

---

# Epiphany

REFERENCE SUITE

*A companion to the Core Specification*

---

Version 0.1.0 — Phase 3 (the initial entry set: six scores, required at Minimal, declared for Standard)

Normative for solver conformance claims

# Contents

---

|          |                                                  |           |
|----------|--------------------------------------------------|-----------|
| <b>1</b> | <b>About This Companion</b>                      | <b>2</b>  |
| 1.1      | Relationship to the Core Specification . . . . . | 2         |
| 1.2      | Conformance . . . . .                            | 3         |
| <b>2</b> | <b>The Suite Entry Model</b>                     | <b>4</b>  |
| 2.1      | Score Referencing . . . . .                      | 4         |
| 2.2      | Solve Configuration . . . . .                    | 5         |
| 2.3      | Passing an Entry, Passing the Suite . . . . .    | 5         |
| 2.4      | Fixed-Expectation Tests . . . . .                | 6         |
| 2.5      | Suite Versioning . . . . .                       | 7         |
| <b>3</b> | <b>The vo.1 Entry Set</b>                        | <b>8</b>  |
| 3.1      | Overview . . . . .                               | 8         |
| 3.2      | RS-1: Ten-Measure Single Staff . . . . .         | 9         |
| 3.3      | RS-2: Rich Multi-Region Score . . . . .          | 9         |
| 3.4      | RS-3: B-flat Major Scale . . . . .               | 9         |
| 3.5      | RS-4: Two-Voice Counterpoint . . . . .           | 10        |
| 3.6      | RS-5: Notes and Rests . . . . .                  | 10        |
| 3.7      | RS-6: Three-Four Meter Line . . . . .            | 10        |
| <b>4</b> | <b>Harness Binding (Non-Normative)</b>           | <b>11</b> |
| <b>5</b> | <b>Revision History</b>                          | <b>12</b> |

# About This Companion

---

The *Reference Suite* is a companion to the Epiphany Core Specification. It fulfils the core specification’s delegation in its COMPANION SPECIFICATIONS appendix (the `sec:deferred:companions` section), which charters this document as “the collection of test scores against which solver conformance is established, with per-tier inclusion, per-tier metric thresholds, and any fixed-expectation tests. Versioned with this specification.”

This release (v0.1.0) delivers:

- the **suite entry model** — how entries name their test scores, how each solve is configured, what passing an entry and passing the suite mean, and how fixed-expectation tests work (Chapter 2);
- the **v0.1 entry set**: six scores, every one required for Minimal-tier conformance and the same six constituting the Standard-tier subset (Chapter 3);
- a non-normative note on the reference implementation’s suite harness (Chapter 4).

This document does *not* cover:

- the definition, normalization, and default thresholds of the quality metrics — those are the *Quality Metric Catalog* companion’s, and this document consumes them;
- performance conformance (edit traces, frame budgets) — the *Performance Reference Suite* companion’s, per the core specification’s explicit boundary;
- the reference solving algorithm — the non-normative *Reference Algorithm* companion’s.

## 1.1 Relationship to the Core Specification

The suite’s charter is the core specification’s Chapter 9 (THE CONSTRAINT SOLVER INTERFACE, the `ch:solver` chapter), CONFORMANCE: THE REFERENCE SUITE section (`sec:solver:conformance`). That section’s suite-entry requirement fixes what every entry consists of — a test score in canonical `.musc` form; per-tier inclusion; per-tier metric thresholds; and optional fixed-expectation tests, used sparingly — and fixes the pass rule: “A solver claiming a given tier **MUST** pass every entry required at that tier. Failure on any single entry is conformance failure at the claimed tier.”

Three more core anchors bind this document:

- The **Cross-Implementation Conformance** requirement (in the core’s solver-determinism section) enumerates the four conditions a conforming solver meets across the suite; this document’s per-entry evaluation rule (Section 2.3) is those conditions applied entry-by-entry.
- The **Reference Algorithm** section makes fixed-expectation tests the *only* place the suite may force a particular layout; everywhere else, any algorithm within thresholds conforms. Section 2.4 inherits that discipline.
- The *Quality Metric Catalog* companion defines every metric, its normalization, and the default per-tier thresholds this suite’s entries reference. This document never restates a threshold; it names the catalog’s defaults and records per-entry overrides (v0.1: none).

