

ase-health: User Manual

Version 0.5.1 | From unzipping to your first real scan

- [0. Read this first](#)
- [1. What the tool does](#)
- [2. What you need](#)
- [3. Installing](#)
- [4. Tour with the demo data \(no ASE needed\)](#)
- [5. Connecting to a real ASE server](#)
- [6. Your first real scan](#)
- [7. Reading the reports](#)
- [8. Management summary and CSV](#)
- [9. Configuration](#)
- [10. The history database](#)
- [11. Scheduling daily scans](#)
- [12. Rule reference](#)
- [13. Verify the tool against native ASE \(your first-user test plan\)](#)
- [14. Troubleshooting](#)
- [15. Security and data handling](#)
- [16. Optional AI explanations](#)
- [17. Known limitations and what is unverified](#)
- [18. Command reference](#)
- [19. Glossary](#)
- [20. Licensing](#)
- [Appendix A: every SQL statement the tool can run](#)
- [Appendix B: files in the zip](#)
- [Appendix C: first-user test log](#)

0. Read this first

Who this is for: you, as the first user, testing on your own laptop. It assumes you know ASE well and Python only a little.

Suggested path (about 1 to 2 hours):

1. Section 2: check you have the tools.
2. Section 3: install, run the tests.
3. Section 4: tour the tool on the built-in demo data. No ASE needed.
4. Section 5: connect to a real ASE and run `ase-health ping`.
5. Sections 6 and 13: scan, then check the tool's answers against native ASE commands.
6. Keep notes in the test log (Appendix C). Anything wrong is useful: that is what first-user testing is for.

Licence: a customer build of the tool only runs when a valid licence file is installed in your `.ase-health` folder; otherwise it quits (exit code 3). A 14-day trial is simply a short licence file that we email you on request: install the tool, run `ase-health licence request`, email the text it prints to `hello@solverelay.com`, then `ase-health licence install FILE` (section 20). Developer builds without a built-in key run unrestricted with a warning.

Honest status: the logic is covered by 60 automated tests, but the SQL that talks to ASE has been tested only against a simulated connection, never a real ASE server. Expect some queries to need adjusting on your version. Section 17 lists what is unverified.

1. What the tool does

`ase-health` reads metadata and statistics from an SAP/Sybase ASE server, stores a snapshot, applies a set of rules, and produces reports:

- a technical report (Markdown, HTML or JSON),
- a one-page management summary (HTML or Markdown),
- a CSV for spreadsheets and fix tracking.

It is read-only. It runs only SELECT statements (Appendix A lists every one, and a test enforces this). It never changes the server, never reads your business tables, and sends nothing over the network except to your ASE server. The optional AI explanation (section 16) is off by default.

It is deterministic. The same snapshot always gives the same findings. Every finding carries the evidence behind it.

2. What you need

Item	Needed for	Notes
Laptop: Windows 10/11, Linux or macOS	everything	4 GB free RAM is plenty; the tool itself is tiny
Python 3.11 or newer	everything	Check with <code>python --version</code>
pip	installing	Comes with Python
The zip file <code>ase-health-v0.5.1.zip</code>	everything	
A text editor	editing the config file	Notepad is fine; VS Code is nicer
A web browser	viewing HTML reports	
<i>For a real server only: an ASE server you may legally and safely test against</i>	sections 5 to 6	See 5.1
<i>For a real server only: pyodbc</i>	connecting	Installed with <code>pip install -e "[db]"</code>
<i>For a real server only: an ODBC driver (SAP ASE ODBC driver, or FreeTDS)</i>	connecting	See 5.2
<i>For a real server only: <code>isql</code> (ships with ASE or its client)</i>	creating the login, cross-checking results	
Excel or LibreOffice	opening the CSV	optional

Sections 3 and 4 need only the first six rows. You can do them today without any database.

3. Installing

3.1 Unzip

- **Windows:** right-click the zip, choose *Extract All*, and pick a folder without spaces, e.g. `C:\work`. You should get `C:\work\ase-health`.
- **Linux/macOS:** `unzip ase-health-v0.5.1.zip -d ~/work`

Open a terminal (PowerShell on Windows) in the `ase-health` folder. It contains `pyproject.toml`, `src`, `tests`, `samples` and `docs`.

