

Bug –Assist

Step by Step Instructions

System Requirements.

- 1 Download the eclipse plugin for Bug-Assist from the download page for your architecture. Currently we support Linux (32/64) and Mac only.
- 2 Must have eclipse version 3.2 or above. For C/C++ development it is advised to have Eclipse IDE for C/C++ languages.
- 3 The system should have gcc/g++ version above 3.0 and awk installed in the path variable.

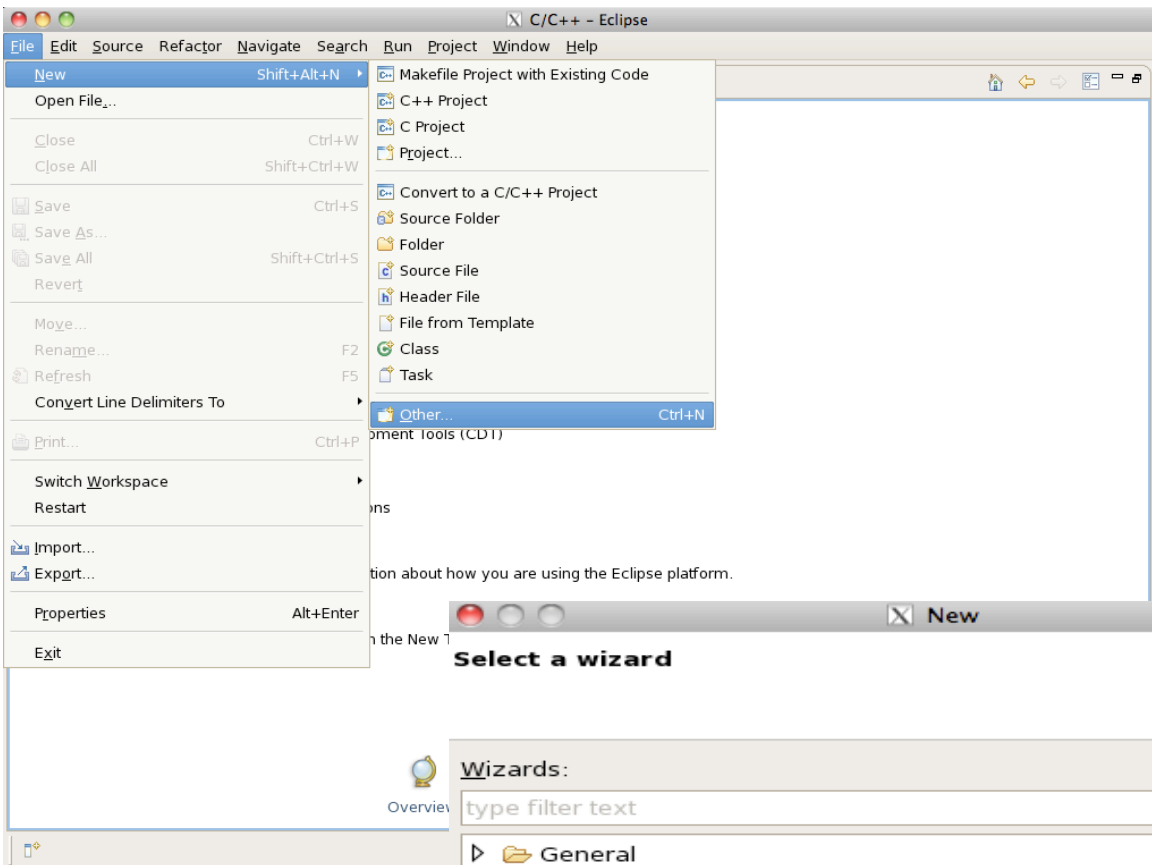
Plugin Installation.

- 1
- 2 Go to the eclipse installation direction [eg : `$ cd /local/share/ECLIPSE/`]

Unzip bug-assist.plugin.zip inside the plugin directory inside ECLIPSE.

