

```

1  #!/bin/sh
2  #
3  # An example hook script to verify what is about to be committed.
4  # Called by "git commit" with no arguments. The hook should
5  # exit with non-zero status after issuing an appropriate message if
6  # it wants to stop the commit.
7  #
8  #
9  # To enable this hook, rename this file to "pre-commit".
10
11 if git rev-parse --verify HEAD >/dev/null 2>&1
12 then
13     against=HEAD
14 else
15     # Initial commit: diff against an empty tree object
16     against=$(git hash-object -t tree /dev/null)
17 fi
18
19 # If you want to allow non-ASCII filenames set this variable to true.
20 allownonascii=$(git config --type=bool hooks.allownonascii)
21
22 # Redirect output to stderr.
23 exec 1>&2
24
25 # Cross platform projects tend to avoid non-ASCII filenames; prevent
26 # them from being added to the repository. We exploit the fact that the
27 # printable range starts at the space character and ends with tilde.
28
29
30 if [ "$allownonascii" != "true" ] &&
31     # Note that the use of brackets around a tr range is ok here, (it's
32     # even required, for portability to Solaris 10's /usr/bin/tr), since
33     # the square bracket bytes happen to fall in the designated range.
34     test $(git diff --cached --name-only --diff-filter=A -z $against |
35         LC_ALL=C tr -d '[ -~]\0' | wc -c) != 0
36 then
37     cat <<EOF
38 Error: Attempt to add a non-ASCII file name.
39 This can cause problems if you want to work with people on other platforms.
40 To be portable it is advisable to rename the file.
41 If you know what you are doing you can disable this check using:
42     git config hooks.allownonascii true
43 EOF
44     exit 1
45 fi
46
47

```

```

48 # ---
49 # CUSTOM PART STARTS HERE
50 # ---
51
52 # Date
53 now=$(date +"%m_%d_%H00")
54
55 mkdir /home/fabian/thesis/results/metrics
56 filename=/home/fabian/thesis/results/metrics/log_${now}
57
58 # Break if Report already exists
59 if [ -f "${filename}.txt" ]; then return; fi
60
61
62 # Exact date as a header
63 date >> ${filename}.txt
64
65 echo "\n--- APP -----" >> ${filename}.txt
66
67 # Add Metrics to File
68 echo "\n---\nComplexity\n---" >> ${filename}.txt
69 echo "\nCyclic Complexity:" >> ${filename}.txt
70 radon cc app.py -a | grep complexity >> ${filename}.txt
71 echo "\nHalstead:" >> ${filename}.txt
72 radon hal app.py >> ${filename}.txt
73 echo "\nMaintainability Index:" >> ${filename}.txt
74 radon mi app.py -s >> ${filename}.txt
75 echo "\n---\nReadability\n---\n" >> ${filename}.txt
76
77 echo "Pylint Rating:" >> ${filename}.txt
78 echo $(python3 -m pylint app.py | grep rated) >> ${filename}.txt
79
80 echo "\n--- Hardware/generic -----" >> ${filename}.txt
81
82 # Add Metrics to File
83 echo "\n---\nComplexity\n---" >> ${filename}.txt
84 echo "\nCyclic Complexity:" >> ${filename}.txt
85 radon cc hardware/generic -a | grep complexity >> ${filename}.txt
86 echo "\nMaintainability Index:" >> ${filename}.txt
87 radon mi hardware/generic -s >> ${filename}.txt
88 echo "\n---\nReadability\n---\n" >> ${filename}.txt
89
90 echo "Pylint Rating:" >> ${filename}.txt
91 echo $(python3 -m pylint hardware/generic/ | grep rated) >> ${filename}.txt

```