Manual

How to use S.T.A.R.: the Workspace and its rooms, every tool, the Security Center, the states and the evidence, MURF, and the answers to common questions. Generated from the same texts the pages show.

Contents

1. What S.T.A.R. is
2. Getting started: the Workspace, My Desk and the profile
3. First tasks, step by step
4. The twelve rooms
5. The Workspace tools, one by one
6. Receipts
7. States, labels and evidence
8. The Security Center
9. MURF
10. The interactive museum
11. Packages and downloads
12. Privacy and data
13. Frequently asked questions
14. Glossary

1. What S.T.A.R. is

S.T.A.R. is an independent project that publishes engineering software together with the evidence of how it was checked. Its central product is pHisis-Ai, a local workstation for assembling AI models, scientific engines and verifiers: it is a concept being designed and cannot be downloaded yet; each of its components and its state is on the Components page.

What you can use today is in four areas: WORK, tools for everyday files and data that run in your browser; PROTECT, the Security Center; DISCOVER, research with its real status, including the interactive museum; SIMULATE, the MURF digital twins, still in research.

Every product and module carries one of seven states, defined once in the chapter on states. A state says how far an item has been checked; it is computed from the S.T.A.R. registry wherever a registry record exists.

2. Getting started: the Workspace, My Desk and the profile

Open the Workspace from the button at the top of every page. No account and no installation are needed. The first time a tool runs, the browser downloads its engine once; after that the tools answer locally.

At the top of the Workspace, My Desk lists twelve rooms. Without a profile every visitor sees them in the same order. The profile is a set of 36 multiple-choice questions in ten sections; its answers order the rooms and the suggestions and never hide anything. It is kept in your browser only if you save it, and can be exported, imported on another device and deleted.

If you allow it in the profile, My Desk also lists the five tools you opened last, on that device only.

Your profile →

3. First tasks, step by step

Five short tasks that use the site as it is today. The names in quotation marks are the labels of the buttons and fields.

Convert a value and keep a receipt

1. Open the Workspace and choose “Convert units”.
2. Select “Use an example” to fill in an example, or type a value and choose the two units.
3. Select “Run”. The first run downloads the engine once; later runs are faster.
4. Read the converted value and the exact factor used.
5. Select “Download the receipt (JSON)” and keep the file with the result: it records the software, its version, the input and the checks.

Find out what can be checked in a table

1. Open the Workspace and choose “What can I check in this file?”.
2. Choose a file or paste a table; it is read in your browser.
3. Select “Run”. The result says what each column seems to contain and which checks apply.
4. Open a suggested check from the result: the whole-table check opens with your table, a single check opens with a value taken from it.

Check that a file is the one you were sent

1. Open the Workspace and choose “Check that a file is unchanged”.
2. Choose the file in “File”.
3. If the sender gave you a fingerprint, paste it in “Expected value (optional)”.
4. Select “Run” and compare: the result shows the fingerprints of the file and whether the expected value matches.

Run the Security Center on its built-in example

1. Open the Security Center and select “Run on a built-in example”. The example is a synthetic project created in memory, with no real credentials.
2. Read the summary: the number of findings by severity, the files read and what was not reported, and why.
3. Read the table: findings are sorted from the most severe, each with what to do, the place and a short fingerprint, never the value.
4. Select “Download the report (JSON)” to keep the report. Then select “Choose a folder” to check a folder of your own; it is read in your browser and never uploaded.

Create a profile and see your rooms

1. Open the profile page from My Desk at the top of the Workspace.
2. Answer the questions you want; every question can be skipped. The order of your rooms updates as you answer.
3. Select “Save on this device” to keep the profile on this device, or “Export (JSON)” to keep it as a file.
4. Back in the Workspace, My Desk shows the rooms suggested for you; every room is still listed.

4. The twelve rooms

Each room has a purpose, a state and, where it exists, a page. Rooms that are specified but not built say so.

ROOM	WHAT IT IS FOR	STATE
My Desk	Your profile, the rooms in the order that suits you, and the tools you opened last, on this device only.	Beta
Daily Work	Calculations and checks for everyday files and data: distances, units, tables, fingerprints, tax codes, IBANs, working days.	Released
Developer Studio	Review a change to your code: the differences line by line and four checks on the added lines, in your browser. Code analysis and tests are specified, not built.	Beta
Security Center	A security check of a folder for exposed secrets, risky files and risky configuration, in your browser.	Beta
Data & Local Vault	Seal a data file into a record with its fingerprint, and see everything S.T.A.R. keeps in this browser and how to remove it. Today everything stays in the browser.	Beta
Science Lab	The Scientific Workbench: five engineering packages that run in your browser.	Beta
Space & Space Safety	Orbits in 3D, missions and verified astrodynamics.	Research
Chip & Hardware Lab	Processor architectures and quantum circuits, for learning; the hardware toolchain is not public.	Research
Energy & Systems	Time scales, synchronisation and the energy of computing.	Research

ROOM	WHAT IT IS FOR	STATE
MURF	Multi-fidelity digital twins with independent checks. Twin 0, collision probability, is being measured.	Research
Downloads	Packages and files with their fingerprints and how to verify them.	Released
Learn & Help	Documentation, the meaning of each state, and where to ask.	Beta

5. The Workspace tools, one by one

Each tool below has the same sheet you find under 'About this tool' on its panel: what it solves, what it takes, how to read the result, the checks recorded in the receipt and what it cannot guarantee.

What can I check in this file?