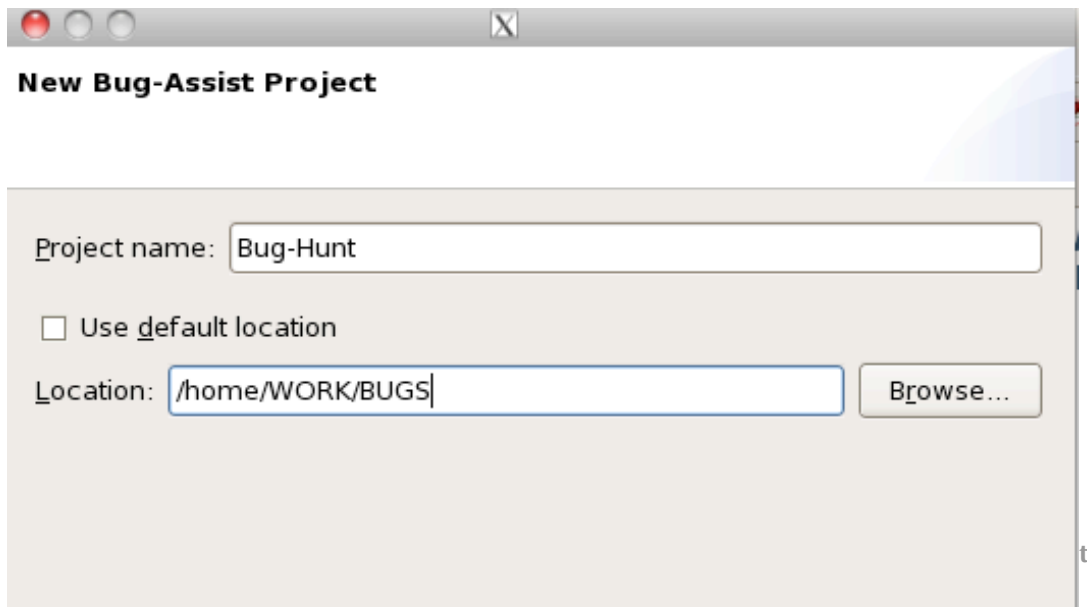
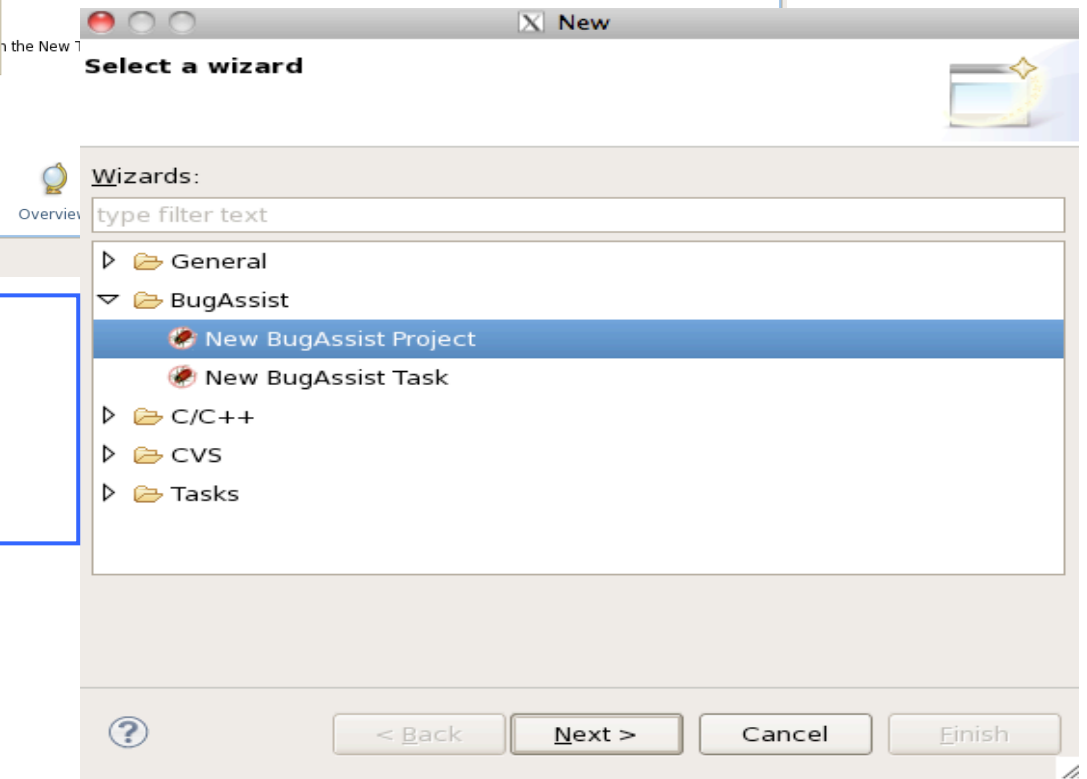
Creating a new bug-assist Project.

