# prd02 jpe language and jpe xml

PRD 02 — JPE Language and JPE-XML

1. Product Vision
The JPE language and its XML fork (JPE-XML) provide a human-readable, structured way to describe Sims 4 mods. JPE is a plain-English DSL; JPE-XML is an XML dialect using English tag names that maps cleanly to game tuning. Both compile to the same Intermediate Representation.

2. Goals
- Make mod logic easy to read and write in plain language.
- Preserve enough structure for deterministic compilation to game tuning.
- Keep the language small and opinionated while allowing extension via templates and libraries.
- Facilitate high-quality error messages by tying grammar and schema to diagnostics.

3. JPE DSL Overview
Example (illustrative syntax):

```
mod "Petty Texts" version 1.0

interaction "Send Petty Text" for sims:
  when sim uses phone on another_sim
  test:
    relationship_with another_sim <= -10
  effect:
    add buff "PettyText_Angry"
    change relationship_with another_sim by -5
    show notification "That text was cold as hell."
```

4. JPE Language Requirements
- R1: Top-level constructs include: mod, interaction, buff, trait, loot, test_set, enum, and string_table.
- R2: Support named blocks like when, test, effect, and loot within interactions and other entities.
- R3: Provide consistent syntax for state changes, such as:
  change STAT_NAME by VALUE
  add buff BUFF_NAME
  grant loot LOOT_NAME
- R4: Allow references to traits, buffs, statistics, and other resources by name, with optional explicit IDs.
- R5: Support optional metadata such as versions, authors, and tags.
- R6: Provide a clear, machine-readable grammar to enable robust tooling.
- R7: Associate each grammar rule with friendly error messages for malformed syntax.

5. JPE-XML Overview
Example:

```
<mod name="Petty Texts" id="khaotic_petty_texts" version="1.0">
  <interaction name="Send Petty Text" id="khaotic_SendPettyText">
    <when>sim uses phone on other_sim</when>
    <tests>
      <relationship target="other_sim" operator="less_or_equal" value="-10"/>
    </tests>
    <effects>
      <add_buff ref="PettyText_Angry"/>
      <relationship_change target="other_sim" delta="-5"/>
```

```
    <notification text="That text was cold as hell."/>
   </effects>
  </interaction>
</mod>
```

6. JPE-XML Requirements
- R1: Provide an XML schema (XSD or similar) defining valid tags and attributes.
- R2: Ensure each JPE-XML construct has a deterministic mapping to IR.
- R3: Allow explicit IDs and references to support advanced users.
- R4: Allow future additions (new tags or attributes) via versioned schema.
- R5: Surface XML schema violations as structured EngineError instances with clear messages.

7. Mapping to IR and Tuning
- JPE text → parsed according to the grammar → IR entities.
- JPE-XML → validated against schema → IR entities.
- IR → XML tuning via the Core Translator Engine.
- Keep the mapping rules documented and versioned so that updates do not silently break existing mods.

8. Error Handling
- When syntax or schema errors occur, report the line/column, offending token or element, and a clear description.
- Provide specific error codes for:
  Unexpected keywords.
  Missing required blocks (e.g., effect or test).
  Invalid or unknown tags or attributes in JPE-XML.
- Integrate with diagnostics so errors appear in the Problems pane and mobile indicators.

9. Versioning
- Maintain jpe_version and jpe_xml_version fields in projects and documents.
- Document breaking changes and provide migration guides.