# JVDT-7 GEM Resource Documents Specification

This document outlines the necessary data and logic resources required for a GEM to accurately interpret JVDT-7 diagnostic results and generate reports according to the jvdt7\_gem\_instructions\_v2.md specifications.

**Goal:** To provide the GEM with all static knowledge and procedural rules needed to function independently and consistently.

**1. JVDT-7 Core Framework Definitions (framework\_definitions.json)**

* **Axes & Poles:**
  + A list of the 7 axes, each specifying:
    - axisID (e.g., "Perception", "Interpretation", ...)
    - pole1\_key (Namespaced, e.g., "PERCEP\_Assoc")
    - pole1\_letter (Standard letter, e.g., "A")
    - pole2\_key (Namespaced, e.g., "PERCEP\_Anal")
    - pole2\_letter (Standard letter, e.g., "N")
    - one\_liner\_description (The exact text specified in instructions, e.g., "How the learner takes in information...")
* **Virtue Ladder Stages:**
  + A list defining each stage (1-5):
    - stage\_number (1 to 5)
    - stage\_name (e.g., "Instinct", "Awareness", "Balance", "Mastery", "Wisdom/Integration")
    - core\_meaning (Brief description, e.g., "Highly skewed; focus on recognizing counter-pole value.")
* **Train Journey Stations:**
  + List of the 4 stations: "Information", "Integration", "Comprehension", "Application" with brief definitions.
* **Core Values:**
  + List of the 3 values: "Love", "Respect", "Happiness" with brief definitions.

**2. Scoring & Interpretation Rules (interpretation\_rules.md)**

* **Input Prioritization:** Explicit rule: Percentages > Margin > Dominance Label > Code Only.
* **Pole Dominance Calculation:** Algorithm for determining the dominant pole from percentages or margin. Specify tie-breaking if necessary (though current logic leans towards Pole 1/default).
* **Code Generation:** Procedure for constructing the 7-letter standard code from dominant poles.
* **Margin Calculation:** Formula: Margin = |Pole1% - Pole2%|.
* **Virtue Ladder Stage Mapping:**
  + Explicit margin bands: ≥ 80% -> Stage 2, 65-79% -> Stage 3, 55-64% -> Stage 4, ≤ 54% -> Check II.
  + Integration Index Check for Stage 5: Rule IF margin ≤ 54% AND integration\_index ≥ 66 THEN Stage 5 ELSE Stage 4.
  + Handling Label-Only Input: Rule Default to Stage 4, adjust based on Notes.
  + Handling Explicit Stage Input: Rule Use provided Stage, override margin calculation.
* **Integration Index Estimation:** Formula/lookup table: S2 ≈ 25%, S3 ≈ 40%, S4 ≈ 60%, S5 ≈ 80%. State clearly this is an estimate and *cannot* be used to confirm Stage 5.
* **Missing Data Handling:** Rule If data for an axis is missing, output "Data not provided" for that axis and proceed. Calculate global metrics (like average II) only on available data.
* **Consistency Checks:** Rule If code provided conflicts with calculated dominance from percentages/margin, note the discrepancy and prioritize percentage/margin data.

**3. Content Libraries (Separate JSON files recommended)**

* **Epithets Library (epithets.json):**
  + Key-value store mapping 3/4-letter namespaced patterns (e.g., "PERCEP\_Assoc-INTERP\_Root-REFLEC\_Internal") to { title, tagline, description }. Include the default/fallback entry.
* **Macro-Families Library (macro\_families.json):**
  + Key-value store mapping 2-letter standard patterns (e.g., "A-R", "N-C") to { title, tagline, description, core\_virtue, inner\_question }. Include the 4 complex families and a default/fallback.
* **Axis Microtexts Library (microtexts.json):**
  + Key-value store mapping all 14 namespaced pole keys (e.g., "PERCEP\_Assoc", "VALUE\_Respect") to { title, text }. Include a default/fallback.
* **Blind Spots Library (blind\_spots.json):**
  + Key-value store mapping namespaced blind spot patterns (e.g., "PERCEP\_Assoc\_High", "PERCEP\_Assoc+PERCEP\_Anal\_High", "PERCEP\_Assoc+APPLIC\_Ideal\_High") to { pattern, tendency, prompt }. Include a default/fallback.
* **Virtue Ladder Practices Library (virtue\_ladder\_practices.json):**
  + Nested structure: AxisID -> StageNumber -> { stage, title, description, practice, reflection }. Contains all 35 entries. Include default/fallback rung data.

**4. Report Template Structures (report\_templates.md)**

* **Parent Report:** Detailed outline specifying sections (Header, What JVDT-7 Measures, Axis Snapshots, Growth Focus, Action Plan, Language Tests, Closing) and the exact data points/content library keys to populate each part. Specify formatting (bullets, tables).
* **Teacher Plan:** Detailed outline specifying sections (Student Info, Development Targets, Classroom Strategies, Rubrics) and data/content mapping. Specify format.
* **Student Summary:** Outline specifying content (Epithet, Strengths, Habits, Encouragement) and constraints (≤200 words, second person).
* **One-Liner:** Format definition.
* **Progress Tracker:** Table column definitions (Target Axis, Current Stage, Target Stage, Practice Focus, Metric, Start Date, Review Date).

**5. Language Test Interpretation Guide (language\_test\_guide.md - Optional but Recommended)**

* Provides qualitative interpretations for common score ranges on relevant tests (e.g., CEFR A2, IELTS 5.5).
* Suggests potential correlations between specific test results (e.g., low reading comp) and JVDT Train Stations or Axis imbalances (e.g., potential challenge in Integration station, possibly linked to A/N balance). *This requires careful JVDT pedagogical expertise.*

**6. Few-Shot Examples (examples.json)**

* An array of complete examples, each containing:
  + input: Mock learner profile, JVDT percentages/labels, optional language scores, optional notes.
  + expected\_output: Fully formatted Parent Report, Teacher Plan, and Student Summary exactly as the GEM should generate them based on the input and rules. Include examples covering different stages, balanced axes, and missing data.

**7. Glossary of JVDT Terms (glossary.json)**

* Simple key-value store defining all specific JVDT terms used throughout the instructions and content libraries (e.g., "Virtue Ladder", "Integration Index", "Train Journey", axis names, pole names) to ensure the GEM uses them correctly.

**Rationale:**

Providing these resources in a structured, machine-readable (where possible, like JSON) or clearly defined format ensures the GEM has an unambiguous knowledge base. This minimizes errors, ensures consistency with the JVDT methodology, and makes future updates to the system (like adding new practices or refining archetype descriptions) much easier by simply updating the relevant resource file.