a hard and fast set of rules, please keep the following in mind when contributing code to CORD and XOS.

Code Style and Formatting

- Use a tab size of 4 spaces
- Eliminate trailing whitespace from files
- Use the [PEP8](#) formatting style in Python files

Code Submission

- See [Gerrit Workflow](#) for details on how to create and submit code changes.
- If you're doing a reformatting or cleanup to a file, and also making code changes to that same file, this should be done as two separate submissions (cleanup first, then changes).
- Unused or commented out code shouldn't be submitted, and any code that is no longer used or made obsolete should be removed.

Dependency management

- If the changes require a new dependency or other 3rd party code to function, these dependencies should be added to the appropriate Dockerfile, preferably using the system package manager (`apt`). See [Best practices for writing Dockerfiles](#) for further information on how to do this efficiently.
- Take steps to guarantee the integrity of your dependencies. The package manager (`apt`), `pip`, and others perform these steps behind the scenes, so using them is highly recommended.
- If you need to download something from an arbitrary URL, generate a checksum (`sha256` or better recommended) for a known good copy and have your build step verify that it matches before using it. Do not pipe an URL into a shell or other interpreter.
- If you are bringing large files into the docker image and using `make` or a similar dependency based build tool, have that tool download the file once then use Docker's `COPY` to add it to the docker image, rather than downloading it every time the image is rebuilt.
- You must include all the code that is being used at runtime during the build steps. This allows the system to continue to operate the same way behind a restrictive firewall or on a network disconnected from the internet.
- Javascript libraries used by a web browser when someone access the CORD or XOS web interface should be downloaded and added to the build as described above, not retrieved from the internet at runtime by the user's browser.

Temporary and generated files

- Any files generated as artifacts of the build process should be added to `.gitignore` files to prevent them from being inadvertently added to future commits.
- Don't include any passwords, keys, or credentials in the code you submit.

Copyright and licensing

- All included files must be compatible with the Apache 2.0 license, and free from any copyright or other licensing issues.