



HIGG ASSESSMENT MODEL

Foundational Environmental Performance (FEP)

Version ham-v0.3.0 · 11 September 2026

This document explains how Foundational Environmental Performance (FEP) is represented in the Higg Assessment Model — what it measures, how the achievement aggregates are calculated, and what to read into the four FEP fields exposed on the Assessment record.

Contents

- Purpose of the Calculation 3**
- Record Types 3**
- The Four Aggregate Fields 4**
- Methodology 4**
 - Per-question Logic 4
 - Aggregation 4
- Caveats 5**
- Cadence Coverage 5**
 - FEP Standalone Variant (fem2026 and later) 5
- Cadence Specific Guidance 6**
 - FEM 2023 6
 - FEM 2024 and FEM 2025 6
 - FEM 2026 and FEM 2027 6

This document explains how Foundational Environmental Performance (FEP) is represented in the Higg Assessment Model — what it measures, how the achievement aggregates are calculated, and what to read into the four FEP fields exposed on the `Assessment` record.

FEP is a tag overlay on a subset of the Higg FEM Level 1 questions. The questions themselves are existing FEM content; FEP marks the ones that represent the industry-aligned baseline of environmental management practices. Calscale published the definition through the AIDFEP (Aligned Industry Definition of FEP) consultation process in 2024–2025; the result is a curated set of ~60 to ~100 reference question IDs per cadence, each scored against a small allow-list of answer values. The full stakeholder rationale is in `stl-local-release/reference/fep.md`; this guide covers what's actually in the bundle.

Purpose of the Calculation

FEP rolls up the assessment's responses against the curated foundational question set into four summary fields, giving consumers a single view of how a facility scores on the baseline expectations without re-deriving the achievement criteria question-by-question.

The achievement criteria are deliberately narrow:

1. **Most foundational questions are pass/fail on "yes".** The facility either confirms the practice is in place or it doesn't.
2. **Some questions accept partial credit on "partialYes".** Where a foundational practice can meaningfully exist in a partial form (e.g. wastewater tracking implemented for some streams but not all), "partialYes" is treated as a passing answer alongside "yes".
3. **A few questions accept "not applicable".** Where a foundational practice can plausibly not apply to a given facility (e.g. chemical management at a facility with no chemicals on site), "na" is treated as a passing answer — the facility hasn't failed; the question just doesn't apply to it.

The aggregate fields treat *applicability* and *achievement* as distinct: a question that resolves to `undefined` (the facility's answer or conditional logic indicates the question doesn't apply at all) is excluded from both the numerator and denominator of the achievement ratio. This means a facility that's never asked a particular foundational question doesn't get penalized for not answering it.

Record Types

The FEP aggregates live on a single record type:

- **Assessment:** Whole-facility FEP roll-up. Four fields per assessment, described below.

The underlying per-question Fep records are also present in the bundle — they appear as compiled blocks under each cadence's RFI PID in `model.json.gz`. Each carries:

- `__name__`: `fep_<refid>` (e.g. `fep_chemstorage`).
- `__type__`: `Fep`.
- `ref_id`: the FEM question's ref ID (e.g. `chemstorage`).
- `logic`: a JS expression that evaluates to `true`, `false`, or `undefined` once the facility's answers are supplied.

Consumers that want per-question detail rather than the aggregates read the `Fep` blocks directly. The aggregates on `Assessment` are the convenience layer.

The Four Aggregate Fields

All four are `number?` or `bool?` and follow a single gating contract: if no Fep in the assessment evaluates to `logic === true` (either because no Feps are defined for the cadence, none are applicable, or all applicable ones failed), every field is `undefined`. The achievement story is treated as not meaningful in that state rather than projecting a misleading “0 out of N” mix into the data.

- **fep_achieved** (`bool?`): `true` when every applicable Fep has `logic !== false` (so achieved or N/A); `false` if at least one applicable Fep failed; `undefined` per the gate above.

Read this as “did the facility meet the foundational baseline?” when you want a single yes/no answer.

- **fep_achievement** (`number?`): The ratio of achieved Feps to applicable Feps. Always in `[0, 1]`. `1.0` means every applicable Fep passed; a fractional value means some applicable Feps failed.

Read this when ranking or benchmarking — the ratio is the most useful field for comparing facilities to each other or to themselves over time.

- **fep_achieved_count** (`number?`): Count of Feps whose logic evaluated to `true` (achieved). Use alongside `fep_unachieved_count` to display “X of Y” on a UI.
- **fep_unachieved_count** (`number?`): Count of Feps whose logic evaluated to `false` (applicable, not achieved). Use alongside `fep_achieved_count`. The two counts plus the count of `undefined`-logic Feps sum to the total Fep set defined for the cadence.

