

Environment Controls for Accessibility-Tree Node-Count Benchmarks

A two-host Chrome DevTools Protocol field study

Exzil Calanza

Independent Researcher and Software Builder, Iloilo City, Philippines

23 August 2026 | Version 1.0

Publication status. Independent research preprint. This work has not been accepted, certified, or peer reviewed by a standards body, publisher, or the protocol vendors cited.

Reproducibility record. [skynet-chrome-cdp commit d19cd0222797](#)

Abstract

Raw accessibility-tree node totals are often reported as if they were stable properties of a page or instrument. This field study shows why that interpretation is unsafe without environment controls. Two Windows hosts running the same Chrome 151.0.7922.140 build against the same deterministic fixture initially returned 1,684 and 1,244 nodes, while DOM characters, actionable-element count, and extracted scene characters agreed. A controlled viewport experiment on one host then changed the raw total from 1,684 to 1,874 solely by narrowing the viewport from 1,280 to 800 CSS pixels. The 190-node increase was exactly the increase in `InlineTextBox` nodes. Subtracting `InlineTextBox` nodes yielded 1,244 at every tested width. These observations establish viewport dependence within the controlled host and support reporting a semantic candidate count separately from layout-derived nodes. They do not prove why the first 1,244-node reading occurred on the second host because that run lacks a role histogram. The paper therefore proposes a field-level comparability contract: pin and read back viewport state; record browser, protocol, fixture, and accessibility conditions; report role histograms; separate semantic and layout-derived counts; and scope each cross-machine claim to the measurements that actually reproduced.

Keywords: accessibility tree; Chrome DevTools Protocol; browser instrumentation; benchmark comparability; `InlineTextBox`; reproducible evaluation

1. Research question and contribution

Browser automation systems increasingly use the accessibility tree as a semantic perception layer. The Chrome DevTools Protocol (CDP) Accessibility domain exposes roles, names, states, and relationships that are closer to user-action semantics than raw markup [1]. That convenience creates a measurement risk: a raw node total can look like a property of the page even when part of the tree is produced by layout and browser accessibility state.

This study asks a narrow question: under what recorded conditions can accessibility-tree node counts be compared across machines? Its contribution is empirical and procedural. It reports one two-host observation, one controlled viewport experiment, and a benchmark contract that distinguishes measured invariance from assumed portability. It does not claim that the proposed semantic count is universally invariant or that the experiment isolates an operating-system or hardware cause.

2. Measurement design

The benchmark creates its fixture inside a tab that it owns. The page contains 685 total HTML elements: 200 fixed-copy paragraphs and 120 form rows, each containing one label, one textbox, and one button. The actionable subset is therefore 240 controls: 120 textboxes plus 120 buttons. The fixture contains no randomness, wall-clock input, or network content. The exact generator is preserved in the commit identified in the reproducibility record [4][6].

The original cross-host run used two physical Windows machines. Both reported Chrome 151.0.7922.140, V8 15.1.206.17, and CDP 1.3. Host A was Windows 11 with six logical CPUs and Python 3.13.7; Host B was Windows 10 with four logical CPUs and Python 3.14.7. The browser build was controlled, but operating system, CPU count, Python version, window state, and active accessibility mode were not all held constant. This is an observational comparison, not a causal hardware benchmark.

The controlled follow-up used CDP `Emulation.setDeviceMetricsOverride` at widths of 800, 1,280, 1,920, and 2,560 CSS pixels, with height fixed at 900 pixels and `deviceScaleFactor` fixed at 1 [2]. The harness read `window.innerWidth` and `window.innerHeight` after each override before collecting `Accessibility.getFullAXTree`. The owned tab was closed at the end of the run.

Table 1. Recorded host conditions. The run is observational because several host variables differ.

Condition	Host A	Host B
Machine	Zeke	DESKTOP-OJ9K9QR
Operating system	Windows 11 (10.0.26200)	Windows 10 (10.0.19045)
Logical CPUs	6	4
Python	3.13.7	3.14.7
Chrome	151.0.7922.140	151.0.7922.140
V8	15.1.206.17	15.1.206.17
CDP	1.3	1.3

3. Cross-host observation

The two hosts agreed on the page-derived measurements that were independent of the raw tree total: 55,945 DOM characters, 240 actionable elements, 4,579 extracted scene characters, and a 12.22x DOM-to-scene reduction. They did not agree on raw accessibility-tree nodes or screenshot payload size.

Host A returned 1,684 raw nodes. Host B's first run returned 1,244. The absolute gap is 440 nodes; relative to Host A, Host B's first total is 26.13% lower. That arithmetic is measured. The role composition of Host B's first tree is not: the run stored only the bare total.

A later Host B run reported TOTAL 1684, INLINETEXTBOX 440, and SEMANTIC 1244 at an inner window of 929 x 925. That later observation supports the stability of the 1,244 semantic candidate across the two tested hosts, but it cannot retroactively identify which roles were absent from Host B's first 1,244-node tree.

Table 2. Original cross-host fixture measurements.

