Symbolic Mechanics

Prototype v0.2.1

Test and Validation Report

Version 1.0

Implementation test and validation record for the v0.2.1 extension layer; not a full Human OS validation and not a total-system validation report.

1 P1 - What Was Tested

This validation round tested the v0.2.1 extension layer.

The preserved v0.1-B1 base core remained the supporting foundation for execution, but this report does not revalidate the full preserved base spine as a total object of review.

The test target of this round was narrower and more specific: to determine whether the frozen v0.2.1 extension layer could operate in a stable manner within its approved package boundary when executed over the preserved base core.

This report is therefore an extension-layer test and validation record. It is not a full-system validation document, not a full Human OS validation, and not a total completion claim for the broader Symbolic Mechanics system.

2 P2 - Test Boundary

This test round was restricted to the frozen v0.2.1 extraction boundary only.

Within that boundary, the executable and validation target covers only the v0.2.1 extension package derived from Volumes 21–35.

Volumes 36 onward were not part of the tested extraction layer. They were permitted only for boundary checking and exclusion control. They were not valid source material for executable schema, transition rules, trace requirements, conformance logic, or package-level claims in this validation round.

The following later-topology material remained excluded throughout the test process:

- family-table material
- lineage material
- birth-order material
- role-allocation material

Accordingly, the pass/fail status of this round applies only to bounded v0.2.1 extension-layer implementation stability inside the frozen extraction line. It does not constitute a judgment on full-system completion.

3 P3 - Test Corpus / Trace Set

The implementation pass used a frozen test corpus and trace set.

The required trace family for this round contained 12 extension traces. These traces were not designed as a display of full theoretical richness. Their function was narrower and more disciplined: to test whether the v0.2.1 extension layer could execute in a stable, repeatable, and structurally controlled manner inside the frozen package boundary.

The trace set therefore served three purposes. First, it provided bounded coverage of the approved extension-band mechanics. Second, it imposed execution discipline by requiring the same protected distinctions to remain visible across the run. Third, it allowed stability checking under a fixed implementation-facing corpus rather than under open-ended interpretation.

This trace family remained subordinate to the boundary rules, the exclusion checklist, and the protected-distinction requirements of the frozen package.

4 P4 - Execution Protocol

The implementation protocol for v0.2.1 was defined as a controlled execution-layer procedure rather than as open-ended interpretive testing.

Runner classes were distinguished at the protocol level so that manual, rule-based, toy-simulator, and later-engine implementations could be recognized as different execution styles while remaining subordinate to the same frozen package, trace family, and protected law structure.

Each trace was required to pass through a fixed execution sequence. The trace had to be loaded from the frozen corpus, checked first through the exclusion and validator layer, executed through the approved extension-layer run procedure if still eligible, reviewed through the conformance layer, and then stored together with its execution artifacts.

Output requirements were fixed at trace level. Each executed trace was required to preserve a trace input record, an execution record, a validator record, a conformance record, and a final trace-level judgment in stable retrievable form.

Pass/fail taxonomy was also fixed in advance. Trace results were classified through bounded status categories for stable execution, bounded drift, instability, or exclusion-triggered invalidation. Exclusion-rule checking therefore preceded ordinary conformance review and could disqualify a trace before further validation if contamination or protected-distinction collapse was detected.

Artifact retention was part of the protocol itself. Each run was required to preserve a stable audit trail linking the frozen trace input, execution output, validator result, conformance result, and final judgment so that later comparison and review could be performed without reconstructive guesswork.

5 P5 - Results Summary

The first implementation round for Prototype v0.2.1 produced the following batch result.

12 extension traces were run.

12 were judged operationally stable.

0 were judged unstable.

0 were classified as invalid_by_exclusion_rule.

Batch result = Implementation-Stable.

This result indicates that the frozen v0.2.1 extension trace family completed its first implementation pass without exclusion-triggered invalidation and without trace-level instability inside the approved package boundary. The result applies to bounded extension-layer implementation stability only and does not constitute a claim of full Human OS success or total-system completion.

Prototype Timeline

Base core, additive extension, and implementation validation.