Drop or paste a table: the page recognises tax codes, VAT numbers, IBANs, dates, coordinates and measurements, and selects the applicable checks.

Choose a CSV or text file, or paste a table. It is read here, in your browser; nothing is uploaded.

About this tool

What it solves

See at once which checks fit a table you have, instead of trying the tools one by one.

What it takes

A CSV or text file, or a pasted table. Example: a small register with tax codes, VAT numbers, IBANs, dates and amounts, with some mistakes.

How to read the result

What each column looks like, how many values are valid, and buttons that open the right tools already filled in.

Checks in the receipt

Each value is tested with the same engines as the released checks (tax code, VAT number, IBAN) and the same reader of tables as the column tool.

What it cannot guarantee

It recognises kinds from the form of the values, so it can be wrong on unusual columns; it examines up to 5,000 values per column.

Run →

Check a whole table in one pass

Numbers, tax codes, VAT numbers, IBANs and dates of a table, checked together: a cleaned copy, every value read differently, one receipt.

Choose a CSV or text file, or paste a table, and tick the checks to run. Each check takes the columns of its kind; nothing is uploaded.

About this tool

What it solves

Check a register, an export or a spreadsheet in one go, instead of value by value, and keep a cleaned copy with a single receipt.

What it takes

A CSV or text file, or a pasted table. Example: a small register with tax codes, VAT numbers, IBANs, due dates and amounts, with some mistakes.

How to read the result

A summary for each check, the list of values read differently (before and after), the cleaned table to download and one receipt for all the checks.

Checks in the receipt

Each code goes through the same engines as the released checks, row by row; the tests compare every row with python-stdnum and read the cleaned table back to prove your columns are unchanged.

What it cannot guarantee

It marks values, it does not correct them; valid means formally valid; columns are chosen from the form of their values.

Run →

Distance between two places

On the real shape of the Earth, with the direction to travel.

Type two places as latitude and longitude (decimal, like 41.9028, or degrees-minutes-seconds, like 41°54'10"N), or pick a city.

About this tool

What it solves

The real distance and starting direction between two places on the Earth, not a straight line on a flat map.

What it takes

Two places as latitude and longitude (decimal or degrees-minutes-seconds), or picked from the list. Example: Rome to New York.

How to read the result

The big number is the distance along the Earth's surface (WGS-84 ellipsoid); directions are from North, clockwise; the sphere line shows how far off a simpler formula would be.

Checks in the receipt

The distance is walked back along the computed direction to see where it lands (round trip), and compared with a sphere; both are in the receipt.

What it cannot guarantee

Not a road, sea or flight route; no heights; not for navigation or safety decisions; only as accurate as the coordinates you type.

Run →

Convert coordinates

Decimal degrees, degrees-minutes-seconds and UTM, with a round-trip check.

Type a latitude and a longitude in any common format.

About this tool

What it solves

Rewrite a position in the formats maps and GPS devices use, without copying mistakes.

What it takes

Latitude and longitude in decimal degrees or degrees-minutes-seconds, with N/S/E/W. Example: Milan cathedral.

How to read the result

Decimal degrees, degrees-minutes-seconds, and UTM (zone, easting and northing in metres).

Checks in the receipt

Each conversion is converted back and compared with your input; special and polar UTM zones are flagged.

What it cannot guarantee

No change of datum (everything is WGS-84); UTM is not defined near the poles.

Run →

Check a column of numbers

Find missing, mistyped and suspicious values in a CSV; summary statistics.

Choose a CSV file, or paste the text. From Excel, LibreOffice or Google Sheets: select the column with its title, copy, and paste it here (or save the sheet as CSV; .xlsx files cannot be opened directly). Separators , ; | and tab are recognised; Italian numbers like 1.234,5 are read correctly and every conversion is listed.

About this tool

What it solves

Find what is wrong in a column of numbers before you trust a total or an average.

What it takes

A CSV file or pasted text; separators , ; | or tab; Italian numbers such as 1.234,5; text saved by Excel on Windows. Example: daily readings with mistakes.

How to read the result

How many values, the mean and median, the spread, and the values to double-check; every converted, missing or unreadable cell is listed; you can download a checked copy.

Checks in the receipt

Mean and median are computed by released S.T.A.R. packages; suspicious values are flagged by a stated rule (modified z-score above 3.5).

What it cannot guarantee

It cannot determine whether a suspicious value is actually wrong; it identifies the values to review. It does not open .xlsx files.

Run →

Fit a straight line

The trend in your readings, and how sure it is (slope and intercept with their uncertainty).

Choose a CSV file with two columns of numbers, or paste them. The first column is x and the second y, unless you choose otherwise after the first run.

About this tool

What it solves

See whether one quantity grows with another, and by how much.

What it takes

Two columns of numbers (x and y) as a CSV file or pasted text. Example: energy used against hours of work.

How to read the result

The line y = slope × x + intercept, the uncertainty of each, R^2 and the typical distance of the points from the line; a plot.

Checks in the receipt

The fit is computed by a released S.T.A.R. package; the uncertainties come from how the points scatter.

What it cannot guarantee

A straight line only. It does not prove cause and effect, and its uncertainty assumes independent, evenly scattered points.

Run →

Check that a file is unchanged

A fingerprint of the file (SHA-256 and others), compared with the value you expect.

Choose a file (up to 100 MB). Optionally paste the value it should have, as published by whoever sent it.

About this tool

What it solves

Show that a file is exactly the expected one — after a download, a copy, or before you send it.