Start eclipse and do File->New->Other as shown in the figure below.

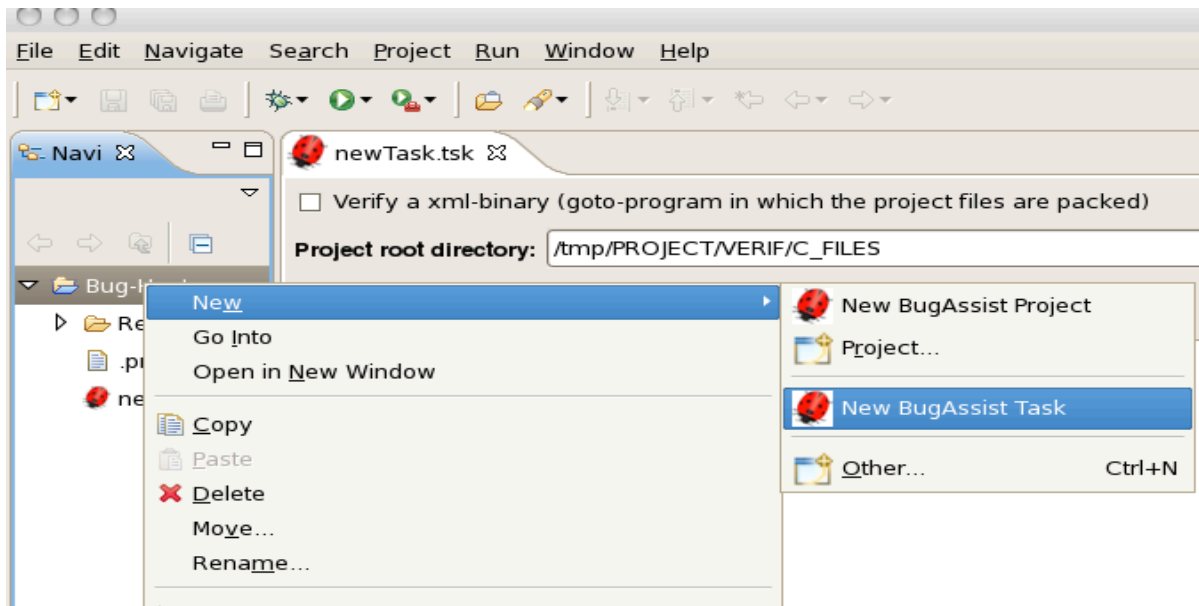


Select "New BugAssist Project" as in right.