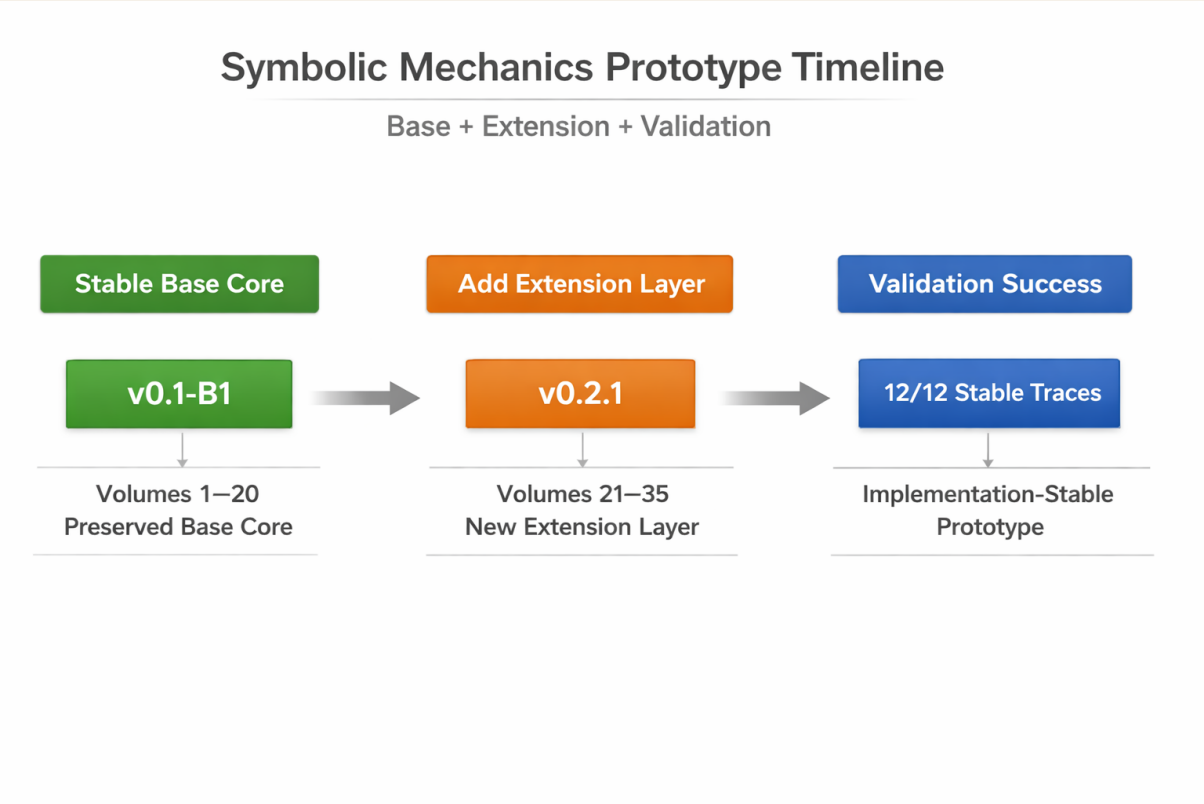


Figure 1. Prototype line progression from preserved base core to extension layer and validation result.

6 P6 - Interpretation of Result

This pass confirms that the frozen v0.2.1 extension layer succeeded in its first-round implementation testing within the approved prototype boundary.

Within this prototype line, implementation-stable means that the extension-layer package could be executed in a stable, bounded, and validator-disciplined manner over the preserved base core without protected-distinction collapse and without exclusion-triggered invalidation across the required trace family.

The result therefore has a specific and limited meaning. It establishes first-round execution stability for the v0.2.1 extension layer inside the frozen package boundary only.

It does not mean full Human OS completion.

It does not mean total-system completion, and it does not mean full implementation of the broader 80+ volume Symbolic Mechanics archive.

The correct interpretation is therefore narrow and exact: the frozen v0.2.1 extension layer has successfully crossed its first implementation-layer stability threshold within prototype scope.

7 P7 - Artifact Retention / What Was K

The implementation round preserved a complete archive-facing artifact set rather than only a final pass label.

The retained record included the governing test plan, the batch-level pass summary, the conformance-layer record, the exclusion-layer record, the retained trace corpus together with its resulting outputs, and the relevant implementation artifacts produced during execution. These materials were preserved as one linked validation set rather than as isolated documents.

This retention discipline serves a technical archive function. It keeps the test conditions, execution logic, boundary control, judgment pathway, and resulting outputs reconstructable after the run. It also prevents the validation result from being reduced to an unsupported conclusion without the materials required for later review, comparison, and audit.

What was kept, therefore, was not only the fact of passage, but the structured evidence of how the v0.2.1 extension layer was tested, how it remained within scope, how conformance was judged, and what execution-facing artifacts were produced and retained as part of the frozen record.

8 P8 - Bottom Line

Prototype v0.2.1 is no longer a speculative extension.

It is now the first implementation-stable additive extension layer over the preserved v0.1-B1 base core.

The test result establishes that this extension layer has crossed the threshold from proposed architecture to retained prototype-stage implementation. Within the frozen boundary of the approved package, it is now keepable, citable, and usable as a preserved extension-layer foundation for later controlled work.

At the same time, its status remains bounded and must be recorded accurately.

Prototype v0.2.1 is not a full-system endpoint. It is a preserved and implementation-stable extension milestone within prototype scope, not a claim of total completion.