What it takes

Any file, read only by your browser; optionally the value you expect (SHA-256, SHA-1, MD5 or CRC-32). Example: a short text file.

How to read the result

The fingerprints of the file; if you paste an expected value, a clear match or no match.

Checks in the receipt

SHA-256, SHA-1 and MD5 by Python's hashlib; CRC-32 by zlib and, for files up to 1 MB, also by the released star-crc, bit for bit.

What it cannot guarantee

A match proves the file is identical to the one the value was computed from — not that the file is safe, nor that the value came from someone you trust.

Run →

Convert units

Length, speed, energy, pressure, temperature and more, with the factor shown.

Type a value and choose the units.

About this tool

What it solves

Convert a value between units without slips of a power of ten.

What it takes

A number and two units of the same kind. Example: 26.2 miles in kilometres.

How to read the result

The converted value and the exact factor used.

Checks in the receipt

Factors from exact SI definitions in a released S.T.A.R. package; units of different kinds are refused.

What it cannot guarantee

Unit conversion only: no currencies, no material properties, no rounding rules of a particular trade.

Run →

Check an Italian tax code

Codice fiscale: check character, omocodia, sex, date and place of birth.

Type the 16 characters (spaces and lower case are fine). Optionally add surname and name to check their letters.

About this tool

What it solves

Catch a mistyped tax code before it goes on an invoice, a contract or a form, and read what the code says.

What it takes

The 16 characters; optionally surname and name. Example: RSSMRA80A01H501U, the textbook example.

How to read the result

Valid or not; the sex and possible dates of birth coded in it; whether it is an omocode, with its original form.

Checks in the receipt

The check character recomputed (D.M. 23/12/1976 tables); omocodia letters in the official order; a real birth date; surname and name letters if given.

What it cannot guarantee

It cannot say the code was issued or belongs to a living person; the place of birth is shown as a code, not a name.

Run →

Check an Italian VAT number

Partita IVA: check digit, office code; what it can and cannot tell.

Type the 11 digits; an IT prefix and spaces are removed and listed in the receipt.

About this tool

What it solves

Catch a mistyped VAT number before you issue or pay an invoice.

What it takes

11 digits, with or without IT. Example: a made-up number that only satisfies the rule (serial 0000001, office 001).

How to read the result

Valid or not; the serial and the office code; the check digit expected.

Checks in the receipt

The check digit with the Italian rule — which gives the same digit as the Luhn rule on 11 digits, and the receipt says so; the office code in its allowed ranges; a serial that is not zero.

What it cannot guarantee

It cannot say the number is active or whose it is: that is on VIES or at the Agenzia delle Entrate.

Run →

Check an IBAN

Any country: length and check digits; for Italy also CIN, ABI, CAB and account.

Type or paste the IBAN; spaces, lower case and a leading 'IBAN' are fine.

About this tool

What it solves

Catch a mistyped IBAN before a payment, and see how an Italian IBAN is made.

What it takes

An IBAN of any country that uses one. Example: IT60 X054 2811 1010 0000 0123 456, the example used in banking documents.

How to read the result

Valid or not; the IBAN in groups of four; for Italy the CIN, ABI, CAB and account.

Checks in the receipt

The length for the country (SWIFT registry); the two check digits by ISO 7064 MOD 97-10; for Italy the CIN letter.

What it cannot guarantee

A valid IBAN can still belong to someone else or to a closed account; the bank name is not looked up.

Run →

Working days and deadlines

Italian holidays, Easter, your patron saint: days between two dates, or a deadline.

Count the working days between two dates, or find the date N working days after a start. Dates as 2026-12-31 or 31/12/2026.

About this tool

What it solves

Count working days or set a deadline without forgetting Easter Monday or 4 October.

What it takes

Two dates, or a date and a number of working days; your patron saint's day if it is a holiday where you work. Example: December 2026.

How to read the result

How many working days, or the deadline and its weekday; which holidays fell in the period; possible bridge days.

Checks in the receipt

National holidays (L. 260/1949 as amended, 4 October from 2026); Easter by the Gregorian rule, checked to fall on a Sunday between 22 March and 25 April.

What it cannot guarantee

No regional, sector or company closures; not legal advice on how a legal deadline is counted (some rules move a deadline that falls on a holiday).

Run →

Compare two tables ● PROTOTYPE

Two versions of a list: rows removed, rows added, and every cell that changed, matched by a key column.

Paste the old table in A and the new one in B, with a first row of column names (CSV, semicolons or tabs). The key is the column that identifies a row, such as a code; leave it empty to use the first column the two tables share.

About this tool

What it solves

See exactly what changed between two versions of a price list, an inventory or a register, without checking row by row.

What it takes

Two tables with column names and a key column. Example: a parts list before and after an update.

How to read the result

Rows removed and added; for each changed row, the column and the old and new value.

Checks in the receipt

Rows matched by key, duplicate keys refused; counts compared with pandas (merge and compare) on the same tables.

What it cannot guarantee

It matches rows only by the key; 0.10 and 0.1 are the same number, but 10 % and 0.1 are different values.

Run →

Find duplicate rows ● PROTOTYPE

Rows that are the same once spaces and capitals are ignored, and rows that are nearly the same.

Paste a table with a first row of column names. Choose how similar two rows must be to be reported (1 = identical apart from spaces, capitals, accents and punctuation, 0.9 = about one character in ten may differ), and optionally which columns to compare, separated by commas.