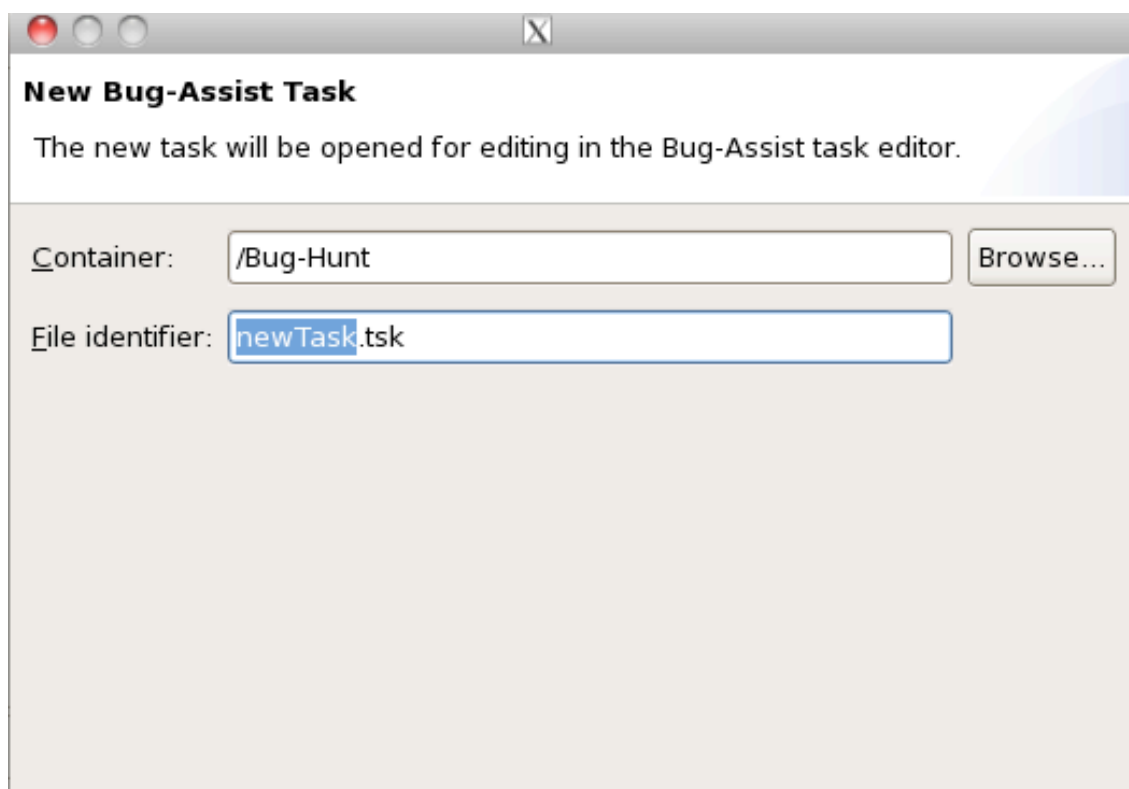
Create a Project name and a location to work with the files you want to check for files. This way you don't tamper the files in the original project. make the



Creating bug-assist Task and adding C files.