Where this document and a ratified core requirement disagree, **the core requirement governs** and the discrepancy is a defect in this document. One deliberate v0.1 deviation from the charter’s letter — naming scores by deterministic builder rather than shipping `.musc` bundles — is called out as such, with its open question, in Section 2.1.

## 1.2 Conformance

The entry set, entry construction rules, solve configurations, and tier inclusions in this document are **normative**. A solver conformance claim at a tier is a claim about *this* suite at *this* version:

- claiming tier  $T$  means passing every entry (Section 2.3) that Chapter 3 requires at tier  $T$  — failure on any single entry is conformance failure at the claimed tier;
- per the core specification, suite versions are tied to specification versions, and a conforming implementation **MUST** declare which suite version it passes (Section 2.5).

# The Suite Entry Model

## 2.1 Score Referencing

### REQUIREMENT

A v0.1 suite entry names its test score by **reference-implementation builder and seed**: a deterministic constructor exported by the reference implementation’s test kit (the `epiphany-testkit` crate), together with any seed argument, reproduces the score graph bit-for-bit. Two referencing forms are used:

- a **seeded builder**: a public function taking a `u64` seed (e.g. `fixtures::ten_measure_single_staff`, or a generator re-exported through the test kit), with the entry pinning the exact seed value;
- a **corpus name**: the `name` string of an entry of the test kit’s tagged corpus (`corpus()`), whose builder takes no arguments and is deterministic by construction.

The constructed score graph — not any serialized artifact of it — is the entry’s test score. Implementations under test **MUST** evaluate the entry against a score graph identical to the one the named builder produces with the named seed.

### RATIONALE

The core charter describes each entry as “a test score in canonical `.musc` form.” v0.1 deliberately references builders instead of shipping bundles, because today *the builders are the canonical definition*: they are versioned, reviewed, deterministic (seeded `SplitMix64` identity minting), asserted invariant-clean, and reproducible bit-for-bit by anyone building the reference crates — while the `.musc` byte format is still absorbing schema-major evolution, so a shipped bundle would rot faster than the builder that made it. A corpus caution: the test kit tags corpus fixtures with an *eligibility-taxonomy* tier (`Common / Edge / Torture`); that taxonomy is unrelated to solver conformance tiers (`Minimal / Standard / Advanced`) and carries no normative weight in this suite.

**OPEN QUESTION**

**Builder references versus shipped .musc bundles.** The charter’s letter — entries in canonical .musc form — is not met by v0.1: builder-plus-seed is implementation-anchored, which makes the suite awkward for an independent implementation that does not link the reference test kit (it must re-derive the score graphs from the builders’ sources). A future revision **MAY** ship canonical .musc bundles for every entry, exactly as the charter describes, once schema-major bytes are stable enough that shipped bundles do not rot; at that point builder references would remain as the bundles’ provenance record. Until then, the deviation is deliberate and this open question owns it.

## 2.2 Solve Configuration

A suite entry is only reproducible if the solve it prescribes is fully specified: the same score under a different page geometry casts off into different systems, and several metric axes are defined over the declared content bounds.

**REQUIREMENT**

Each suite entry declares the full solve configuration it is evaluated under:

- the **page geometry**: page size and margins, in staff spaces. In v0.1 every entry uses the reference implementation’s documented default — A4 portrait at an 8 mm staff height: page  $105 \times 148.5$  staff spaces, margins 7.5 staff spaces on all four sides, hence a content area of  $90 \times 133.5$  staff spaces. (The score graph has no home for page geometry yet: the core names `Canvas.layout_defaults` without defining a type, tracked as Pass-12 row P12-I7, so geometry is declared per entry as a solver parameter.)
- the **solver configuration**: the `SolverConfig` fields. In v0.1 every entry uses the default configuration — the `Standard` profile, an unbounded deterministic budget, and the Quality Metric Catalog’s default tie-breaking weights.