Methodology

Per-question Logic

Each foundational question’s `logic` is a JS expression that maps the facility’s answer to the question into one of three states:

- `true` — achieved. The answer was on the allow-list for this question.
- `false` — not achieved. The answer was outside the allow-list but the question was answered.
- `undefined` — not applicable to this facility (the question was never resolved, or the facility’s conditional path skipped it).

The allow-list pattern depends on the question’s option list:

- **Yes/No questions:** allow-list is `["yes"]`.
- **Yes/PartialYes/No questions:** allow-list is `["yes", "partialYes"]` — both pass.
- **Yes/No/N-A questions:** allow-list is `["yes", "na"]` — “doesn’t apply” is a pass, since you can’t fail an expectation that doesn’t apply to you.

The `undefined`-guard wrapper (`<ref> !== undefined ? <check> : undefined`) is what keeps a hidden conditional question from counting against the facility. This is the canonical Fep pattern; every Fep block follows the same shape.

Aggregation

The four aggregate fields are computed by walking the cadence’s Fep set in compile context and producing JS expressions that:

- `fep_achieved`: `ANDS logic !== false` across every Fep.
- `fep_achievement`: `divides count(logic === true) by count(logic !== undefined)`.
- `fep_achieved_count`: `sums logic === true ? 1 : 0` across Feps.

- `fep_unachieved_count`: `SUMS logic === false ? 1 : 0`.

Each is gated on `OR logic === true` across every Fep — if no Fep passed, all four aggregates evaluate to `undefined`.

Caveats

- **No additional scoring penalty.** A failed Fep does not subtract from the FEM score on its own. The underlying FEM questions are Level 1 content with the usual scoring weight, so a “no” answer on a foundational question already reduces the FEM score through that path. The FEP aggregates are a reporting overlay, not a separate scoring mechanism.
- **Cadence-specific question set.** The set of questions tagged as Foundational evolves cadence to cadence. A direct year-over-year comparison of `fep_achievement` is meaningful only if the question set is unchanged; otherwise the ratio’s denominator has shifted underneath you. The cadence-specific guidance below notes when the set changes.
- **Single-question dependencies.** Each Fep’s logic operates on one question. There are no compound Feps in the current cadence set. If a foundational expectation involves multiple FEM questions, each gets its own Fep record and the relationship is captured implicitly by both Feps appearing in the aggregate.
- **No country-specific Feps except `sipiperecords`.** The foundational set is global, with one documented exception: the China-only IPE-database question. Don’t expect to find FEPs keyed to region in the bundle today.

Cadence Coverage

FEP is available for FEM 2023 onwards. Earlier cadences (`fem2017–fem2022`) have no Fep records and the aggregate fields on `Assessment` evaluate to `undefined` for those cadences.

Cadence	FEP available	Notes
<code>fem2017 – fem2022</code>	No	No <code>extras/fep.stl</code> ; aggregate fields <code>undefined</code> .
<code>fem2023</code>	Yes	Initial Fep set (~67 records).
<code>fem2024 – fem2025</code>	Yes	Expanded set (~98–99 records).
<code>fem2026 – fem2027</code>	Yes	Same shape as 2024–2025 plus the <code>fep_standalone</code> variant — see below.

FEP Standalone Variant (`fem2026` and later)

Starting in `fem2026`, FEM cadences expose a `fep_standalone` variant alongside the regular `fem` variant. The standalone variant serializes only the FEP-tagged questions, so a consumer that wants to render only the foundational baseline can do so without pulling the full FEM module.

For HAM consumers this is transparent — the same `Fep` blocks appear in the compiled output regardless of which variant generated the assessment. The standalone variant is a platform-side question-set filter, not a separate logic path.

If a facility completes only the FEP standalone assessment, the aggregate fields on its `Assessment` reflect that subset directly. There is no “is this a standalone assessment?” flag on the record; infer it from the cadence’s variant tagging if you need to.

Cadence Specific Guidance

FEM 2023

The initial FEP set was scoped to ~67 questions. Worker-engagement, chemical management, and wastewater controls dominate the set. `fep_standalone` is not available for this cadence.

FEM 2024 and FEM 2025

The set expanded to ~98–99 questions, broadening into energy and emissions topics. A small number of fem2023 Feps were retired where the underlying FEM question changed shape. `fep_standalone` is not available for these cadences.

FEM 2026 and FEM 2027

Same Fep set shape as fem2024–fem2025, with continued additions and adjustments where the underlying FEM content shifts. The `fep_standalone` variant is introduced — see above. As before, the aggregate fields on `Assessment` reflect whichever subset is present in the evaluated assessment.