Coverage.py Python code coverage

CONCEPTS

Two steps: run records execution into a .coverage data file, then report, html, xml, json or lcov read it. Reports never run your code again. Without --append each run overwrites the data.

coverage run your tests or script
↓ writes
.coverage SQLite data file
↓ read by
report / html / xml json, lcov, annotate
↓ for
Humans and CI --fail-under gate, upload

MEASURE

coverage run *<script.py>* **[args]**
Run a script under measurement

coverage run -m *<module>* **[args]**
Run a module, like python -m

coverage run -m pytest
Measure the test suite

coverage run --branch -m pytest
Count branches, not just lines

coverage run --source=*<dir-or-pkg>* **...**
Limit measuring to this code

coverage run --omit=*<pat>* **...**
Skip files matching a pattern

coverage run -a ...
Append to the existing data

coverage run -p ...
Separate data file per process

REPORT

coverage report
Text table of coverage per file

coverage report -m
Add the missing line numbers

coverage report --skip-covered
Hide files at 100%

REPORT (cont.)

coverage report --fail-under=*<n>*
Exit 2 if total is below n

coverage report --format=markdown
Markdown table (or text, total)

HTML AND OTHER FORMATS

coverage html
Write the HTML report to htmlcov/

coverage html -d *<dir>*
Choose the output folder

coverage xml -o *<file>*
Cobertura XML (default coverage.xml)

DATA FILES

coverage combine
Merge .coverage.* files into .coverage

coverage erase
Delete the collected data

EXCLUDING CODE

 # pragma: no cover
Exclude this line, or the block it opens

 # pragma: no branch
Don't count a missing branch here

exclude_also = [...]
Add regexes to the default exclusions

[run] omit = [...]
Config: files never measured

[report] fail_under = *<n>*
Config: default threshold

[run] branch = true
Config: always measure branches

[run] source = ["*<pkg>***"]**
Config: always limit to your code

PARALLEL AND CI

[run] parallel = true
Config: unique data file per process