An implementation **MUST NOT** substitute its own defaults for a declared configuration when evaluating an entry.

## 2.3 Passing an Entry, Passing the Suite

**REQUIREMENT**

A solver **passes an entry at tier  $T$**  when, solving the entry’s score under the entry’s declared configuration, all four of the core specification’s cross-implementation conformance conditions hold for that solve:

1. every hard constraint of the solve is satisfied;
2. the result is internally deterministic per the core’s within-implementation rule (byte-identical `ResolvedLayoutIR` for repeated identical solves within one imple-

mentation version);

3. the `SolveReport` is well-formed and diagnostically accurate — in particular the metric vector is computed per the Quality Metric Catalog, never a placeholder;
4. every normative quality metric is at or below tier  $T$ 's threshold for its axis — the Quality Metric Catalog's default per-tier threshold table, unless the entry declares a per-entry override (no v0.1 entry does);

and additionally every fixed-expectation test the entry declares is reproduced exactly (no v0.1 entry declares any).

A solver **passes the suite at tier  $T$**  when it passes every entry required at tier  $T$ . Per the core specification, failure on any single entry is conformance failure at the claimed tier; a status of `Unsatisfiable` or a budget-exhausted partial solve on a suite entry is a failure of condition (1) or (3), not an exemption.

## 2.4 Fixed-Expectation Tests

The core specification permits an entry to pin specific layout properties — a particular bar's width, a specific system-break location — that all conforming solvers must reproduce, and directs that they be used sparingly. In the resolved layout such expectations are checkable against concrete structures: a bar's width against its `ResolvedMeasure` bounding box, a system count or break location against the `ResolvedSystem` list and the slots at which systems begin.

### REQUIREMENT

v0.1 declares **no fixed-expectation tests**. The mechanism is deliberately unused: every v0.1 entry is evaluated by validity, determinism, report accuracy, and thresholds alone. An entry of a future suite revision that adds a fixed expectation **MUST** state the expected value, the resolved structure it is checked against, and the exact comparison (including any tolerance).

### RATIONALE

A fixed expectation binds *every conforming solver* to one layout fact forever after; it is the only place the suite may force a layout, and none is warranted yet — no cross-implementation ambiguity has surfaced that thresholds fail to resolve. The reference implementation's golden files (SVG snapshots, byte-anchored layouts) are *implementation regression locks*: they pin what *that* implementation produced so its own drift is caught, and they are deliberately not suite conformance — promoting them to fixed expectations would freeze the ecosystem to the reference algorithm's choices, exactly what the core's threshold-based conformance model exists to avoid.

## 2.5 Suite Versioning

### REQUIREMENT

Per the core specification, suite versions are tied to specification versions: each release of this companion names the core-specification version whose conformance story it serves, and a revision of the core's solver chapter that changes tier obligations requires a corresponding suite revision. This v0.1.0 suite serves the pre-1.0 core working draft. A conforming implementation **MUST** declare which suite version it passes; a claim without a suite version is not a conformance claim.

This document keeps its own revision history (Chapter 5), independent of the Quality Metric Catalog's: entry-set growth and threshold-override changes are suite revisions, while metric definitions and default thresholds revise in the catalog.



# The vo.1 Entry Set

## 3.1 Overview

### REQUIREMENT

The vo.1 entry set is exactly the six entries of Table 3.1. Every entry is **required at the Minimal tier**. The **Standard-tier subset is the same six entries**, evaluated against the Quality Metric Catalog’s Standard threshold column. No entry declares a per-entry threshold override; no entry declares a fixed-expectation test. No entry is designated Advanced-only in vo.1 (the Advanced tier adds obligations, not entries, until an extension’s layout requirements enter the suite).