3.2 Check Python

```
python --version      # Windows (try "py --version" if this fails)
python3 --version     # Linux/macOS
```

You need 3.11 or higher. If you have less or nothing:

- **Windows:** install from python.org and tick *Add python.exe to PATH*.
- **Ubuntu/Debian:** `sudo apt install python3 python3-venv python3-pip`
- **macOS:** `brew install python`

3.3 Create a virtual environment (recommended)

This keeps the tool’s packages separate from the rest of your system.

Windows PowerShell:

```
python -m venv .venv
.\.venv\Scripts\Activate.ps1
```

If PowerShell refuses with “running scripts is disabled”, run this once in that window and retry:

```
Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned
```

Linux/macOS:

```
python3 -m venv .venv
source .venv/bin/activate
```

Your prompt should now start with `(.venv)`. **Re-activate this each time you open a new terminal.**

3.4 Install the tool

```
pip install -e .
```

This installs the tool and its one dependency (`click`). The `-e` means “editable”: if you change a rule in the source, the change is live immediately. This step needs internet access to fetch `click`.

3.5 Check the installation

```
ase-health --help
python -m unittest discover tests
```

You should see the command list, then `Ran 60 tests ... OK`. If `ase-health` is “not found”, your virtual environment is not active; as a fallback use `python -m ase_health.cli --help`.

4. Tour with the demo data (no ASE needed)

If you received a customer build, install your licence first (section 20); every command except `licence` quits without one.

The `samples` folder holds a made-up server called `DEMO_PROD` with deliberate problems, so every rule has something to find.

4.1 Build a demo history

```
python samples/make_demo_history.py
```

This creates `demo_history.db` with six weekly snapshots.

4.2 Your first report

```
ase-health report --server DEMO_PROD --db demo_history.db
```

You get a Markdown report in the terminal: a summary line, “Fix these first”, then one block per finding. Try the other formats:

```
ase-health report --server DEMO_PROD --db demo_history.db --format summary
ase-health report --server DEMO_PROD --db demo_history.db --format summary-html --out summary.html
ase-health report --server DEMO_PROD --db demo_history.db --format html --out full.html
ase-health report --server DEMO_PROD --db demo_history.db --format csv --out findings.csv
```

Open `summary.html` in a browser. That is the page you would give a manager. Use the browser’s *Print to PDF* to make a PDF.

4.3 Look around

```
ase-health history --db demo_history.db    # what is stored
ase-health rules                          # the rule IDs
ase-health queries                        # every SQL statement the tool can run
ase-health config --out my.toml           # all thresholds, with defaults
```

4.4 Try changing a threshold

Edit `my.toml` , delete everything except these lines and save:

```
[txn]
medium_after_hours = 0.2
high_after_hours = 12
```

Then:

```
ase-health report --server DEMO_PROD --db demo_history.db --config my.toml
```

The long-transaction finding drops from HIGH to MEDIUM, and the report header says which settings were customised. Now deliberately break it (change `medium_after_hours` to `medium_after_hour`) and see the clear error. The tool never silently ignores a typo.

Delete `demo_history.db` when you are done. It is only demo data.

5. Connecting to a real ASE server

5.1 Which server to test on

Use a **non-production server** you are allowed to access. Options:

- **A test or development ASE at your employer.** Only with written permission, and note that using employer systems or client data for a side business raises legal and contractual questions. Keep personal-business testing off employer systems unless you have clear approval.
- **Your own ASE on the laptop.** SAP has historically offered a free *Developer Edition* of ASE 16.0 (and the “AS ABAP on ASE” trial). Availability, links and licence terms have changed over the years and I could not confirm the current position, so check the SAP Community and SAP Trials and Downloads pages and read the licence terms before relying on it. Developer editions are typically restricted to non-production use and may limit things such as connection counts.
- **A VM** running ASE if you have an installer and a licence already.

Whatever you use, create a few user databases with some data so the checks have something to look at. Deliberately create the situations the rules look for (see section 13), so you can see the tool catch them.

5.2 Install the ODBC driver and pyodbc

The tool connects through ODBC. Install pyodbc into your virtual environment:

```
pip install -e "[db]"
```

(The quotes matter in PowerShell and zsh.)

You also need an ODBC driver for ASE. Two routes:

Route A: SAP’s ASE ODBC driver (Windows, or Linux if you have the client/SDK installed). It is part of the ASE client/SDK install. Afterwards, open *ODBC Data Source Administrator (64-bit)* on Windows and look at the *Drivers* tab for the exact driver name. It is typically something like “Adaptive Server Enterprise” but **use whatever name you see**. Python and the driver must both be 64-bit (or both 32-bit).