About this tool

What it solves

Clean a contact list or a register before you send, count or import it.

What it takes

A table with column names; how similar rows must be; the columns to compare. Example: five contacts, two of them written twice.

How to read the result

Groups of identical rows by line number, and pairs of nearly identical rows with their similarity.

Checks in the receipt

Spaces, capitals, accents and punctuation normalised before comparing; edit distance checked against rapidfuzz.

What it cannot guarantee

Similar is not the same person: two rows that look alike can be two people, and the same person written very differently is missed.

Run →

Check the VAT of an invoice ● PROTOTYPE

Each line's VAT to the cent, totals by rate, and the gap that rounding line by line can leave.

Paste the invoice lines: taxable amount, VAT rate in percent and, if you have it, the VAT written on the line. Columns are recognised by name (imponibile, aliquota, IVA; taxable, rate, VAT) or by position. Optionally type the total VAT on the invoice.

About this tool

What it solves

Catch a wrong VAT amount or a rounding difference before you issue, pay or record an invoice.

What it takes

The invoice lines (taxable, rate, VAT if written) and the total VAT if you want it checked. Example: five lines at 22, 10 and 4 %.

How to read the result

VAT line by line, totals by rate, the invoice total, and the gap between rounding each line and rounding the total.

Checks in the receipt

Decimal arithmetic to the cent, halves up; totals checked against exact fractions on the same lines.

What it cannot guarantee

It does not know special regimes (reverse charge, split payment, exemptions) or which rounding your system uses; not tax advice.

Run →

What a file says about you ● PROTOTYPE

Author, device, place and dates hidden in a photo, a PDF or an Office file, read without uploading it.

Choose a photo (JPEG, PNG), a PDF or a Word, Excel or PowerPoint file. The file is read in your browser and never sent anywhere.

About this tool

What it solves

Before you send a photo or a document, see whether it carries your name, your device or the place where it was made.

What it takes

A JPEG, PNG, PDF, DOCX, XLSX or PPTX file. Example: a small photo with a camera name, an author and a position.

How to read the result

Every field found, what it reveals (person, device, place, dates, software, content), and a warning if there is a position.

Checks in the receipt

EXIF, XMP, PNG text, PDF information and Office properties read by the tool's own parser and compared with Pillow, exifread, pypdf, python-docx and openpyxl.

What it cannot guarantee

It shows the metadata but does not remove it; formats and fields it does not know are not read, so an empty report is not proof that the file is clean.

Run →

6. Receipts

A receipt records which software produced a result and from what. Keep it with the result or attach it to a report; anyone can recompute the fingerprint of the input and compare.

1. which S.T.A.R. software performed the work, with its version; 2. the input provided, with its SHA-256 fingerprint (the same input always yields the same fingerprint); 3. every change applied to the input before use; 4. the checks performed, each passed or failed; 5. the limits, and any item requiring human review. Keep it with the result, or attach it to a report.

7. States, labels and evidence

The seven states and what each requires:

STATE	WHAT IT MEANS	WHAT IT TAKES	WHAT IT DOES NOT MEAN
Released & Verified	Published, versioned and checked against references.	Every gate of the registry passed, including a mutation score of at least 0.85 and at least two independent implementations that agree within a stated tolerance.	A certification, a qualification for aircraft or spacecraft, or the absence of defects: defects found after release are listed in the failure observatory.
Released	Published and supported.	Its own tests and acceptance checks pass; the two-independent-implementations gate is not required.	Checked against an independent reference.
Beta	Usable, verified against references, still changing.	Working code with tests and at least one comparison with a reference; some release gates still open.	Stable interfaces or results that will not move.
Prototype	Working code that shows the idea.	It runs and has tests for what it claims.	Verified against a reference, or ready for use.
Research	An open question with a written method.	A plan, a preregistered criterion where there is a measurement, and the data it will use.	A result: no number is published before the measurement.
Planned	Specified, not built.	A written specification.	A date or a promise.

STATE	WHAT IT MEANS	WHAT IT TAKES	WHAT IT DOES NOT MEAN
Blocked	Stopped by a named obstacle.	The obstacle and what would remove it, stated on the item's page.	Abandoned.

The labels of the catalogue

Product and component pages use five short labels. Each one is one of the states above, said from the side of the person who wants to use it.

LABEL	WHAT YOU CAN DO WITH IT	STATE
● AVAILABLE	Open it and use it today, in the browser or as a published package. Its page says what it was checked against: its own tests and independent oracles, or the released packages it runs.	Released
● BETA	Open it and use it: it works and has been compared with references, but it is still changing.	Beta
● PROTOTYPE	Open it and use it: it is a Workspace preview, not released yet, and its results or its interface can still change.	Prototype
● RESEARCH	Read what is being studied and its limits. There is nothing to use yet, and no number is published before the measurement.	Research
● COMING SOON	Read what it will do and its limits. There is no button, because there is nothing to open yet.	Planned

A state says how far an item has been checked. Labels on a single result say where that number came from: SIMULATED, PREDICTED, CROSSCHECKED, FORMALLY VERIFIED, EXPERIMENTALLY VALIDATED. An item can be Beta and still publish a CROSSCHECKED number; it cannot be called Released & Verified until every gate is passed.

What the states mean → · Failure observatory →

8. The Security Center

Checks that find known risks before you share, publish or hand a project to an AI agent. Each report says what was checked, what was found, and what was not checked.