| Id   | Entry                    | Construction                                                                                                                     |
|------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| RS-1 | Ten-measure single staff | <code>fixtures::ten_measure_single_staff</code><br><code>with seed 0x000A_11CE</code>                                            |
| RS-2 | Rich multi-region score  | <code>generators::valid_score_rich</code><br><code>with seed 0xF302 (= corpus entry</code><br><code>gen_valid_score_rich)</code> |
| RS-3 | B-flat major scale       | <code>corpus entry b_flat_major_scale</code>                                                                                     |
| RS-4 | Two-voice counterpoint   | <code>corpus entry two_voice_counterpoint</code>                                                                                 |
| RS-5 | Notes and rests          | <code>corpus entry notes_and_rests</code>                                                                                        |
| RS-6 | Three-four meter line    | <code>corpus entry meter_three_four</code>                                                                                       |

Table 3.1: The vo.1 entry set. All six entries are required at Minimal; the same six constitute the Standard subset. Solve configuration for every entry: the declared default geometry and solver configuration of Requirement 2.2.

No implementation claims Standard-tier conformance as of this suite version; the Standard listing exists so that the first Standard claim is made against a pre-declared bar rather than a bar negotiated after the fact. Where an entry’s layout lacks a metric’s geometry class (a single system, a single page, no drawn slurs or beams), the affected axes evaluate to 0.0 under the catalog’s vacuous-geometry rule and the entry’s threshold on those axes is trivially met; each entry’s coverage note below says which axes it exercises non-degenerately.

## 3.2 RS-1: Ten-Measure Single Staff

**Construction.** `epiphany_testkit::fixtures::ten_measure_single_staff(0x000A_11CE)` — the seed the reference implementation’s engraving and rendering acceptance goldens document.

**Content.** A 10-measure, single-staff, single-voice metric score: 40 quarter notes (four per measure, all C<sub>4</sub>), plus a tie, a spanner, a marker, and a chord symbol. Invariant-clean by construction.

**Coverage.** The multi-system workhorse. Under the declared geometry the spaced line (about 99 staff spaces) exceeds the content width of 90 staff spaces and casts off into **two systems**, chosen among the barline break candidates — so the `SystemBreak`, `CastingOff`, and `SymbolDensity` axes are exercised non-degenerately, alongside `Collision` and `Spacing` (axis names per the catalog’s `QualityMetricKind` mapping). The cross-cutting objects (tie, spanner, marker, chord symbol) ride through the projection. Single page: the `PageFill` axis degenerates.

**Tiers, thresholds, expectations.** Required at Minimal; in the Standard subset. Catalog default thresholds; no overrides; no fixed expectations.

## 3.3 RS-2: Rich Multi-Region Score

**Construction.** `generators::valid_score_rich(0xF302)` (the core crate’s generator), identically reachable as the test-kit corpus entry `gen_valid_score_rich` (the corpus pins the same seed).

**Content.** Three *concurrent* regions on disjoint staves: a metric region (measures, an eighth-note triplet tuplet, a tie, a spanner, a marker, a chord symbol, a decomposition attachment), a proportional region (wall-clock events), and an aleatoric region (musical-discipline events) — plus tombstoned pitch and event ids and a spelling attachment resolving to a tombstoned pitch. Every core graph invariant holds.

**Coverage.** Multi-region, multi-staff validity: concurrent regions must each be cast off and placed without cross-region interference. At the Minimal tier this entry demands validity and honest diagnostics on the projection it induces — the non-metric regions reach the solver as ordinary constrained IR (their decompositions deferred upstream), so the entry does *not* smuggle the Advanced tier’s proportional/aleatoric layout obligation into Minimal; it guards that a multi-region document neither breaks hard constraints nor corrupts the report.

**Tiers, thresholds, expectations.** Required at Minimal; in the Standard subset. Catalog default thresholds; no overrides; no fixed expectations.

## 3.4 RS-3: B-flat Major Scale

**Construction.** Corpus entry `b_flat_major_scale` (zero-argument deterministic builder).

**Content.** Seven quarter notes ascending through B-flat major (B $\flat$  C D E $\flat$  F G A) in one metric region, one staff, one voice — accidentals inferred in a flat context.