Measurement	Host A	Host B	Comparability statement
DOM characters	55,945	55,945	Matched on both tested hosts
Raw AX nodes	1,684	1,244	Not comparable; Host B role histogram absent
Actionable controls	240	240	Matched on both tested hosts
Scene characters	4,579	4,579	Matched on both tested hosts
DOM / scene	12.22x	12.22x	Matched on both tested hosts
Screenshot bytes	119,683	109,932	Not comparable; viewport and pixels unpinned

4. Controlled viewport results

Narrowing Host A from 1,280 to 800 CSS pixels increased the raw node total by 190, from 1,684 to 1,874, while DOM characters and actionable nodes did not move. `InlineTextBox` nodes increased by exactly the same 190, from 440 to 630. The semantic candidate, defined here as raw nodes minus `InlineTextBox` nodes, remained 1,244 at all four widths.

This establishes a controlled relationship within the tested host and fixture: the raw count changed with viewport width, and the observed change was fully accounted for by `InlineTextBox`. It does not establish that every raw-count change in every browser is caused by the same role, nor that subtraction is a standards-defined universal semantic metric.

Table 3. Controlled viewport experiment on Host A.

Viewport	Raw AX	InlineTextBox	Raw - Inline	Actionable	DOM chars
800 x 900	1,874	630	1,244	240	55,945
1,280 x 900	1,684	440	1,244	240	55,945
1,920 x 900	1,684	440	1,244	240	55,945
2,560 x 900	1,684	440	1,244	240	55,945

Viewport width changes the raw tree, not the semantic candidate

Chrome 151.0.7922.140 on Host A; fixed fixture and 900 px height

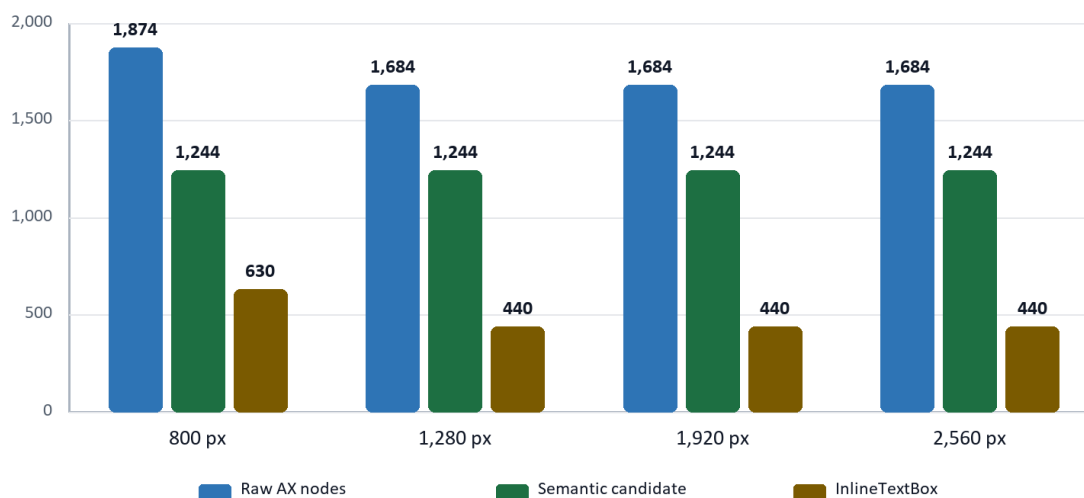


Figure 1. The 800 px raw-total increase is exactly the `InlineTextBox` increase; raw minus `InlineTextBox` remains 1,244. Source: this study's frozen viewport probe.

5. Interpreting the 440-node gap

The later role-aware measurements make absent `InlineTextBox` nodes a plausible explanation for Host B's first 1,244 total, but the original run cannot prove it. In the measured role histogram, three disjoint role sets each happen to total 440: `InlineTextBox`; `StaticText`; and generic plus paragraph. Subtracting any one of those sets from 1,684 yields 1,244. Arithmetic alone therefore does not identify the missing population.

Viewport width also does not reproduce the original 1,244 raw total in the four-width follow-up. The `InlineTextBox` count floors at 440 between 1,280 and 2,560 pixels and rises at 800 pixels. This favors a difference in accessibility state or consumer behavior over ordinary reflow, but that remains an inference because the first Host B tree was never enumerated by role.

The correct public claim is therefore asymmetric. Viewport dependence of the raw count is measured. The cause of the original cross-host total is unknown. The semantic candidate reproduced in the tested conditions, but universal machine invariance is unproven.

6. Proposed comparability contract

A benchmark should treat comparability as a property of individual fields, not as a blanket boolean for an entire report. The minimum record below is designed to make a second operator capable of deciding which values can be compared and which remain environment readings.

Table 4. Minimum evidence contract for cross-machine accessibility-tree benchmarks.