How your files are handled: the folder you choose is read by your browser, file by file, and never uploaded. The page opens no connection while it scans. The report shows where a secret is and a short fingerprint of it, never the secret itself.

Secrets (API keys, tokens, private keys, passwords in addresses, secret-looking assignments), files that should not be in a repository, and risky configuration of CI workflows, containers and installs.

Secrets

RULE	SEVERITY	WHAT TO DO
Private key	Critical	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
AWS access key ID	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
AWS secret access key	Critical	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
GitHub token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
GitHub fine-grained token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
GitLab personal access token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.

RULE	SEVERITY	WHAT TO DO
Slack token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Slack incoming webhook	Medium	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Stripe live secret key	Critical	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Stripe test secret key	Low	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Google API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
OpenAI API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
OpenAI project or service key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Anthropic API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
OpenRouter API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Groq API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the

RULE	SEVERITY	WHAT TO DO
		files and from the version history.
xAI API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Hugging Face token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Twilio API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Twilio account SID	Medium	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Twilio auth token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
npm access token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
PyPI upload token	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
SendGrid API key	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Telegram bot token	Medium	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.

RULE	SEVERITY	WHAT TO DO
Azure storage account key	Critical	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
JSON Web Token	Medium	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Password inside a URL	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Bearer token in an Authorization header	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Secret encoded in base64	High	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Secret-looking value assigned to a secret-looking name	Medium	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.
Unquoted secret-looking value assigned to a secret-looking name (.env, YAML, INI)	Medium	Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history.

Files that should not be shared

RULE	SEVERITY	WHAT TO DO
Environment file with possible secrets	High	Keep .env files out of the repository; commit a .env.example with placeholders instead.
SSH private key file	Critical	Remove the key from the repository and its history, and replace it: treat it as compromised.

RULE	SEVERITY	WHAT TO DO
Key store or certificate bundle	High	Key stores and password databases do not belong in a repository.
PEM or key file	Medium	If it holds a private key, remove it; a public certificate alone is not a secret.
Credential configuration file	High	These files usually hold tokens or passwords: keep them in the user's home directory or a secret manager.
Terraform state file	High	State files can contain plaintext secrets; keep them in a protected backend.
Database file in the repository	Low	Check that it holds no personal or production data.

Risky configuration

RULE	SEVERITY	WHAT TO DO
Workflow runs pull-request code with repository secrets	High	With pull_request_target the workflow has write access and secrets; do not check out and run the contributor's code in it.
Untrusted event text inside a shell step	High	Pass event text through an environment variable instead of interpolating it into the script.
Workflow token with write access to everything	Medium	Grant only the permissions each job needs.
Third-party action not pinned to a commit	Low	Pin actions to a full commit SHA so that a moved tag cannot change what runs.
Script downloaded and run in one step	Medium	Download, check the hash or signature, then run.

RULE	SEVERITY	WHAT TO DO
Container image runs as root	Low	Add a USER line so that the container does not run as root.
Package runs a script when installed	Low	Install scripts run on every machine that installs the package; check what they do.
Python requirement without a fixed version	Info	Pin exact versions (==) or use a lock file for reproducible installs.

What the check does not cover

- known vulnerabilities in dependencies (no vulnerability database is used offline)
- the Git history: secrets removed from the files but still in earlier commits
- binary files, archives inside the folder, and files larger than the size limit
- secrets in formats without a rule, however random: a value is reported only if a rule recognises its format or it is assigned to a secret-looking name
- whether a secret that was found is still valid (it is never sent anywhere to be tested)
- secret-looking assignments whose value has no digit or is only letters (the generic rule requires letters and digits, to avoid reporting words and identifiers)

Command line

The same rules as a Python package with no dependencies, for a pre-commit hook or a CI job. It exits with 1 when a finding reaches the chosen severity. Not yet a public package: it is in the engineering repository and cannot be installed today, so the command below is what it will look like, not something to run now.

```
python -m star_preflight FOLDER [--json] [--fail-on high]
```

Report a vulnerability

If you find a security problem in S.T.A.R. software or in this site, write to the contact address with the details; please do not publish it before we have answered.

Security Center →

9. MURF

Digital twins that combine reference solvers, fast surrogate models and independent checks, and that say how far each answer can be trusted. In development: this page says exactly what exists today.

A full simulation is accurate and slow; a fast model is quick and right only inside the region where it was checked. A multi-fidelity twin keeps both: it answers fast where the fast model is known to hold, calls the reference solver where it is not, and shows which of the two produced each number.

COMPONENT	WHAT IT IS	WHAT IT IS NOT	STATE
MURF Museum	the 3D museum of this site: forty rooms of declared models that show the ideas	a validated simulator	Prototype, for teaching
MURF Assistant	an agent for a small set of tasks, with budget, time limit, log and human approval	an agent with unlimited permissions	Planned
MURF Twin Engine	twins specialised by domain: reference solver, surrogate, uncertainty, escalation, independent checks	one model of everything; a space twin is not a chip twin	Research
MURF Connect	adapters to the libraries, solvers and data an organisation already uses	a replacement for its data centre	Planned

The first twin starts from an engine already measured, which is not yet a public package. S.T.A.R.'s two-dimensional collision probability agrees within 1e-7 (relative) with the adjusted value of NASA CARA on 53 of 53 real conjunctions and with Orekit on 48 of 48. Against a different CARA value, computed with another encounter definition, an earlier pre-registered rule of 1e-6 was not met on 11 of 48 cases; the Research page reports it. On 29 of those 53 cases the assumptions of the two-dimensional method do not hold. The question of Twin 0 is whether a second, more faithful level can be called only where the first one stops being valid, with a rule measured in advance.

