

Testing checklist: desktop application

Source: <https://howtotestaproduct.com/how-to-test/desktop-app/>

Generated: 2026-10-07. Licence: CC BY 4.0.

☐ 1. release smoke tests

Done when: The signed installer opens on clean macOS and Windows machines with no Gatekeeper or SmartScreen warning, and lists the same files and services as the previous release plus documented changes.

Risks: store-rejection, data-breach. Tool class: device cloud. Usually run by: QA engineer.

☐ 2. penetration testing

Done when: A hostile web page opened in each supported browser cannot launch the app, join a session or read app data through a local port or URL handler.

Risks: data-breach, regulatory-penalty. Tool class: proxy. Usually run by: security specialist.

☐ 3. compliance checks

Done when: Each build passes notarization and Microsoft Store certification, and release notes name every helper, service or login item the update installs.

Risks: store-rejection, regulatory-penalty. Tool class: test management. Usually run by: product manager.

☐ 4. end-to-end functional tests

Done when: After install, update and uninstall on a test machine with redirected Documents and a moved data folder, every user file is still present and no helper remains.

Risks: data-corruption, data-breach. Tool class: device cloud. Usually run by: QA engineer.

☐ 5. regression suite

Done when: Install, update, uninstall and local-port cases rerun before each release, and any new listening port or deleted user file blocks it.

Risks: data-corruption, data-breach. Tool class: test management. Usually run by: QA engineer.

☐ 6. device and browser compatibility

Done when: Install, update and uninstall cases pass with no Gatekeeper or SmartScreen warning on each supported macOS, Windows and Linux version and on both Intel and Arm machines where offered.

Risks: store-rejection, data-corruption. Tool class: device cloud. Usually run by: QA engineer.

☐ 7. unit tests

Done when: In the test report, the cleanup and repair code refuses to delete the home folder, the root folder, an empty path or any folder the app did not create.

Risks: data-corruption. Tool class: unit framework. Usually run by: developer.

☐ 8. security scanning

Done when: A port and process scan of a test machine after install shows no listening service beyond those in the docs, and the bundled runtime has no known critical advisory.

Risks: data-breach. Tool class: DAST scanner. Usually run by: security specialist.

☐ 9. data integrity and migration testing

Done when: A test profile with known files keeps the same file list and contents before and after each update, including files saved in redirected folders.

Risks: data-corruption. Tool class: data validation. Usually run by: QA engineer.