Route B: FreeTDS (free, common on Linux).

```
sudo apt install unixodbc freetds-bin tdsodbc      # Debian/Ubuntu
odbcinst -q -d                                   # should list [FreeTDS]
```

If FreeTDS is not listed, register it in `/etc/odbcinst.ini` :

```
[FreeTDS]
Description = FreeTDS
Driver = /usr/lib/x86_64-linux-gnu/odbc/libtdsodbc.so
```

(the path differs by distribution; `dpkg -L tdsodbc` shows it). On Windows, FreeTDS needs a separate build and is less convenient; prefer Route A there.

5.3 The connection string

Find your server's host and port (default often 5000, but check your interfaces/sql.ini file).

FreeTDS:

```
DRIVER={FreeTDS};SERVER=myhost;PORT=5000;UID=ase_health_ro;PWD=YourPassword;TDS_Version=5.0
```

SAP driver (adjust the driver name to what you saw):

```
DRIVER={Adaptive Server Enterprise};server=myhost;port=5000;db=master;uid=ase_health_ro;pwd=YourPassword
```

If the password contains `;` or `}`, wrap it in braces: `PWD={pa;ss}`.

Put the string in an environment variable so it never appears in command lines or scheduled-task definitions:

Linux/macOS:

```
read -s -p "ASE password: " PW; echo
export ASE_CONN="DRIVER={FreeTDS};SERVER=myhost;PORT=5000;UID=ase_health_ro;PWD=$PW;TDS_Version=5.0"
```

Windows PowerShell (the typed password is visible; use a throwaway test login):

```
$pw = Read-Host "ASE password"
$env:ASE_CONN = "DRIVER={Adaptive Server Enterprise};server=myhost;port=5000;db=master;uid=ase_health_ro;pwd=$pw"
```

The variable lasts for that terminal session. For permanent storage on a real deployment use the operating system's secret mechanisms, not a text file.

5.4 Create a read-only monitoring login

Do not use `sa`. In `isql`, as an administrator:

```
use master
go
sp_addlogin ase_health_ro, "ChangeMe_123!", master
go
-- repeat for every user database you want checked (needed for the
-- statistics and orphan-user checks, which query each database):
use orders
go
sp_adduser ase_health_ro
go
```

Do **not** grant `sa_role`, `sso_role` or `mon_role`. The tool needs none of them today.

What I could not verify: the exact minimum permissions each query needs on every ASE version. Most system tables are readable by `public`, but `lct_admin`, `syslogshold` and the per-database tables may differ by version and configuration. The next step tells you exactly which checks work; any that do not will show as a warning, and you can then grant the minimum needed (or accept the limitation). Please record what you find in the test log. It becomes your “permissions required” page for clients.

5.5 Test the connection: ase-health ping

```
ase-health ping
```

(`ASE_CONN` is read automatically.) Good output:

```
Connected. Adaptive Server Enterprise/16.0 SP03 PL08 ...
databases=7  config_values=...  sa_role_holders=2  db_sizes=7  log_usage=7 ...
All collector queries ran.
```

If some checks cannot run you see `WARNING: ...` lines and exit code 1. That is useful information, not a failure: it shows which SQL needs fixing for your version or which permission is missing. See section 14 for each message.

6. Your first real scan

```
ase-health scan --label TESTBOX1
```

This stores a snapshot in `ase_health.db` in the current folder. The **label** is the key under which history is kept, so use the same label every time for the same server. Then:

```
ase-health report --server TESTBOX1
ase-health report --server TESTBOX1 --format summary.html --out summary.html
```

Run `scan` again a few times over several days. Trend and “since the previous scan” features need history: the growth rule needs at least 3 snapshots spanning 7 days. To see it sooner, lower `trend.min_span_days` in a config file while testing.

7. Reading the reports

Severity: INFO, LOW, MEDIUM, HIGH, CRITICAL.

Priority orders the “Fix these first” list. It is $(\text{severity} + 1) \times \text{exposure} \times 10 / \text{effort}$, where exposure (1 to 3) is how widely the problem reaches and effort (1 to 3) is how hard it is to fix. A quick, widely exposed problem can therefore outrank a harder, higher-severity one. That is deliberate.

A finding contains:

- *Title and rule ID*, e.g. `LOG001` .
- *Detail*: what was found.
- *Why it matters*: plain-English impact, from fixed templates (no AI).
- *Affected*: the databases, logins or tables involved.
- *Fix*: the recommended action.
- *Evidence*: the exact values the rule used, so you can check them by hand.