**Coverage.** The accidental entry: flats produce glyph clusters with left overhang, exercising

overhang-aware spacing, the same-column cluster of accidental and notehead (the collision axis's same-slot exclusion), and the `spacing` axis under uneven column ink. Single system and page: the break-family and page axes degenerate.

**Tiers, thresholds, expectations.** Required at Minimal; in the Standard subset. Catalog default thresholds; no overrides; no fixed expectations.

### 3.5 RS-4: Two-Voice Counterpoint

**Construction.** Corpus builder `two_voice_counterpoint` (zero-argument, deterministic).

**Content.** Two voices in one staff instance: an upper quarter-note line and a lower quarter-note line two octaves beneath it, sounding simultaneously.

**Coverage.** Simultaneity: both voices' notes share musical time columns, so each spring slot carries a two-voice cluster, and the low line sits below the staff (ledger territory). Exercises same-column stacking, the collision sweep across a vertically spread texture, and column-advance regularity when columns are ink-heavy. Single system and page: the break-family and page axes degenerate.

**Tiers, thresholds, expectations.** Required at Minimal; in the Standard subset. Catalog default thresholds; no overrides; no fixed expectations.

### 3.6 RS-5: Notes and Rests

**Construction.** Corpus entry `notes_and_rests` (zero-argument deterministic builder).

**Content.** Note, rest, note, rest — quarter values in one metric region, one staff, one voice.

**Coverage.** Rest glyphs interleaved with noteheads: rest columns carry different ink boxes than note columns, exercising the spacing pass's treatment of mixed column content and the collision sweep over non-notehead glyphs. Single system and page: the break-family and page axes degenerate.

**Tiers, thresholds, expectations.** Required at Minimal; in the Standard subset. Catalog default thresholds; no overrides; no fixed expectations.

### 3.7 RS-6: Three-Four Meter Line

**Construction.** Corpus entry `meter_three_four` (zero-argument deterministic builder).

**Content.** Three quarter notes under a declared  $\frac{3}{4}$  time signature (three quarter-beat groups), in one metric region, one staff, one voice.

**Coverage.** Meter variety: the declared time signature drives measure length past the whole-note default, exercising the meter-resolution path that feeds the layout projection. Guards that a non-default meter neither breaks hard constraints nor perturbs spacing regularity. Single system and page: the break-family and page axes degenerate.

**Tiers, thresholds, expectations.** Required at Minimal; in the Standard subset. Catalog default thresholds; no overrides; no fixed expectations.

# Harness Binding (Non-Normative)

---

This chapter is informative. The reference implementation binds this suite to executable checks in its test kit, following the test kit's established library-module-per-harness pattern (one public module per harness, mirroring the corpus and prepass harnesses, with an integration test driving it). The suite harness constructs each Chapter 3 entry from its named builder and seed, solves it under the declared configuration, and asserts the Minimal-tier pass of Requirement 2.3; it is delivered with the reference implementation.

## NON-GOAL

The harness is not part of this document's normative surface. Conformance is defined by Chapters 2 and 3 alone; an independent implementation may bind the suite with any machinery that evaluates the same entries under the same configurations. Likewise, the reference implementation's golden files (SVG snapshots, byte-anchored layouts) are that implementation's regression locks, not suite conformance (Section 2.4).

# Revision History

---

| Date         | Section | Change                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| July 3, 2026 | All     | 0.1.0 — Initial companion: pins the suite entry model (builder-and-seed score referencing with the .musc-bundle open question, per-entry solve configuration over the declared A4/8mm default geometry, the four-condition per-entry pass rule, the deliberately empty fixed-expectation set, suite-version declaration); delivers the six-entry vo.1 set (ten_measure_single_staff seed 0x000A_11CE, valid_score_rich seed 0xF302, b_flat_major_scale, two_voice_counterpoint, notes_and_rests, meter_three_four), all required at Minimal and all constituting the Standard subset under the Quality Metric Catalog’s default thresholds, with no per-entry overrides. |