

DView Contribution Policy

Version 1.0

The DView team welcomes your contribution to the project. DView is open source (comprised of two main repositories, LK and WEX), and user contributions of code, documentation, or any other kind of contribution moves the project forward. The process for submitting contributions and obtaining approval is described below. If you contribute code, you agree that your contribution may be incorporated into DView and made available under the DView license on the internet or via other applicable means.

Our open source software benefits greatly from the contributions of its user community. If the contribution that you propose is aligned with DView's goals and strategy, we would love to hear from you. The contribution process for DView is composed of three steps:

1. Send consent email

In order for us to distribute your code as part of DView under the DView license, we'll need your consent. An email from you acknowledging that you understand these terms and agree to them is all that will be asked of any contributor. The return email from you should include the following text and a list of co-contributors (if any):

I agree to contribute to DView. I agree to the following terms and conditions for my contributions: First, I agree that I am licensing my contributions under the terms of the current DView license. Second, I agree that, in order to conform to any future open source software license(s) under which DView may be provided, the terms of my license may be modified without any notice to me and without my consent. Third, I represent and warrant that I am authorized to make the contributions and grant the license. If my employer has rights to intellectual property that includes my contributions, I represent and warrant that I have received permission to make contributions and grant the required license on behalf of my employer.

Once we have your consent on file, you'll only need to redo it if conditions change (e.g. a change of employer).

2. Scope agreement and timeline commitment

If your contribution is small (e.g. a bug fix), simply submit your contribution via a GitHub pull request. If your contribution is larger (e.g. a new feature), we'll need to evaluate your proposed contribution. To do that, we need a written description of why you wish to contribute to DView, a detailed description of the project that you are proposing, the precise functionalities that you plan to implement as part of the project, and a timeframe for implementation. After we review your materials, we may ask you to revise your materials and make changes to it, which we will re-review.

Before you do any work we should reach prior agreement on project areas, scope, timeframe, expected contents, and functionalities to be addressed.

3. Technical contribution process

We want DView to adhere to high quality standards. As such, we ask that you follow our development process - particularly with respect to coding standards, code review, unit tests, and code coverage. These items are explained further below. Smaller, non-code contributions may not require as much review as code contributions, but all contributions will be reviewed. Code contributions will initially be in a source control branch, and then will be merged into the official DView repository after review and approval. Any bugs, either discovered by you, us, or any users will be tracked on the GitHub issues page for the specific repository (LK or WEX). We request you that you take full responsibility for correcting bugs. Be aware that, unless notified otherwise, the correction of bugs takes precedence over the submission or creation of new code.

Release Schedule - DView is released publicly, several times per year, and coincide with releases of OpenStudio and SAM. If your contribution is incomplete or comes in past the cutoff date for a release, we reserve the right to hold your code for a later release to ensure adequate testing is accomplished.

Development Process - Work items are tracked and planned in GitHub issues as bugs or enhancements. Your bug or enhancement issue should be entered on GitHub issues, with a workplan. As your work commences, you should periodically comment (every couple of weeks) on the issue to give a progress update. A review of your contribution will start when you've completed your work and issued a pull request.

Coding Conventions - We will point you to coding guidelines to help you write DView code that is consistent with style we would like you to adopt. We recognize that the style of the current LK and SSWEXC code repositories is inconsistent, and does not necessarily follow the guidelines, but we plan to update the code as we have the time.

Code Reviews - You will be working and testing your code in a fork or branch. When a piece of functionality is complete, tested and working, send a pull request and we will review your code. If the functionality that you contributed is complex, we will ask you for a written design document as well. We want your code to follow the coding guidelines, be clear, readable and maintainable, and of course it should do what it is supposed to do. We will look for errors, style issues, comments (or lack thereof), and any other issues in your code. We will inform you of our comments and we expect you to make the recommended changes. New re-reviews may be expected until the code complies with our required processes.

Unit Tests - We ask that you supply unit tests based on Google Test along with your code (currently in progress) . A unit test is a program that exercises your code in isolation to verify that it does what it is supposed to do. Your unit tests are very important to us. First, they give an indication that your code works according to its intended functionality. Second, we will eventually execute your unit tests automatically along with our own tests to verify that the overall DView code continues to work.

Code Coverage - We require that your unit tests provide an adequate coverage of the source code you are submitting. You will need to design your unit tests in a way that all critical parts of the code (at least) are

tested and verified. A good rule of thumb for code coverage is 70% or greater. There are tools to help you with code coverage.

Documentation - Proper documentation is crucial for our users, without it users will not know how to use your contribution. If you add a new feature to DView, we require that you create user documentation so that end users know how to use it. You can submit your documentation as a Word, LaTeX, Markdown or simple text document that our team will integrate into the appropriate Wiki.

For further questions or information:

Nicholas DiOrio

SAM Software Engineering

nicholas.diorio@nrel.gov

303.384.7359