Twin 0 sheet

System	two objects in Earth orbit at their closest approach
Inputs and units	positions and velocities (km, km/s), position covariances (km²), hard-body radius (m), from a conjunction data message
Reference	the Monte Carlo published by NASA CARA for each case (two-body motion), with its number of trials and its interval
Data	53 real conjunctions with the probability published by NASA CARA
Levels	level 0: the two-dimensional engine star.ssa.pc2d, cross-checked, not yet a public package; level 1: our own Monte Carlo on two-body orbits, written from the published description of the method
Validity of the fast level	encounter short compared with the orbits, motion straight during it, covariances constant: measured indicators, not assumed
Uncertainty	a 95 % interval for every Monte Carlo value; level 1 is accepted only if its interval overlaps NASA's on at least 90 % of the cases
When the fast level is not trusted	a rule chosen on one half of the cases (split fixed by a hash before any run), frozen with its fingerprint, then applied once to the other half
Independent checks	NASA CARA values, Orekit
Metrics	decisions at the thresholds 1e-7, 1e-5 and 1e-4 that differ from NASA's; cases sent to level 1; time per case; error on the logarithm of the probability
Result and limits	published with the cases where the method fails
When it does not know	it says so and returns the reference value or no value

Where it stands

Fast level (2D engine)	Cross-checked; not yet a public package
Benchmark and validity rule	Inconclusive: on 24 unseen cases the twin makes 5 decision errors, as many as the Monte Carlo alone and fewer than the 11 of the 2D engine alone, but it sends 62.5 % of the cases to the Monte Carlo (60 % allowed) and saves about 10 % of the time
Monte Carlo level	Second version (states drawn in orbital elements) agrees with NASA's Monte Carlo on 15 of 15 comparable cases never used to build it; the first version, drawn in Cartesian coordinates, was biased and is kept as a recorded failure
Web demonstration	After the benchmark, with real cases only

It does not replace CPUs, GPUs, CUDA, PyTorch or cloud computing: it uses them where they are strong. No speed-up or accuracy figure is published before a benchmark on declared hardware with its baseline. A fast answer is not proof of accuracy, and a model's own uncertainty is not a guaranteed error bound.

10. The interactive museum

State: Prototype, for teaching. Records and badges stay in the page unless you tick Remember my progress; unticking deletes them.

The museum is a set of exhibits you can handle: chips, orbits, quantum circuits, data. Each exhibit runs a declared model; the museum explains ideas and is not a validated simulator. It is the MURF Museum component described in the chapter on MURF.

The interactive museum →

11. Packages and downloads

The packages are Python libraries, plus one C library, published under the Apache License 2.0. Each one has a page with its installation command, its supported Python versions, its dependencies and its evidence. The table lists them with their state, read from the registry.

PACKAGE	VERSION	WHAT IT DOES	STATE
star-anomaly	0.1.0	Mean, eccentric, hyperbolic and true anomaly	Released & Verified
star-atmosphere	0.1.0	U.S. Standard Atmosphere 1976 (below 86 km)	Released & Verified
star-coverage	0.1.0	What a satellite sees, and from where it is seen	Released & Verified
star-cw	0.1.0	Relative motion near a circular orbit (Clohessy-Wiltshire)	Released & Verified
star-elements	0.2.4	Classical orbital elements to and from Cartesian state (Curtis Algorithms 4.2 / 4.5)	Released & Verified
star-gasdynamics	0.1.0	Isentropic flow and normal shock of a perfect gas	Released & Verified
star-iod	0.1.0	Velocity from three positions (Gibbs, Herrick-Gibbs)	Released & Verified
star-j2	0.1.0	What the Earth's oblateness does to an orbit, on average	Released & Verified
star-kepler	0.1.1	Two-body propagation with universal variables	Released & Verified
star-lambert	0.1.3	Universal-variable Lambert orbital boundary value problem solver	Beta

PACKAGE	VERSION	WHAT IT DOES	STATE
star-maneuver	0.2.2	Closed-form vis-viva and Hohmann orbital transfer engine	Beta
star-moon	0.1.0	Where the Moon is, to about 0.4 degree (0.2 degree for 95 % of dates)	Released & Verified
star-orbit	0.2.2	Numerical orbit propagation (Cowell, fixed-step RK4) with two-body + J2 gravity, linked to star_timescales	Beta
star-sun	0.1.0	Sun direction, beta angle and eclipse test	Released & Verified
star-tle	0.1.0	A strict reader of two-line element sets	Released & Verified
star-twobody	0.1.0	The sizes and speeds of a Keplerian ellipse	Released & Verified
star-calendar	0.1.0	Calendar dates and Julian day numbers, exactly	Released & Verified
star-epoch	0.1.0	Julian and Besselian epochs	Released & Verified
star-era	0.1.2	Earth Rotation Angle and IAU 2006 sidereal time	Released & Verified
star-sidereal	0.1.2	Greenwich mean sidereal time (IAU 1982)	Released & Verified
star-tdb	0.1.1	TDB - TT with the USNO Circular 179 series, verified against ERFA full series within 10 microseconds	Beta
star-timecode	0.1.0	CCSDS ASCII time codes A and B, strict	Released & Verified