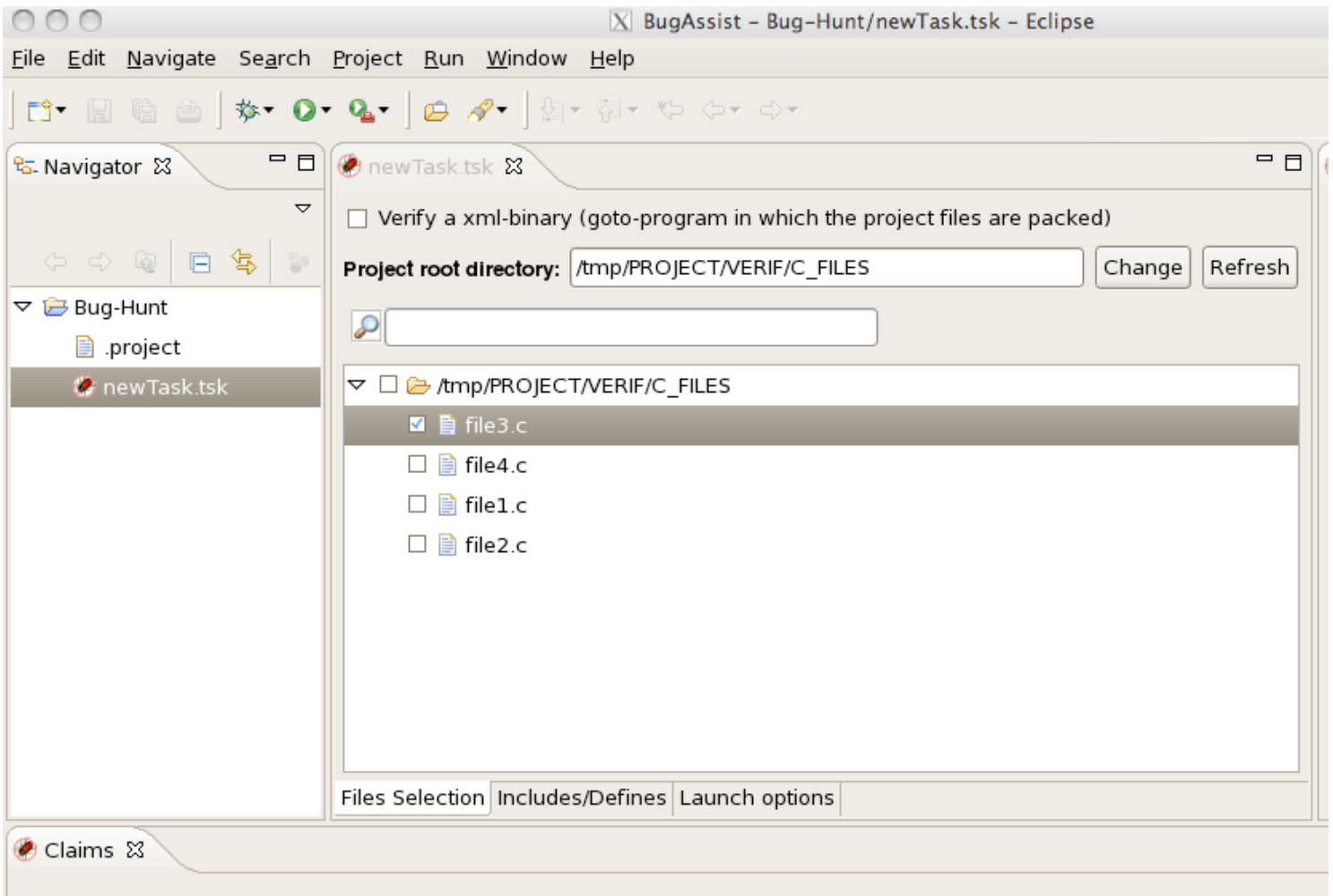


As Shown in the figure above, right click on the project folder “bug-Hunt” and select “New BugAssist Task” and then create a new task as shown in the



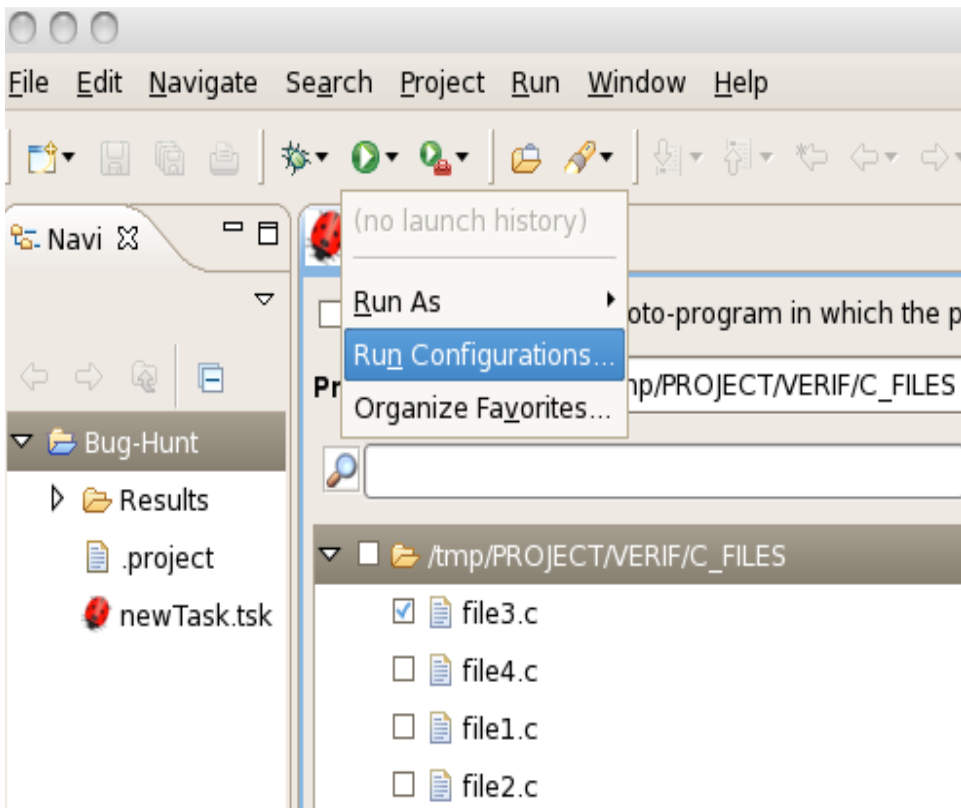
(continued)

