Software defect (bug) reporting

Reporting a bug

Bugs found in the VAPOR package should be reported on the sourceforge project site for VAPOR: https://sourceforge.net/p/vapor under Tickets/Bugs

When reporting a bug keep in mind that both developers and our user community have access to, and make use of, the information in the bug database. Even after the bug is closed developers and users may find the information in the database useful. For example, trying to determine if a problem found in an older release has been addressed in a newer release. Be sure to include/perform the following:

1. **Title**: A *descriptive* title that hopefully has meaning to someone not familiar with the bug.
2. **Milestone**: If the bug can be reproduced in a released version of the code set the milestone to that release number. If the bug only occurs in code that has never been publicly released (e.g. a development regression or code providing new functionality) set the milestone to "regression".
3. **Owner**: If you think you know who has responsibility for the code involved go ahead and assign it to them, otherwise leave this field blank. It is no big deal to re-assign bugs to different developers at any time. So don't worry about getting this right.
4. **Labels**: Sourceforge's current platform does not support definition of categories to help classify what components of the VAPOR package might be impacted by the bug. The **labels** field can, however, be abused for this purpose. Enter as many labels as necessary to isolate the impacted code/capability. For example, always enter the name of the tool (e.g. vaporgui, vdcwizard, romsvdfcreate, etc.), or the library if multiple tools are impacted (e.g. vdf). For vaporgui, it is helpful to also include a label identifying the tab (e.g. DVR, iso, Preferences, etc.) if appropriate.
5. **Priority**: It is not important to get the priority "correct" when opening a bug. We generally review bugs after they are opened and assign priorities to them. In general any bug that is "sev 6" or higher is one that we make every effort to fix in the next release. If you're unsure about how critical a bug is just use the default priority: 5. If you think it's something that is really critical go ahead and give it a higher priority. Even though sourceforge provides 10 different levels we generally only use 3 or 4:

    - sev 4: minor issue that may not get fixed ever
    - sev 5: we'll try to fix it, but it may not happen;
    - sev 6: fix in next release;
    - sev >6: super critical

1. **Description**: This is the big one.
    a. Provide enough detail to describe the problem. Remember, even if you plan to fix the bug you're reporting you should still provide enough information so that other readers can understand what the problem is. Someone else may be assigned to fix a problem even if it is in "your" code.
    b. **Provide all the steps necessary to reproduce the problem. Again, assume that someone else may have to fix the problem even if it you "own" the code**. Try to provide:
        - **Step-by-step** instructions. E.g. load this data, then enable DVR, turn this nob, etc.
            - For **command line utilities** provide the **EXACT** invocation that you used.
        - A reference to a data set that demonstrates the problem. Use the canonical data sets in the VAPOR "repository" whenever possible (/glade/p/DASG/VAPOR/Data). If the problem can only be reproduced with a data set not found in our library, try to reduce that data as much as possible and place a copy in /glade/p/DASG/VAPOR/Bugs/BUGNUMBER, where BUGNUMBER is the number of the bug generated by sourceforge. **Do NOT reference data sets that are in your home directory, on scratch, etc. as these tend to go away**. If the problem can only be demonstrated with a data set that is rediculously large (e.g. > TB) make other arrangements with the bug owner.
        - A session file can be really helpful for vaporgui problems
        - For subtle rendering problems producing rendering artifacts attaching an image can be helpful
        - The platform you are running on (OS name and version), or the name of the system (e.g. geyser)

Fixing bugs

What should you do after you "think" you fixed a bug? Edit the bug report:

1. **Milestone**: Milestones for closed-fixed bugs that ARE NOT REGRESSIONS will get set to the release that they were fixed in. It's not critical to remember this, as after the released is dropped we go back and set the milestones for all fixed, non-regressive bugs. If the bug was a regression leave the milestone UNCHANGED.
2. **Status**: If you made a code change that fixed a bug change the status to "closed-fixed". If you are unable to reproduce the bug set the status to "open-works-for-me", and notify whoever submitted the bug that you can't reproduce it. This notification can be done with the "comment" field. If the bug creator is unable to reproduce the problem change the status to "closed-works-for-me"
3. **Comment**: Use the comment field to enter any additional information that might be helpful.