Header line: scan time, tool version, rules version. Always quote the rules version if you report a problem.

Custom settings and **Collector warnings** appear near the top when relevant. A report with collector warnings may be missing checks.

Time zones: `collected_at` is stored in UTC. Age checks (dump age, open transactions) use the *server’s own clock*, read from `getdate()` , because ASE stores times without a time zone.

8. Management summary and CSV

8.1 Management summary

`--format summary.html` (print to PDF from the browser) or `--format summary` (Markdown for email). It shows:

- overall **RED / AMBER / GREEN** status and a one-line headline,
- the top risks in plain language (default 5),
- **quick wins** (effort: hours) and **needs planning** items, each with effort and a suggested owner,
- change since the previous scan (new and resolved issue types),
- a note of any custom thresholds and incomplete checks.

Default status logic: RED at 1 or more CRITICAL or 3 or more HIGH; AMBER at 1 or more HIGH or 3 or more MEDIUM (or any CRITICAL); otherwise GREEN. All of these are settings under `[summary]` .

Suggested owners come from the `OWNER` table in `src/ase_health/summary.py` . They are guesses; edit them to match your organisation.

8.2 CSV

`--format csv` writes one row per finding. `ase-health export --out fleet.csv` writes the latest findings of every server in the history file.

Column	Meaning
server, collected_at	which server and when

rule_id, severity, priority	what, how bad, fix order
title, affected, why_it_matters, recommended_fix	the finding
effort, exposure	effort label; exposure 1 to 3
evidence_json	the data behind the finding
suggested_owner	from the OWNER table
owner, status, due_date, notes	blank tracking columns (status starts as “Open”)

In Excel use *Data > From Text/CSV* and choose UTF-8 if accented characters look wrong. Cells that start with `=`, `+`, `-` or `@` get an apostrophe added on purpose: object names come from the database, and this stops a malicious name from running as a spreadsheet formula.

9. Configuration

Generate a template, edit, and pass it with `--config` :

```
ase-health config --out my.toml
ase-health report --server TESTBOX1 --config my.toml
```

Delete any line to use its default. `disabled_rules = ["STA001"]` switches a rule off. The tool refuses unknown keys, non-numbers, out-of-range values, and inconsistent pairs (for example a HIGH threshold at or below its MEDIUM threshold).

Setting (section and key in the file)	Default	Meaning
[version] warn_months	18	warn this many months before end of mainstream maintenance
[security] max_sa_role_holders	3	flag when more logins than this hold sa_role
[security] min_password_length	8	flag when ‘minimum password length’ is below this
[backup] max_log_dump_age_days	7	flag user databases with no log dump for this many days
[trend] min_snapshots	3	snapshots needed before growth is projected
[trend] min_span_days	7	days of history needed before growth is projected
[trend] medium_growth_pct_30d	10	MEDIUM if projected 30-day growth reaches this percent
[trend] high_growth_pct_30d	25	HIGH if projected 30-day growth reaches this percent
[log] high_below_free_pct	20	HIGH if free log space is below this percent
[log] critical_below_free_pct	10	CRITICAL if free log space is below this percent
[txn] medium_after_hours	1	MEDIUM if a transaction has been open this many hours
[txn] high_after_hours	4	HIGH if a transaction has been open this many hours
[stats] max_age_days	30	flag table statistics older than this many days (LOW; MEDIUM at 3x)
[stats] min_table_rows	10000	ignore tables smaller than this many rows
[conn] medium_above_pct	80	MEDIUM if connections exceed this percent of the limit
[conn] high_above_pct	95	HIGH if connections exceed this percent of the limit
[summary] top_risks	5	how many risks the management summary lists
[summary] red_if_critical_at_least	1	overall status RED at this many CRITICAL

		findings
[summary] red_if_high_at_least	3	overall status RED at this many HIGH findings
[summary] amber_if_high_at_least	1	overall status AMBER at this many HIGH findings
[summary] amber_if_medium_at_least	3	overall status AMBER at this many MEDIUM findings

Note: DB003 (offline databases) is produced by rule DB002 ; to disable both use "DB002" .

10. The history database

ase_health.db is a single SQLite file in the folder where you run scan (change with --db). It holds one row per snapshot per server.

```
ase-health history          # servers, counts, first and last scan
ase-health prune --days 365  # delete snapshots older than a year
ase-health load old_snapshot.json # import JSON snapshots
```

- **Back it up** by copying the file while no scan is running.
- It contains database, login and table names. Treat it as confidential.
- Re-scanning the same server with a different label starts a separate history.
- Retention currently deletes old snapshots. Weekly roll-ups are not implemented.