Select “newTask.tsk” and change the root directory to the directory where the C files to check for bugs are located. In this example /tmp/PROJECT/VERIF/C_FILES. You have an option to select one or a bunch of files to check for bugs as shown in the figure below. Double clicking a file will open the file in the editor.

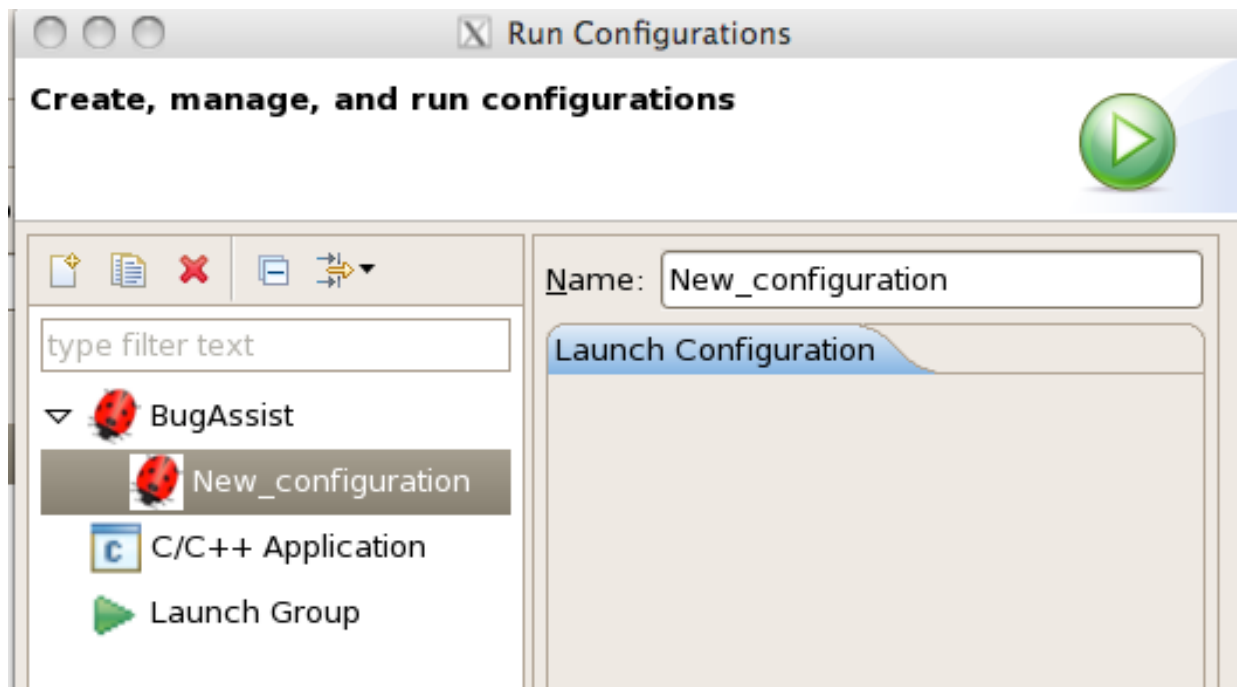


This completes loading the files in to the bug-assist system for checking. Now we start the model checking engine to look for assert statement violations with in the selected files. First step is to set up a run configuration which calls the model checker and lists the assertions specified in the program.

Select the green play button on the menu and Select Run Configurations as shown in the figure below.



