Compliance sheet — star-audit

Everything a supplier qualification, a legal review or an export check usually asks about star-audit 0.13.1, in one document.

1. Identity

Document	STAR-PKG-STAR-AUDIT-EN · v2.3 · 2026-11-14
Package	star-audit
Version	0.13.1
Capability	star.audit.geodesy
Maturity	Beta (VERIFIED)
Python	>=3.9
Source commit	dc09f61d
Source	https://github.com/Andrea07072000/S.T.A.R.-/tree/main/packages/star_audit

2. Licence and obligations

Apache License 2.0. You may use, copy, modify and distribute it, also commercially, and you receive a patent licence from the contributors. When you redistribute it: keep the LICENSE file and the copyright notices (and a NOTICE file, if a future version adds one), and state in modified files that you changed them. You may not use the S.T.A.R. name to endorse your product.

3. What it needs at run time

NEEDS	LICENCE	SOURCE
star-geodesy	Apache-2.0	S.T.A.R.
mpmath	BSD-3-Clause	https://mpmath.org/

4. Integrity

SHA-256 fingerprints of the files published on this website. Check a downloaded file with:

```
python -c "import
hashlib,sys;print(hashlib.sha256(open(sys.argv[1],'rb').read()).hexdigest())" star_audit-
0.13.1-py3-none-any.whl
```

FILE	SHA-256
star_audit-0.13.1-py3-none-any.whl	a6740807909318783d61d5c1c5889622621ca9a439bb5841b55531ab8f75d978
star_audit-0.13.1-source.zip	1985fe4c3b1cff42481ff2078cfa02f7ac5fa0a0524bea6eea1c61246f2d529c

5. What it does, and its limits

Measures the accuracy envelope (per altitude band, 0-36,000 km) of any ECEF->geodetic library against an exact reference.

That these libraries are "wrong": they may be tuned for near-surface use; other call options were not audited.

- `audit_function(fn)` / `run_external(python, code)`: 450 points (10 latitudes x 5 longitudes x 9 altitudes from 0 to 36,000 km) with KNOWN geodetic truth; the library converts their exact ECEF; errors are measured against truth.
- Envelope per altitude band: max latitude error (deg), height error (m), 3-D position error (m); worst case.
- The library runs in its own interpreter (`run_external`); a failing library run raises, never returns a score.
- `audit(python, driver)`: drives a time library ONLY through its documented UTC-string -> TAI-string API on 7 probes (incl. instants inside 23:59:60) and classifies OK / WRONG / REJECTED / PAST_TABLE_WARNED / PAST_TABLE_SILENT against a hand-written IERS Bulletin C truth.
- `audit(command, driver, cases)` vs Vallado SGP4-VER (33 cases, 667 epochs; fixtures with provenance): per-case max 3-D error, median, refusals. Cases are paired by ORDER (NORAD 20413 appears twice: short window and 3.5-year arc).
- `audit(impls, cases)`: every implementation runs in its own interpreter (driver `teme_to_gcrs(utc, r_km)`); probe validation first: Vallado's published TEME->GCRF example (2004-04-06

07:51:28.386009 UTC) must be reproduced to < 1 m; then pairwise divergence envelopes per year 1980-2050 on LEO and GEO vectors, and refusals.