11. Scheduling daily scans

Trend features need regular snapshots. Schedule scan , nothing else.

Linux cron (daily at 06:30; put ASE_CONN in a file readable only by the user, e.g. chmod 600 ~/.ase_env):

```
30 6 * * * . ~/.ase_env && cd ~/work/ase-health && .venv/bin/ase-health scan --label PROD1 >> scan.log 2>&1
```

Windows Task Scheduler: create a basic task that runs daily, action *Start a program*:

- Program: C:\work\ase-health\.venv\Scripts\ase-health.exe
- Arguments: scan --label PROD1
- Start in: C:\work\ase-health

Set ASE_CONN as a user environment variable (System Properties > Environment Variables) for the account that runs the task. Be aware that this stores the password in that account’s environment, so use a restricted login on a protected machine.

Use in automation: ase-health report --server PROD1 --fail-on CRITICAL exits with code 1 if any finding is at or above that level.

12. Rule reference

All thresholds below are defaults; see section 9.

ID	What it checks	Data source	Severity logic
VER001	Version against end of mainstream maintenance	@@version and a table in rules/core.py	HIGH if the date has passed; MEDIUM within 18 months; INFO beyond. Versions older than 15.7: CRITICAL
CFG001	Direct updates to system tables enabled	sysconfigures	CRITICAL if on
CFG002	Auditing disabled	sysconfigures	MEDIUM if off
CFG003	Minimum password length below 8	sysconfigures	MEDIUM
SEC001	More than 3 logins hold sa_role	syslogins , sysloginroles , sysssrvroles	HIGH

SEC002	Database users with no server login	per-database <code>sysusers</code> , <code>sysprotects</code> , <code>master..syslogins</code>	LOW; MEDIUM if any has explicit permissions
DB001	<code>trunc log on chkpt</code> on user databases	<code>sysdatabases.status</code> bit 8	HIGH
DB002	Suspect databases (and DB003: offline)	<code>status</code> bit 256; <code>status2</code> bit 16	CRITICAL (suspect); LOW (offline)
BKP001	No log dump for more than 7 days	<code>sysdatabases.dumptrdate</code>	HIGH
TRD001	Allocated database size growing 10%+ per 30 days	<code>sysusages</code> , stored history	MEDIUM; HIGH at 25%+
LOG001	Transaction log nearly full	<code>sysusages</code> log segment, <code>lct_admin</code>	HIGH below 20% free; CRITICAL below 10%
TXN001	Transactions open too long	<code>syslogshold</code>	MEDIUM over 1 hour; HIGH over 4 hours
STA001	Stale statistics on large tables	per-database <code>sysstabstats</code>	LOW; MEDIUM if older than 3x the limit
CON001	Connection limit nearly reached	<code>sysprocesses</code> , <code>number of user connections</code>	MEDIUM above 80%; HIGH above 95%

Known caveats per rule

- **VER001:** the maintenance dates in the code are unverified and sources conflict for 16.0. Check SAP Note 3504083 and edit the `EOMM` table.
- **BKP001:** a heuristic on the last *log* dump date only. It does not check full dumps or whether restores work. Databases where log dumps do not apply may be flagged: confirm against your backup tooling.
- **TRD001:** measures **allocated** size (`sysusages`), not used space. It detects repeated space additions, but will not notice data growing inside a fixed-size database. Treat it as experimental; improving it (using used-space figures) is planned.
- **LOG001:** databases with data and log on the same segment give meaningless free-page numbers. Review those by hand.
- **TXN001:** legitimate long batch jobs will trigger it. Raise the thresholds or disable the rule if that is noise for you.
- **STA001:** based on the statistics row’s modification date, not on how much data has changed since.
- **CON001:** one sample at scan time. A brief spike will not show, and a transient peak might.

13. Verify the tool against native ASE (your first-user test plan)

Run `ase-health scan` , then for each row run the native command in `isql` and compare with the **Evidence** line of the report. Record results in Appendix C.