PACKAGE	VERSION	WHAT IT DOES	STATE
star-coords	0.1.0	Spherical and cylindrical ⇄ rectangular coordinates	Released & Verified
star-ecliptic	0.1.0	Mean obliquity and equatorial ⇄ ecliptic directions	Released & Verified
star-galactic	0.1.0	Equatorial ⇄ galactic directions	Released & Verified
star-horizon	0.1.0	Hour angle and declination ⇄ azimuth and elevation	Released & Verified
star-mat3	0.1.0	The 3×3 matrix operations of frame and attitude code	Released & Verified
star-precession	0.1.0	Between J2000 and the mean equator of date (IAU 1976)	Released & Verified
star-rotframe	0.1.0	Position and velocity between an inertial and a rotating frame	Released & Verified
star-sphere	0.1.1	Angles between directions on the sphere	Released & Verified
star-vec3	0.1.0	The three-dimensional vector operations of orbit and attitude code	Released & Verified
star-albers	0.1.0	Albers equal-area conic projection	Released & Verified
star-ellipsoid	0.1.0	The WGS-84 ellipsoid as a surface	Released & Verified
star-geodesic	0.1.0	Distance and azimuth on the WGS-84 ellipsoid	Released & Verified

PACKAGE	VERSION	WHAT IT DOES	STATE
star-geodesy	0.1.1	WGS-84 geodetic to/from ECEF with exact inversion at any altitude	Beta
star-lcc	0.1.0	Lambert conformal conic projection	Released & Verified
star-mercator	0.1.0	Mercator on the ellipsoid, and Web Mercator	Released & Verified
star-polar	0.1.0	Polar stereographic projection and UPS	Released & Verified
star-topo	0.1.0	Look angles from a ground site	Released & Verified
star-utm	0.1.0	Transverse Mercator and UTM coordinates	Released & Verified
star-euler	0.1.0	Euler angles for the twelve sequences, one convention	Released & Verified
star-mrp	0.1.0	Modified Rodrigues parameters	Released & Verified
star-quaternion	0.1.0	Rotations with one stated convention	Released & Verified
star-slerp	0.1.0	Interpolation of attitudes along the shortest arc	Released & Verified
star-wahba	0.1.0	Attitude from vector observations	Released & Verified
star-aos-c	0.1.3	CCSDS AOS/TM transfer-frame check in C99 (S.T.A.R.)	Beta

PACKAGE	VERSION	WHAT IT DOES	STATE
star-crc	0.1.0	Cyclic redundancy checks, bit by bit	Released & Verified
star-doppler	0.1.0	Range, range rate and one-way Doppler shift	Released & Verified
star-spacepacket	0.1.1	CCSDS Space Packets, built and read strictly	Released & Verified
star-telemetry	0.2.7	CCSDS AOS/TM transfer frame deframer and Space Packet reassembler	Released & Verified
conjunction-screen-sgp4	0.2.1	Conjunction screening of TLE catalogues with SGP4, KD-tree coarse pass and closed-form TCA refinement	Beta
star-cdm	0.2.2	Strict CCSDS 508.0-B-1 Conjunction Data Message (KVN) parser with consistency report	Beta
star-chebyshev	0.1.0	Chebyshev series with their derivative	Released & Verified
star-interp	0.1.0	Lagrange interpolation through tabulated points	Released & Verified
star-linefit	0.1.0	Least-squares line and correlation, correctly rounded	Released & Verified
star-linsolve	0.1.0	Small linear systems, correctly rounded	Released & Verified
star-poly	0.1.0	Real roots of quadratics and cubics	Released & Verified
star-polyfit	0.1.0	Least-squares polynomial, correctly rounded	Released & Verified

PACKAGE	VERSION	WHAT IT DOES	STATE
star-quadrature	0.1.0	Gauss-Legendre quadrature	Released & Verified
star-quantile	0.1.0	Median, quantiles, IQR and MAD, correctly rounded	Released & Verified
star-rootfind	0.1.0	Bracketed roots to the last float	Released & Verified
star-sexagesimal	0.1.0	Degrees ⇄ degrees/minutes/seconds and hours/minutes/seconds	Released & Verified
star-sigma	0.1.0	What "n sigma" means in one, two and three dimensions	Released & Verified
star-spline	0.1.0	Natural cubic spline, correctly rounded	Released & Verified
star-stats	0.1.0	Summary statistics, correctly rounded	Released & Verified
star-tabint	0.1.0	Integrals of tabulated data, correctly rounded	Released & Verified
star-units	0.1.1	Unit conversion with exact definitions and a dimension check	Released & Verified
star-wrap	0.1.0	Reducing, differencing and averaging angles	Released & Verified
star-audit	0.13.1	Geodetic conversion accuracy audit	Beta
star-crosscheck	0.1.3	Cross-solver verification fabric (S.T.A.R.)	Released & Verified

Software catalogue → · Downloads →

12. Privacy and data

The Workspace tools and the Security Center read your files in your browser and do not upload them. The profile and the list of recent tools are kept in your browser only when you choose so, and can be deleted with one button or by clearing the site's data in the browser. This site has no accounts. The legal pages describe the processing in full.

Legal →

13. Frequently asked questions

Do I need an account or an installation?
No. The Workspace and the Security Center run in the browser. The packages are installed with pip only if you want them in your own software.

Are my files uploaded?
No. The Workspace tools and the Security Center read the files you choose inside your browser. The Security Center opens no connection while it scans.

Why is the first run slower?
The first time a Workspace tool runs, the browser downloads its engine once (about 12 MB). Later runs use the copy already loaded.