- `audit(impls, cases)`: deterministic corpus (seed 20261004, sha256-pinned) of 40 valid TM frames (CCSDS 132.0-B, every header option: OCF, secondary header 1..8 octets) and 260 copies with 1..4 bit flips in header, data or FECF. Probe validation: an implementation counts only if it decodes every valid frame field-exact; a driver exception is never a credible rejection. Reports silent acceptance of bad-FECF frames, CRC collisions and acceptance disagreements.
- `audit(impls, epochs)`: each implementation in its own interpreter (`tdb_minus_tt(jd_tt)`); probe validation: every epoch within 50 us of the published low-precision formula (Explanatory Supplement, 0.001657 sin g + 0.000014 sin 2g); pairwise envelopes over 401 epochs 1900-2100, per century; refusals exclude.
- `audit(impls, epochs)`: each implementation in its own interpreter (`gmst_deg(jd_ut1)`); probe validation on Vallado's published Example 3-5 (GMST = 152.578787810 deg at 1992-08-20 12:14:00 UT1) within 0.1 arcsec; wrap-safe pairwise envelopes in arcsec over 201 epochs 1950-2050, per half-century; refusals exclude.
- `audit(impls, cases)`: rv -> (p, e, i, RAAN, argp, nu) per implementation in its own interpreter; probe validation on Vallado's Example 2-5 (tolerances 0.01 km / 1e-5 / 0.01 deg, sized on the printed digits); seeded, sha256-pinned corpus of 60 regular + 20 singular (near-circular, near-equatorial, both) states; regular cases compared with the generating truth and pairwise; singular cases: each implementation's CONVENTION (0, sentinel, NaN, value, refusal).
- `audit(impls, cases)`: what TLE readers accept. Reference = the format rule (checksum mod 10 with '-' = 1, line numbers, equal satellite numbers, 69 columns). Corpus: the 29 well-formed SGP4-VER TLEs plus 6 derived kinds (field_digit, checksum, line_number, satnum_mismatch, truncated = must reject; resummed = well-formed, must accept), sha256-pinned. Probe validation: every valid TLE accepted, its inclination read within 1e-4 deg and its catalogue number read exactly (a NaN inclination or a wrong number is a misread: before 0.13.1 both were validated).
- `audit(impls, cases)`: what Space Packet decoders accept (CCSDS 133.0-B rule: version '000', length field = data octets - 1). Corpus (sha256 of the rounded corpus pinned): 40 valid + 10 idle (accept) and 4 x 40 defective (version, truncated, trailing, header_only: reject). Probe validation: every valid packet decoded field-exact.
- `audit(impls, cases)`: elliptic Kepler solvers M -> E. Probe validation: Vallado Example 2-1 (M = 235.4 deg, e = 0.4 -> E = 220.512074767522 deg, tolerance 1e-9 rad). Truth: 8 eccentricities (0 to 0.9999) x 41 anomalies, the exact inverse (40 digits, mpmath) of the double M sent to each library; circular error. A non-finite answer on a regular case is a refusal. Hostile inputs (NaN, inf, e = 1, e > 1, e < 0, huge or negative M) are recorded as a behaviour category per library. Corpus fingerprint pinned.
- `audit(impls, cases)`: zero-revolution Lambert solvers, driver `lambert(r1, r2, tof, mu) -> v1` (prograde). Probe validation: Curtis Example 5.2 (v1 = (-5.9925, 1.9254, 3.2456) km/s, tolerance 1e-4 km/s). Truth without any Lambert solver: 315 cases built from known orbits (3 semi-major axes x 5 eccentricities up to 0.95 x 3 inclinations x 7 transfer angles from 10 to 330 deg); r1, r2 and the true v1 from the closed-form perifocal equations, time of flight from the forward chain nu -> E -> M. A

non-finite or wrong-length answer is a refusal. Hostile inputs (NaN, tof <= 0, zero vector, 0 and 180 deg transfers) are recorded per library. Corpus fingerprint pinned.

- `audit(impls, cases)`: Kepler propagators, driver `propagate(r0, v0, tof, mu) -> [x, y, z, vx, vy, vz]`. Probe validation: Vallado Example 2-4 (1e-3 km, 1e-5 km/s; the constant was cross-checked with NAIF prop2b). Truth without any propagator: 450 known orbits (3 semi-major axes x 5 eccentricities up to 0.95 x 3 inclinations x 5 transfer angles x 0 or 10 whole revolutions), states from the closed-form perifocal equations. A non-finite or wrong-length answer is a refusal. Each hostile input runs alone with a 120 s deadline; no answer is recorded as `error:Hang`.
- Every auditor reads the rows a driver printed through one guard: exactly one result per case, on the last output line, otherwise `RuntimeError`. Found by a hostile probe (2026-10-06): a reader that answered 1 case out of 203 was validated (the totals were counted on what came back), and an empty or non-JSON output raised undeclared exceptions.

6. Export, use and certification

The package is published and freely available. That usually takes software out of export-control lists, but not every use of it: read the page on space activities and export control before using it in a controlled programme. It is not certified or qualified for flight; the evidence describes what was tested and how.

7. Citation and bill of materials

```
@software{cavazzini_star_audit_0_13_1,
  author  = {Cavazzini, Andrea},
  title   = {{star-audit}: Geodetic conversion accuracy audit},
  version = {0.13.1},
  year    = {2026},
  url     = {https://andrea07072000.github.io/S.T.A.R.-/software/star_audit/},
  note    = {Apache-2.0. Source: https://github.com/Andrea07072000/S.T.A.R.-, commit dc09f61d}
}
```

CITATION.cff · BibTeX · SBOM CycloneDX 1.5 · SBOM SPDX 2.3

8. Warranty and liability

Provided "as is", without warranties or conditions of any kind (Apache License 2.0, sections 7 and 8), to the extent the law allows. This never excludes liability that the law does not allow to exclude.

Related: space activities and export control · acceptable use · evidence and limits · licence texts