Rule	Cross-check in isql	Make it fire (on a test server only)
VER001	<code>select @@version</code>	none needed
CFG001 to CFG003	<code>sp_configure "allow updates"</code> , <code>sp_configure "auditing"</code> , <code>sp_configure "minimum password length"</code>	change the setting temporarily, rescan
SEC001	<code>sp_displaylogin <login></code> for each reported name	<code>sp_role "grant"</code> , <code>sa_role</code> , <code><test_login></code>
SEC002	<code>sp_helpuser</code> in each database	drop a login that still has a user in a test database
DB001	<code>sp_helpdb <db></code> (look at the status/options)	<code>sp_dboption <db></code> , <code>"trunc log on chkpt"</code> , <code>true</code>
DB002/DB003	<code>sp_helpdb</code>	take a test database offline
BKP001	<code>select name, dumptrdate from master..sysdatabases</code>	do not dump a test database’s log for 8 days, or lower <code>max_log_dump_age_days</code> to 0
TRD001	<code>select db_name(dbid), sum(size) from master..sysusages group by dbid</code>	extend a test database repeatedly and scan weekly
LOG001	<code>sp_spaceused</code> <code>syslogs</code> inside the database	fill a small test log without dumping it
TXN001	<code>select * from master..syslogshold</code> , <code>sp_who</code>	open a transaction (<code>begin tran</code>) and leave it
STA001	<code>optdiag statistics <db>..<table>< code=""></table><></code>	create a big table and never update its statistics

CON001	<code>sp_monitorconfig "number of user connections"</code>	open many sessions, or lower the connection limit on a test server
--------	--	--

For each row record: did the tool’s number match ASE’s? If not, copy the SQL from `ase-health queries` , run it in `isql` , and note the difference (wrong column, different units, permission). Those notes are exactly what is needed to fix the collectors.

14. Troubleshooting

Symptom	Likely cause and fix
<code>python</code> not recognised / <code>python3: command not found</code>	Python missing or not on PATH. Reinstall and tick <i>Add to PATH</i> (Windows); try <code>py</code> instead of <code>python</code>
pip says the package requires a newer Python	You have Python older than 3.11. Install a newer one and recreate the virtual environment
<code>ase-health: command not found</code>	The virtual environment is not active. Activate it again, or use <code>python -m ase_health.cli ...</code>
PowerShell: running scripts is disabled	<code>Set-ExecutionPolicy -Scope Process -ExecutionPolicy RemoteSigned</code> , then activate again
pyodbc not installed	<code>pip install -e ".[db]"</code> inside the activated environment
<code>libodbc.so.2: cannot open shared object file</code>	Install unixODBC: <code>sudo apt install unixodbc</code>
<code>IM002 Data source name not found and no default driver specified</code>	The driver name in <code>DRIVER={...}</code> does not match an installed driver. List them (<code>odbcinst -q -d</code> , or the Drivers tab on Windows) and copy the exact name. Also check 32 vs 64-bit mismatch
Login failed	Wrong user/password, or the login cannot reach the default database. Test the same login in <code>isql</code>
Timeout or cannot connect	Wrong host or port, firewall, or ASE not running. Confirm the port in the <code>interfaces/sql.ini</code> file
FreeTDS protocol errors	Add <code>TDS_Version=5.0</code> to the connection string
<code>WARNING: <check>: ... Invalid column name / Invalid object name / unknown function</code>	The SQL differs on your ASE version. Run <code>ase-health queries</code> , test that statement in <code>isql</code> , and note the exact error for fixing
<code>WARNING: ... permission denied</code>	The monitoring login lacks SELECT or EXECUTE on that object. Grant the minimum needed, or accept that the rule will not run
<code>WARNING: server_time</code>	<code>getdate()</code> failed; age checks fall back to UTC and may be off by your time-zone offset
<code>WARNING: <db>: skipped (unusual database name)</code>	The database name has characters the tool refuses to put into SQL. Rename it or ignore that database
<code>No stored snapshots for X</code>	The <code>--server</code> label does not match. Run <code>ase-health history</code> to see labels, and check <code>--db</code> points to the right file
<code>Unknown setting 'x.y' or other config errors</code>	Read the message; run <code>ase-health config</code> to see valid keys
Growth rule never fires	It needs 3 snapshots over 7 days, and it measures allocated size only
Excel shows odd characters	Import with <i>Data > From Text/CSV</i> , encoding UTF-8
Cells start with an apostrophe	By design; it prevents formula injection
<code>No licence file found / exit code 3</code>	Install a licence: section 20
<code>The licence file is not valid: ... different installation</code>	The file was issued for another Installation ID (or you deleted <code>.ase-health</code>). Run <code>licence request</code> again

If you hit something not listed, copy the full error text and the output of `ase-health ping` . Remove server and login names first if you share it.

