MadeManifest Calculation Engine User Manual

This manual documents how to run the deterministic calculation engine in this bundle, including command-line options, usage, computation details, and output format.

Overview

The engine calculates objective, deterministic outputs for: - Astrology - Human Design - Gene Keys (derived from Human Design)

It uses the pinned Swiss Ephemeris version and the pinned canon files. The output is designed for strict, bit-exact comparison against the golden test case.

Quick Start

From the repo root:

```
make prepare
make compile
make run
make diff
```

The `make run` target produces `mademanifest-engine/out.json` from the golden input. `make diff` performs a strict diff against the golden expected output.

Command-Line Usage

The engine binary is built as `mademanifest-engine/proof-of-capability-2`.

Basic usage:

```
mademanifest-engine/proof-of-capability-2 [options] <input.json> <output.json>
```

The program requires exactly two positional arguments: - `input.json`: JSON input case file - `output.json`: output file to write

Options

- `--canon-directory`, `-cd` (default: `canon`)
  - Base directory for canon files. If relative, it is resolved against the current working directory.
- `--gate-sequence-file`, `-gs` (default: `gate_sequence_v1.json`)
  - Canon gate sequence file. If relative, it is resolved against `--canon-directory`.
- `--mandala-constants-file`, `-mc` (default: `mandala_constants.json`)
  - Canon mandala constants file. If relative, it is resolved against `--canon-directory`.

- `--node-policy-file`, `-np` (default: `node_policy.json`)
  - Canon node policy file. If relative, it is resolved against `--canon-directory`.
- `--dos`
  - Write output with CRLF line endings.
- `--help`, `-h`
  - Print usage and exit.
- `--version`, `-v`
  - Print engine version string and exit.

Example

From `mademanifest-engine/`:

```
./proof-of-capability-2 -cd ../canon ../golden/GOLDEN_TEST_CASE_V1.json out.json
```

Environment Variables

- `SE_EPHE_PATH`
  - Path to Swiss Ephemeris data files. If unset, defaults to:
    - `../ephemeris/data/REQUIRED_EPHEMERIS_FILES/`
- `SE_NODE_POLICY`
  - Controls the node used by `GetPlanetLongAtTime` for `north_node` lookups.
  - If set to `true`, a true node is used; otherwise mean node is used.
  - Note: Human Design uses true node explicitly via a separate call, independent of this variable.

Input Contract

The input file is a JSON document with these required sections: - `case_id` - `birth` - `engine_contract`

The engine merges canon defaults into the input before processing. Input values override canon defaults.

`birth` fields

- `date`: `YYYY-MM-DD`
- `time_hh_mm`: `HH:MM` (seconds are assumed `00`)
- `seconds_policy`: must be `assume_00`
- `place_name`: text name (not used for computation)
- `latitude`: decimal degrees
- `longitude`: decimal degrees
- `timezone_iana`: IANA timezone name

`engine_contract` fields

The engine asserts the following contract values: - `ephemeris`: `swiss_ephemeris` - `zodiac`: `tropical` - `houses`: `placidus` - `node_policy_by_system.astrology`: `mean` - `node_policy_by_system.human_design`: `true` - `node_policy_by_system.gene_keys`: `true` - `human_design_mapping.interval_rule`: `start_inclusive_end_exclusive`

The remaining fields are provided via canon defaults: - `human_design_mapping.mandala_start_deg` - `human_design_mapping.gate_width_deg` - `human_design_mapping.line_width_deg` - `design_time_solver.sun_offset_deg` - `design_time_solver.stop_if_abs_sun_diff_deg_below` - `design_time_solver.stop_if_time_bracket_below_seconds`

Computations Performed

This section summarizes the computation pipeline implemented in code.

1. Time Conversion

- Parse local `birth.date` and `birth.time_hh_mm`.
- Convert local time to UTC using the IANA timezone database (including DST rules).
- Convert UTC time to Julian Day (UT).

2. Ephemeris Longitudes

Using Swiss Ephemeris (version `2.10.03`) and tropical zodiac: - Compute ecliptic longitudes for: - Sun, Moon, Mercury, Venus, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, Chiron - Mean North Node - Derived values: - Earth longitude = Sun + 180° (mod 360) - South Node longitude = North Node + 180° (mod 360)

3. Astrology Module

- House system: Placidus (`swephgo.HousesEx` with `P`)
- Compute Ascendant and Midheaven (MC) from the `ascmc` output.
- For each object in positions:
  - Convert longitude into sign and degree/minute within the sign.
  - North node in astrology output is the mean node.

4. Human Design Module

Human Design uses two snapshots: - Personality: at birth time - Design: time before birth when the Sun longitude equals `birth_sun - sun_offset_deg`

Design time solving: - Target Sun longitude = `Sun(birth) - sun_offset_deg` (normalized 0–360). - Initial bracket: `birth - (sun_offset_deg ± 5)` days. - Expand bracket by 2-day steps up to 10 times until a sign change is found. - Solve with bisection until: - absolute Sun

difference < `stop_if_abs_sun_diff_deg_below`, or - time bracket < `stop_if_time_bracket_below_seconds`.

Mapping to gates and lines: - Use canon constants `mandala_start_deg`, `gate_width_deg`, `line_width_deg`. - Interval rule: start inclusive, end exclusive. - Gate index = floor(r / gate_width), line index = floor((r % gate_width) / line_width). - Gate sequence is the fixed 64-gate array from canon. - Output value format: `gate.line` with one decimal place (e.g., `51.5`).

5. Gene Keys Module

Gene Keys are derived directly from Human Design output: - Activation Sequence: - Life's Work = Personality Sun - Evolution = Personality Earth - Radiance = Design Sun - Purpose = Design Earth

Output Format

The output is a JSON document with deterministic ordering and formatting.

Top-level structure

- `case_id`
- `birth`
- `engine_contract`
- `expected`

`expected.astrology.positions`

Contains position objects with `sign`, `deg`, and `min`: - `sun`, `moon`, `mercury`, `venus`, `mars`, `jupiter`, `saturn`, `uranus`, `neptune`, `pluto`, `chiron`, `north_node_mean`, `ascendant`, `mc`

`expected.human_design`

- `activation_object_order`: fixed array order:
  - sun, earth, north_node, south_node, moon, mercury, venus, mars, jupiter, saturn, uranus, neptune, pluto
- `personality`: map keyed by the same objects, values formatted as `gate.line` with one decimal place
- `design`: map keyed by the same objects, values formatted as `gate.line` with one decimal place

`expected.gene_keys.activation_sequence`

- `lifes_work`, `evolution`, `radiance`, `purpose`
- Each has `{ "key": <int>, "line": <int> }`

Formatting rules

- Output JSON is rendered in a fixed order and spacing.
- With `--dos`, line endings are CRLF; otherwise LF.
- Floating-point values in the emitted JSON are formatted with fixed precision as defined in the renderer.

Determinism Requirements

- Use only the canon files provided.
- Use only Swiss Ephemeris version 2.10.03 and the bundled ephemeris data.
- Do not hardcode results; run the full computation pipeline.
- Output must be bit-exact identical to the golden output.