Commit-misra

Change History

Date	Change	Author
2018.08.06	Initial release	Bruce Eitman

Background

Coding standards rules (i.e. MISRA) create a large volume of findings in Coverity. The high volume is difficult to manage, report on, and increases database size. To combat this, customers would like to focus on only 1st party code and ignore defects from code they do not control – such as system header files, 3rd party libraries, etc.

Coverity does provide components to help manage data, but they only filter defects after the fact without any way to exclude all together. Components will not reduce database size, or help with MISRA report.

Commit-misra is a java applet that can be used to reduce the number of findings, before adding them to Coverity server. It can be configured in 1 of 2 ways:

- REMOVE defects found in specific source files & folders, or
- KEEP defects found in specific source folders and REMOVE all others.

Usage

```
commit_misra [-a] [-c <Config Filename>] [-d <Intermediate Folder>]

     [-D] [-t] [-v] <cov-commit-defects arguments>
```

Options

`-a,--all-finding`

All findings instead of MISRA only findings. Default is to only process and modify MISRA results.

`-c <Config Filename>`

The input config Json file.

`-d,--dir <Intermediate Folder>`

The intermediate folder as documented for cov-commit-defects.

`-D`

Debug option turns on extra output that can be useful for debugging.

`-t,--temp-folder`

Use a temporary folder. The intermediate folder will be copied to a temporary folder used for processing and committing. This leaves the original intermediate folder available to be processed differently and commuted again.

`-v,--version`

Display Version

`<cov-commit-defects arguments>`

All cov-commit-defects arguments must be added after the parameters above. The intermediate directory does not need to be, and must not be, include on the command line twice.

Example Command Line

```
commit-misra
      --dir iDir
      -c file.config
      -t
      --host localhost
      --user admin
      --password 123456
      --stream MyStream
```

Which shows the arguments host, user, password and stream which will be passed on to cov-commit-defects.

Config File

The configuration file is a Json formatted file with the following arrays of objects:

skipfiles	A list of files to remove from the analysis findings.

skipfolders	A list of folders to remove from the analysis findings.

keepfolders	A list of folders to keep in the analysis findings, all non-matching folders will be removed.

Note*: Generally, only one of “skip” or “keep” options should be used. However, you can combine for more advanced cases such as keeping a folder but removing (skipping) some sub folders.*

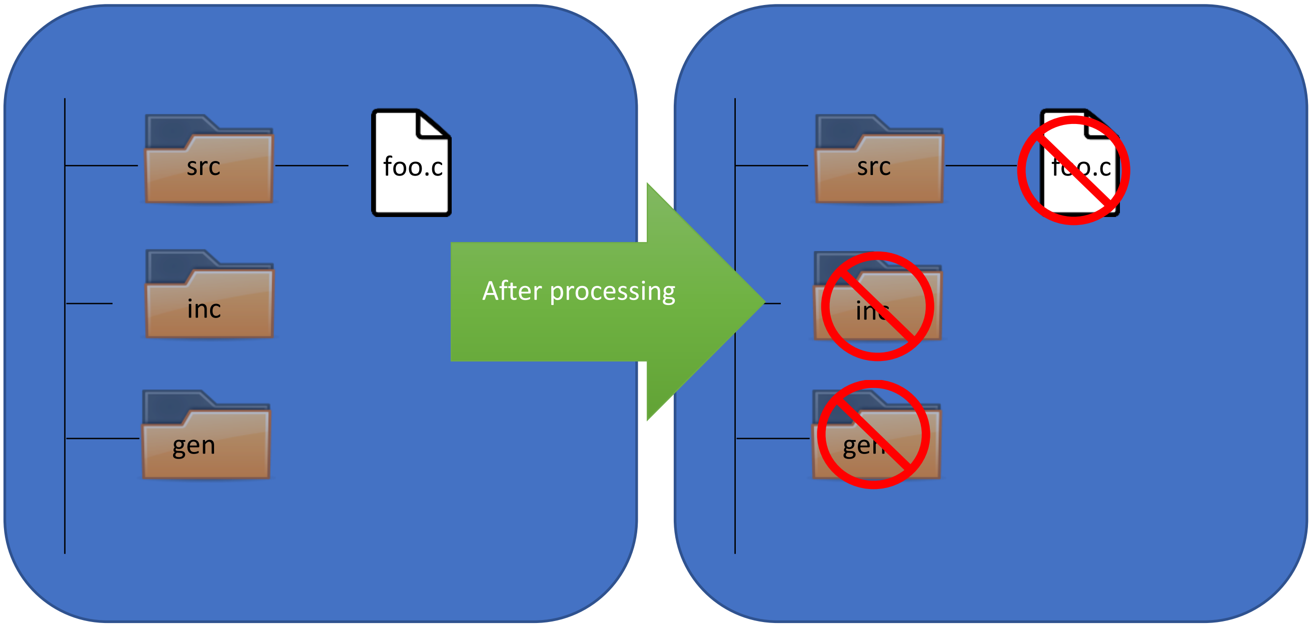
Example config file:

```
      {
        "skipfiles" : [
                 { "name" : "foo.c" }
           ],
        "skipfolders" : [
                 { "name" : "gen" },
                 { "name" : "inc" }
           ],
      }
```