15. Security and data handling

- **Read-only:** every SQL statement is a SELECT (Appendix A). A test fails if that ever changes.
- **Least privilege:** use a dedicated login with no roles. Never use `sa` .

- **Local only:** the tool contacts your ASE server and nothing else. The only exception is the optional `--explain` option (section 16), which is off unless you choose it.
- **What is stored:** database names, login names, table names, configuration values and counts. No business data. These names are still sensitive, because they show where the weaknesses are. Protect the history file, the reports and the CSVs.
- **Passwords:** keep them in the `ASE_CONN` environment variable, not in scripts or command lines.
- **When used for clients:** get their written agreement on what is collected, where it is stored and how long you keep it. Give them the `ase-health queries` output so their security team can review exactly what runs. I am not a lawyer; have the agreement reviewed by one.

16. Optional AI explanations

Off by default. `report --explain` sends each finding to a language model API for a three-sentence plain-English explanation. Object names are replaced with placeholders (`OBJ_1` ...) first, but the finding text still goes to an external provider, so only use it where the client's policy allows. You need `pip install -e ".[ai]"`, an API key in the provider's environment variable, and `ASE_HEALTH_MODEL` set to a current model name. The rules engine never depends on it, and nothing in this manual's workflow requires it. This path is untested against the live API.

17. Known limitations and what is unverified

- The SQL has never run against a real ASE. Treat the first scan as a test of the queries.
- Minimum permissions for the monitoring login are unconfirmed.
- Rule thresholds are sensible starting points, not industry standards. Tune them for your estates.
- End-of-maintenance dates need confirming (SAP Note 3504083).
- TRD001 measures allocated size, not used space.
- BKP001 does not verify full dumps or restores.
- Only metadata checks exist: there is no query performance (MDA), replication or HA coverage yet.
- The licence scheme is deliberately light (section 20): anyone who edits the source can bypass it.
- English only; no graphical interface.

18. Command reference

Command	Purpose
<code>scan --label NAME [--out file.json] [--db file]</code>	collect from a live server into history
<code>ping</code>	test connection and all queries; stores nothing
<code>report --server NAME (or --snapshot FILE)</code>	render a report. Options: <code>--format</code> (md, html, json, csv, summary, summary-html), <code>--out FILE</code> , <code>--config FILE</code> , <code>--fail-on</code> (HIGH or CRITICAL), <code>--explain</code> , <code>--db FILE</code>
<code>export [--out file.csv] [--config FILE]</code>	latest findings of all servers, one CSV
<code>history</code>	list servers and snapshot counts
<code>prune --days N</code>	delete older snapshots
<code>load FILES...</code>	import JSON snapshots
<code>config [--out file.toml]</code>	print or write the settings template
<code>rules</code>	list rule IDs
<code>licence request</code> / <code>licence install FILE</code> / <code>licence status</code>	get, install and check a licence (<code>license</code> also works); the only commands that run without one
<code>queries</code>	print every SQL statement the tool can run

Environment variables: `ASE_CONN` (connection string); `ASE_HEALTH_MODEL` (only for `--explain`). **Exit codes:** 0 success; 1 an error, or `--fail-on` matched, or `ping` found warnings; 3 no valid licence.

19. Glossary

- **ASE:** SAP Adaptive Server Enterprise, formerly Sybase.
- **EOMM:** end of mainstream maintenance: after it the vendor stops issuing routine fixes.
- **Snapshot:** one stored set of metadata from one server at one time.
- **Finding:** one problem the rules detected, with its evidence.
- **Collector:** the code that runs the SELECTs and builds a snapshot.
- **Rule:** a function that turns a snapshot (and history) into findings.
- **Priority:** the fix-order score (section 7).
- **RAG:** red, amber, green overall status.

20. Licensing

The rule: every command except `ase-health licence ...` quits with exit code 3 unless a valid, unexpired licence file is installed at `.ase-health/licence.json` in your home folder (Windows: `C:\\Users\\you\\.ase-health`). If the folder or the file is missing, altered, expired, for another product, or issued for a different installation, the tool quits and tells you why.

Getting a licence (manual, by email):

1. Install the tool. Run `ase-health licence request`. It creates the `.ase-health` folder and prints a short text containing your **Installation ID**.
2. Email that text to hello@solverelay.com, with your name or company and number of servers.
3. You receive a small signed JSON file. A trial is a short licence, normally 14 days.
4. Install it, then check:

```
ase-health licence install licence-1234567890.json
ase-health licence status
```

Renewal works the same way: install the newer file over the old one.