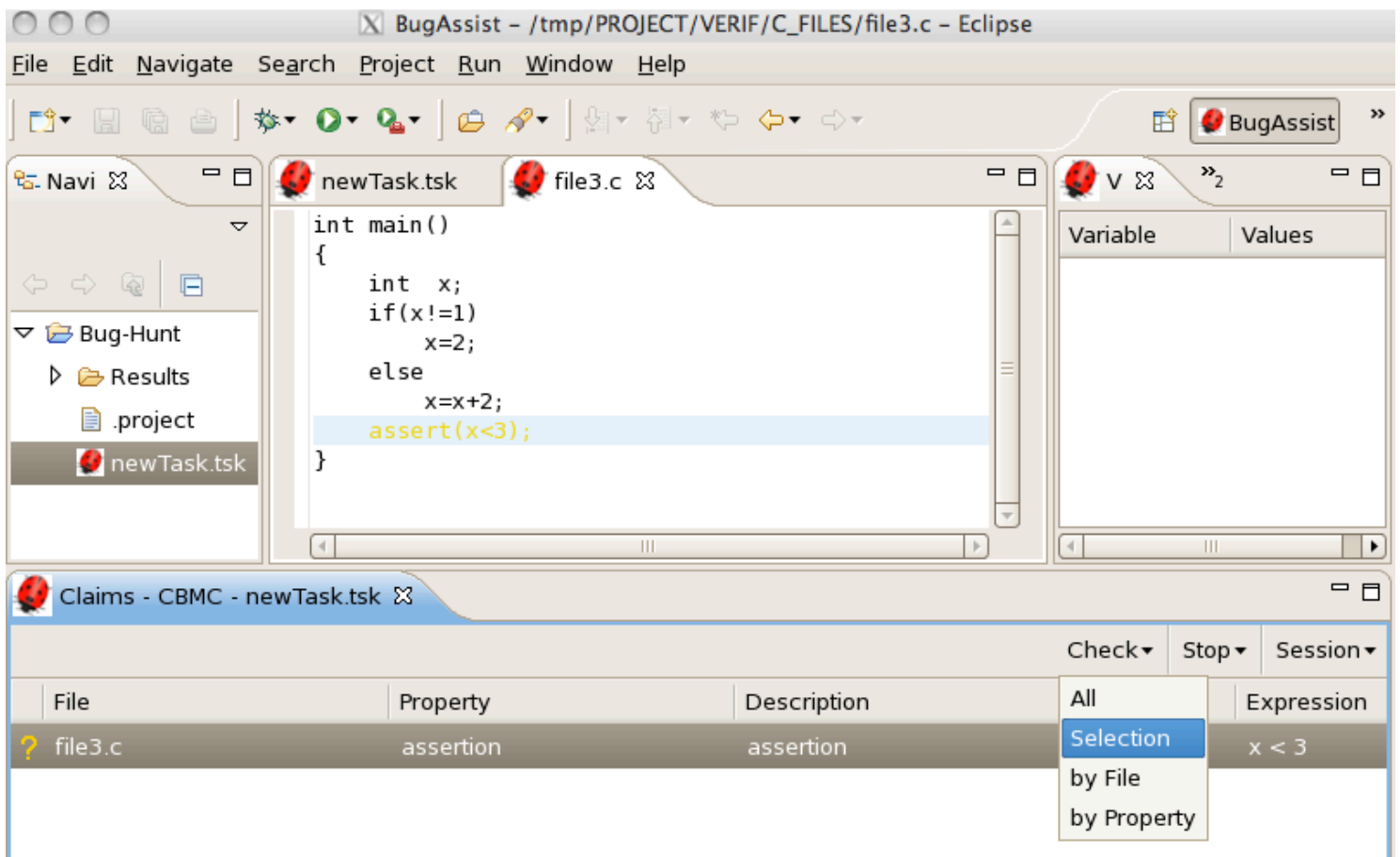
Double click on the BugAssist button which creates a new run configuration. Give a name and save the configuration as shown in the figure.



Click on the Run Configuration newly created. This will list the assertions in the selected file as a list.