Does an AI model look at my data?
No. The Workspace tools and the Security Center are deterministic programs: the same input always gives the same output, and no language model is involved.

What does 'Released & Verified' mean?
Every gate of the S.T.A.R. registry passed, including a mutation score of at least 0.85 and at least two independent implementations that agree within a stated tolerance. It is not a certification.

Does S.T.A.R. software have a certification or a flight qualification?
No. It is tested and compared with independent references; it has no certification and no qualification for aircraft or spacecraft.

A folder with no findings: is it safe?
No. It is a folder in which these rules found nothing. Every report lists what the check does not cover.

Does the Security Center show or send the secrets it finds?
Neither. A finding shows the rule, the file, the line, the column, the length and the first 12 hexadecimal digits of the SHA-256 of the value, never the value itself.

What should I do when the check finds a secret?

Treat it as exposed: revoke or rotate it with its provider, then remove it from the files and from the version history. Removing it from the files alone is not enough.

Can I use the check in a CI job?
The command-line version exits with 1 when a finding reaches the chosen severity, which stops a pipeline. It will be published with its first release; until then it is part of the engineering repository.

Why does the check not report a value I know is a password?
To avoid reporting words and identifiers, a value assigned to a secret-looking name is reported only if it has letters and digits and enough randomness. Values made only of letters are in the list of what is not checked.

What is a receipt?
A record of a result: the software and its version, the input with its SHA-256 fingerprint, every change applied to the input, the checks performed and the limits. It lets anyone check that the result came from that input.

How do I verify a download?
Each file on the downloads page has its SHA-256 fingerprint; compute it on your copy and compare. A small script on the site does it for all files.

What is the profile for, and where is it kept?
It orders the rooms of the Workspace and the suggestions. It is kept in your browser only if you save it, never sent anywhere, and can be exported, imported and deleted.

How do I delete everything the site kept on my device?
Use 'Delete the profile' on the profile page, which also removes the list of recent tools, or clear the site's data in your browser settings.

Why would a room say 'No page yet'?
Every room has a page today. A room that is specified and not built would say 'No page yet' and carry the state Planned, instead of showing a mock-up.

What is MURF?
A research line on multi-fidelity digital twins with independent checks. Its first twin, on the collision probability of two satellites, is being measured against NASA's published values with a rule written before the measurement.

Is the museum a simulator?
No. Each exhibit runs a declared model to explain an idea. Results of the S.T.A.R. software are on the evidence pages, not in the museum.

What is the failure observatory?
The public list of defects found in S.T.A.R. software, including those found before release, with what was changed and the test that now guards against them.

Why do two pages sometimes show different numbers?
They should not. Counts of tools, products and packages are read from the registers when the site is built, and the build stops if a page quotes a different count; other figures come from evidence files. If you find a difference, please report it.

How do I report a vulnerability?
Write to the contact address with the details, and please do not publish it before an answer. The Security Center page describes the process.

Can I use the packages in commercial software?
The packages are published under the Apache License 2.0; its terms apply. Each package page lists its licence and its dependencies.

Which Python versions do the packages support?
Each package page states the supported versions and the dependencies, as declared in the package itself.

Which browsers are supported?
The automated checks of this site run in Chromium. Other recent browsers are expected to work but are not covered by those checks.

What does a label like SIMULATED or CROSSCHECKED on a number mean?
Where the number came from: a model, a surrogate, a comparison with an independent implementation, a formal proof or a real measurement. A number carries only the labels its evidence supports.

Who is behind S.T.A.R.?
An independent project by Andrea Cavazzini, Italy, not a company. The About page has the details.

Can S.T.A.R. work on my organisation's data?

Not in the current phase: no tailor-made work and no assessment is offered, and nothing is sold. The tools, the packages and the evidence are free for you to run on your own data, on your own computer.

Why are some tools marked as a preview?
A product is marked until its live test is recorded. The mark disappears when the test passes.

What happens to the Workspace when I close the tab?
What you entered or uploaded is erased with the tab. Only a saved profile and, if allowed, the list of recent tools stay on the device.

In which languages is the site available?
English and Italian. The texts of both languages are generated from the same sources.

14. Glossary

Conjunction
a close approach between two objects in orbit.

CDM
Conjunction Data Message, the CCSDS standard file describing a conjunction.

Collision probability (Pc)
the probability that two objects come closer than their combined hard-body radius.

Entropy (of a value)
a measure of how random a string of characters is; low values are typical of words and placeholders.

Fingerprint
the SHA-256 digest of a file or value; equal inputs always give equal fingerprints.

Held-out set
data never used while writing a method, kept apart to measure it honestly.

Independent implementation
a second program, written separately, that computes the same quantity and is compared with the first.

Monte Carlo
a method that estimates a quantity by repeating random trials and counting outcomes.

Mutation score
the share of deliberately introduced errors that a test suite detects.

Security check
a check run before an action, here before sharing a folder or handing it to an AI agent.

Preregistration
writing the method and the acceptance rule of a measurement before running it.

Receipt
the record of a result: software, version, input fingerprint, changes, checks and limits.

Registry
the S.T.A.R. records of capabilities, benchmarks, evidence and failures, from which states and counts are computed.

Surrogate model
a fast approximate model trained or fitted to reproduce a slower reference.

Tolerance
the largest difference accepted between two results for them to be said to agree.

Twin (digital twin)
: a model of a specific system, kept consistent with references and with its stated domain of validity.

This manual is generated from the site's own texts when the site is built.