While licensed: a reminder appears on the error stream (it does not disturb CSV or JSON output) during the last 14 days.

Commands that work without a licence: `licence request`, `licence status`, `licence install`, and `--help`.

Offline: licence files are verified with a public key built into the tool. No internet or licence server is needed.

Important: do not delete the `.ase-health` folder; the tool will quit until you install the licence again. Keep a copy of your licence file. If a licence is bound to your Installation ID and you lose the folder, a new request is needed (the ID changes).

Limits to know about: this is a light, manual scheme for the testing stage. Setting the system clock back does not revive an expired licence, but anyone who edits the source code can remove the check. A licence file that is not bound to an Installation ID can be copied to another machine.

Environment variable: `ASE_HEALTH_HOME` moves the folder elsewhere (it must still hold `licence.json`).

Developer builds: if no public key has been built in (see `owner-tools/README.md`), the tool runs with a “DEVELOPER BUILD: licence not enforced” warning and licence files cannot be installed.

Appendix A: every SQL statement the tool can run

Also printed by `ase-health queries`. Statements containing `{db}` are run once per user database after the name is validated.

```
-- version
select @@version;
```

```
-- server_time
select getdate();
```

```
-- databases
select name, status, status2, dumptrdate from master..sysdatabases;
```

```
-- config
select c.name, cu.value from master..sysconfigures c, master..syscurconfigs cu where c.config = cu.config;
```

```
-- db_sizes
select db_name(dbid), sum(size) * @@maxpagesize / 1048576.0 from master..sysusages group by dbid;
```

```
-- sa_role_holders
select l.name from master..syslogins l, master..sysloginroles r, master..syssrvroles s where l.suid = r.suid and r
= s.srid and s.name = 'sa_role';
```

```
-- log_usage
select db_name(dbid), sum(size), lct_admin('logsegment_freepages', dbid) from master..sysusages where segmap & 4 =
group by dbid;
```

```
-- open_transactions
select db_name(dbid), spid, starttime from master..syslogshold where name <> '$replication_truncation_point';
```

```
-- connections
select count(*) from master..sysprocesses where suid > 0;
```

```
-- table_stats
select o.name, t.rowcnt, t.moddate from {db}..systabstats t, {db}..sysobjects o where t.id = o.id and t.indid in (
and o.type = 'U';
```

```
-- orphan_users
select u.name, case when exists (select 1 from {db}..sysprotects p where p.uid = u.uid) then 1 else 0 end from
{db}..sysusers u where u.suid > 0 and u.suid not in (select suid from master..syslogins);
```

Appendix B: files in the zip

ase-health/	
pyproject.toml	package definition
README.md	short quick start
docs/MANUAL.*	this manual
src/ase_health/	
cli.py	commands
collectors.py	the SELECTs and snapshot builder
connector.py	ODBC connection
config.py	thresholds and validation
store.py	SQLite history
rules/core.py	the rules (your expertise goes here)
summary.py	management summary
export.py	CSV export
report.py	technical report
explain.py	optional AI layer
licensing.py, ed25519.py	trial clock and signed licence verification
tests/	60 automated tests
samples/	demo snapshot, demo history builder, config template

Appendix C: first-user test log

Copy this table into a spreadsheet and fill it in as you go.

#	Test	Command or action	Expected	Actual	Pass / Fail	Notes
1	Install	<code>pip install -e .</code>	no errors			
2	Unit tests	<code>python -m</code>	60 OK			

		<code>unittest discover tests</code>				
3	Demo report	section 4.2	RED status, 14 findings			
4	Config typo	section 4.4	clear error			
5	ODBC driver listed	<code>odbcinst -q -d / Drivers tab</code>	driver visible			
6	Connection	<code>ase-health ping</code>	connected			
7	Every query runs	<code>ping</code>	no warnings			
8	Scan stores	<code>scan</code> , then <code>history</code>	1 snapshot			
9 to 20	One row per rule in section 13		tool matches ASE			
21	Summary page	<code>--format summary-html</code>	readable, correct			
22	CSV opens in Excel		columns correct			
23	Second scan next day	<code>scan</code> again	2 snapshots, “since previous” line			
24	Permission list		minimum grants written down			
25	Quits without licence	any command on a customer build	message and exit code 3			
26	Request text	<code>ase-health licence request</code>	Installation ID shown			
27	Install licence	<code>ase-health licence install FILE</code> , then <code>licence status</code>	LICENSED			
28	Delete folder	remove <code>.ase-health</code> , run any command	quits again			