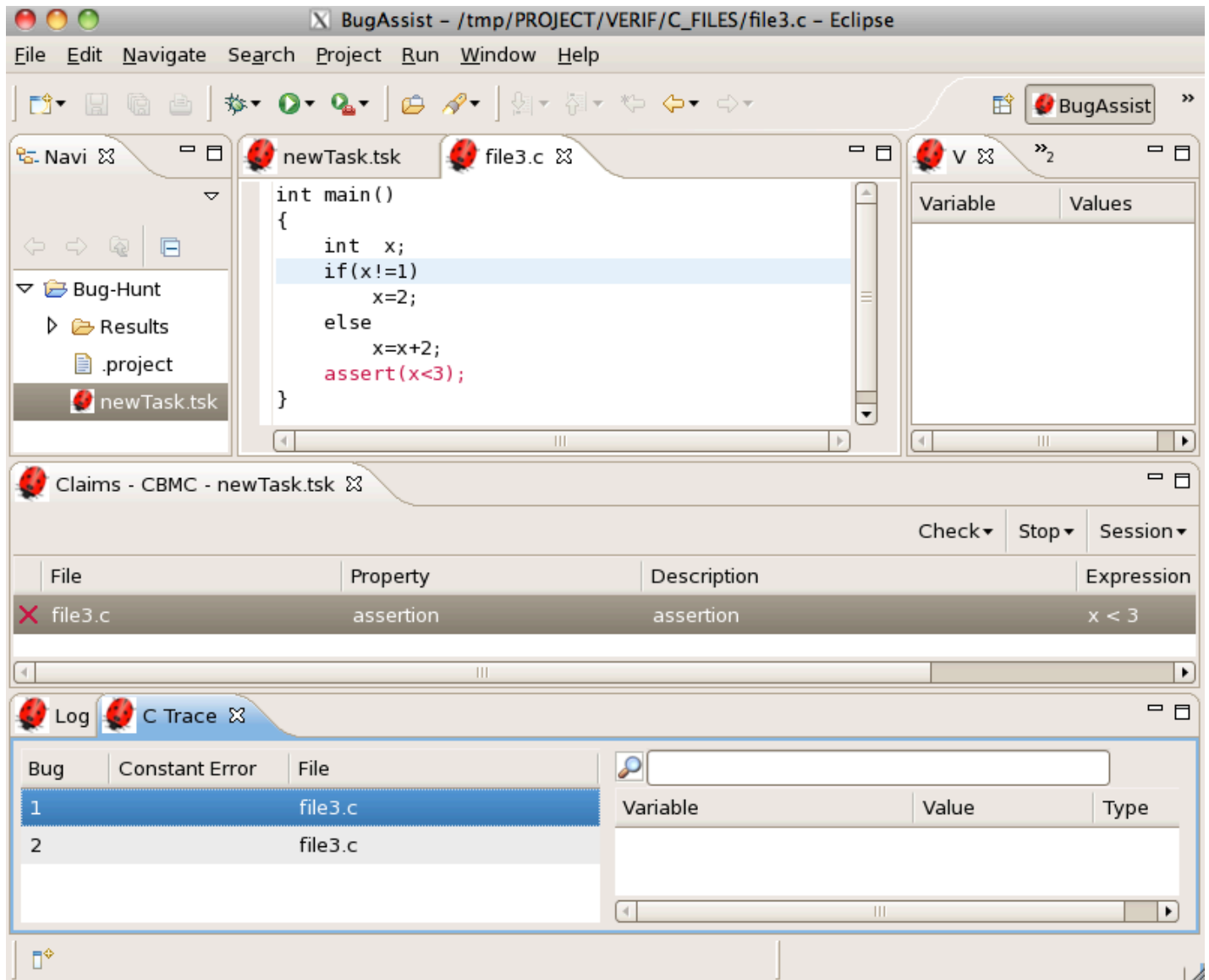
Finding Bugs in the selected C Files.

Select the assertion or list of assertions to check for from the Check menu as shown in the figure below. This runs the Bug-Assist Core algorithm and list the potential bug locations .



If any of the property is violated then the model checker returns failure and is represented using a RED "X" on the property.

The C Trace window displays the potential bug locations where a correction can be made to fix the program. Selecting a bug highlights the corresponding line number in blue in the editor.



User can fix the bug and run the model checker again to verify the correctness again or to find other bugs and error locations.