Example #1: Remove defects found in folders or files

In this example using the config file above, all defects found under the following folders are removed from the result set before committing.

- inc
- gen
- foo.c (an individual source file)



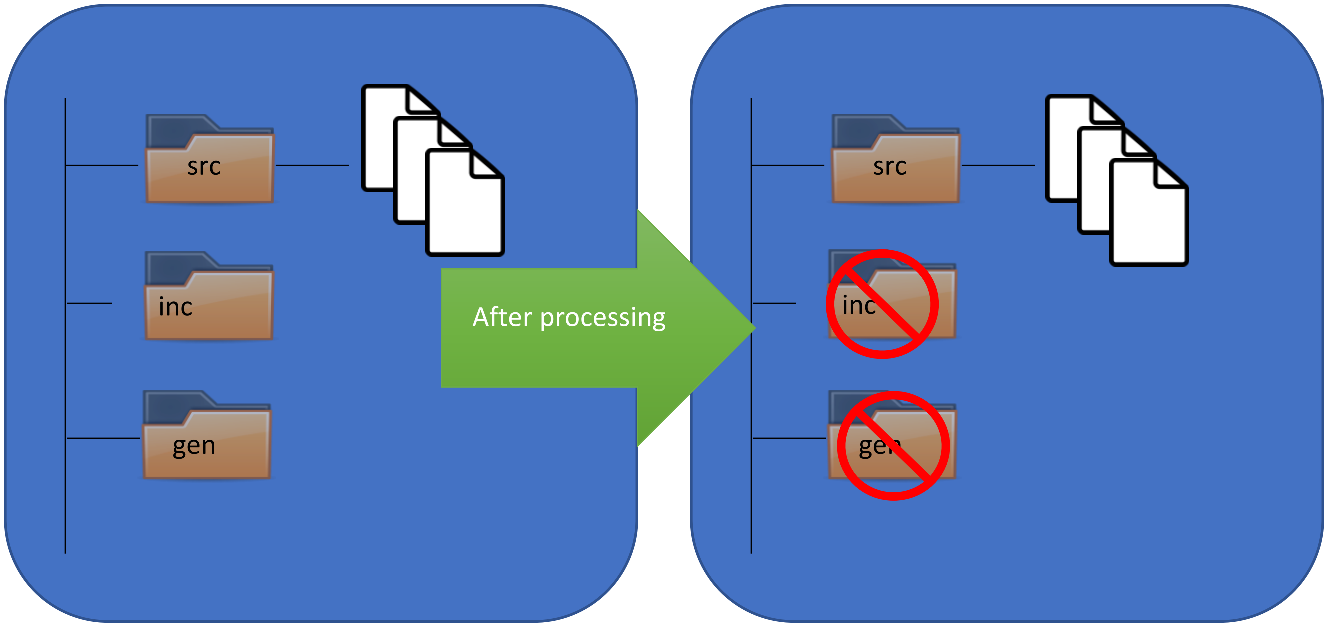
This is because "inc" and "gen" are in the skipfolders list, and "foo.c" is in the skipfiles list. Everything else under "src" is kept.

Example #2: Keep defects found in folders, remove everything else

In this example using the config file below, all defects from files under /inc & /gen are removed from the results before committing. This is because src is in the “keepfolders” list. Therefore, folders outside of src are removed from the results.

Config JSON (“file.config”)

```
{
  "keepfolders" : [
          { "name": "src" }
  ]
}
```


src
inc
gen
After processing
src
inc
gen