Record	Required evidence	Failure state
Fixture	Generator commit, constants, element count, content hash	Unknown fixture equivalence
Browser	Chrome product/build, V8, CDP version	Unknown engine equivalence
Viewport	Requested metrics plus innerWidth/innerHeight read-back	Raw count not comparable
Accessibility	Role histogram, ignored nodes, concurrent consumer/mode notes	Cause of count gap unknown
Node metrics	Raw, InlineTextBox, explicit semantic formula, actionable roles	Do not collapse to one ax_nodes field
Run protocol	Warm-ups, samples, settle interval, owned-tab ID	Timing and state not reproducible
Cleanup	Owned tab closed, no operator tab resized	Run hygiene unverified

7. Threats to validity and claim boundary

External validity is limited. The study covers two Windows hosts, one Chrome build, one deterministic fixture, four controlled widths on one host, and one later role-aware reading on the second host. It does not test macOS, Linux, other browsers, other Chrome versions, zoom, device scale factors other than 1, assistive-technology combinations, or arbitrary production pages.

Construct validity is also bounded. The semantic candidate removes InlineTextBox because that role was the layout-derived population observed here. CDP does not define ax_nodes_semantic as a standard metric. A future browser version or page class may contain other layout-sensitive roles. Role histograms and explicit definitions must accompany the number.

The original cross-host run did not pin or record all window and accessibility conditions. It also did not perform a pixel comparison of screenshots; an 8% difference in PNG bytes is therefore only a payload observation, not proof of a visual difference. Timing differences cannot be attributed to CPU count, operating system, tree size, or Python version because those variables changed together.

This manuscript is an independent research preprint. It has not been accepted, published, certified, or peer reviewed by IEEE, ACM, W3C, Google, or another standards body or publisher. WAI-ARIA and CDP are cited as primary technical references, not as endorsements of this metric [1][2][3].

8. Reproducibility protocol

Use the exact benchmark commit recorded on the cover and in Appendix B. Run only against a tab created by the harness. Record browser product, V8, protocol version, operating system, Python version, fixture constants, requested viewport, read-back inner dimensions, role histogram, ignored count, actionable count, and cleanup result. Preserve raw JSON rather than only a prose summary.

For every cross-machine comparison, report raw nodes, `InlineTextBox` nodes, and the explicitly defined semantic candidate side by side. If the role histogram or viewport read-back is absent, mark raw node comparability unknown. Do not infer a role population from a total alone. The benchmark implementation, cross-machine report, and source repository are linked in the references [4][5][6].

9. Conclusion

A raw accessibility-tree total is not portable by default. In this experiment it moved by 190 nodes when only viewport width changed, and the movement was exactly the `InlineTextBox` delta. The two-host record also shows why an unexplained total must remain unexplained: the original 440-node gap cannot be assigned to a role that was never captured.

The practical remedy is a better measurement contract, not a more confident number. Pin and verify the environment, preserve role-level evidence, separate layout-derived and semantic candidates, and state comparability field by field. Under that contract, unknown remains unknown, and a repeated value earns a scoped claim instead of an unearned universal one.

Appendix A. Role histogram at 1,280 px

The histogram below accounts for all 1,684 nodes measured on Host A at 1,280 x 900. The two actionable roles sum to 240. The two ignored nodes carry role none.

Role at 1,280 px	Count	Interpretation in this fixture
StaticText	440	Text-bearing semantic nodes
InlineTextBox	440	Layout-derived text runs
generic	240	Form-row wrappers
paragraph	200	Fixed-copy blocks
LabelText	120	Form labels
textbox	120	Actionable controls
button	120	Actionable controls
none	2	Ignored nodes
RootWebArea	1	Root
main	1	Fixture container

Appendix B. Frozen evidence manifest

The local evidence below was hashed before manuscript generation. GitHub sources were fetched from commit d19cd02227979b70596dbc440e06f1e08a52e659.

Evidence	Bytes	SHA-256
viewport_probe	3,143	7783ea62f718c9ddaa6a10bc7d2efcc5399c4678b8ec3bc2b25ffe751f67fcc9
benchmark_script	34,483	9a59078f237454452db678146825c5600f5b4bc1e79456019aafe595c507b2b1
cross_machine_report	7,847	0985193fe6b34ec1b2be248404b2b6dd82fddfeaa4bc8f4c2ad8532f98cb4188
repository_readme	9,874	dcd0109192d7f421e27634c6e534d5e9087902f69f6464cc787ce68b8d3f89be

References

- [1] [Chrome DevTools Protocol, Accessibility domain](#). Accessed 23 August 2026.
- [2] [Chrome DevTools Protocol, Emulation domain](#). Accessed 23 August 2026.
- [3] [W3C, Accessible Rich Internet Applications \(WAI-ARIA\) 1.2](#). Accessed 23 August 2026.
- [4] [Zek21, skynet-chrome-cdp repository](#). Accessed 23 August 2026.
- [5] [Cross-machine results at the frozen commit](#). Accessed 23 August 2026.
- [6] [Standalone benchmark v2.0.0 at the frozen commit](#). Accessed 23 August 2026.
- [7] [Reader-facing article and live research-package record](#). Accessed 23 August 2026.

Author note

Exzil Calanza is an independent researcher and software builder in Iloilo City, Philippines. Research engineering, artifact assembly, and guarded publication were performed through Skynet under his direction. [Open the companion Platform article](#).