pravidha

Pramana

Evidence Guide

Governance evidence for Snowflake — classification, policy coverage, drift, audit packs

Document	Pramana — Evidence Guide
Version	v1.0
Date	17 November 2026
Author	Pravidha — Snowflake Select Service Partner
Status	Released
Classification	Public

© 2026 Pravidha. This document and the software it describes are Pravidha intellectual property.

Contents

What Pramana means by evidence
What a pack holds
Reading a pack, part by part
The thirteen controls
What the statuses mean
How controls map to clauses
    UK GDPR (GDPR_UK)
    DORA (DORA)
    ISO 27001 (ISO27001)
How far a pack can be relied on
Exemptions
A pack for a past date
Taking a pack away
Checking that a pack has not been altered
Feeding a GRC tool
Answering an audit request

© 2026 Pravidha · Public

For data protection officers, compliance and risk leads, internal audit and external auditors. It explains what an evidence pack is, how to read one, how far it can be relied on, and how to check that a copy has not been altered. No Snowflake knowledge needed for the first half. About fifteen minutes.

What Pramana means by evidence

Pramana examines a Snowflake account every night and records what it finds. An **evidence pack** is the record of one such examination, written for one framework: UK GDPR, DORA, ISO 27001, or one you define.

A pack is useful as evidence because of four properties:

- **It is dated.** It describes the account as it stood at one moment, and says which moment.
- **It is specific.** It names every control, every object that fails it, and the clause that failure touches.
- **It is complete about exceptions.** Everything your organisation chose to accept is listed, with the reason, the approver and the expiry.
- **It is sealed.** It carries a fingerprint of its own content, so that a copy can be checked.

Packs are never edited. Each run adds new ones; the old ones stay as they were until your retention period removes them.

What a pack holds

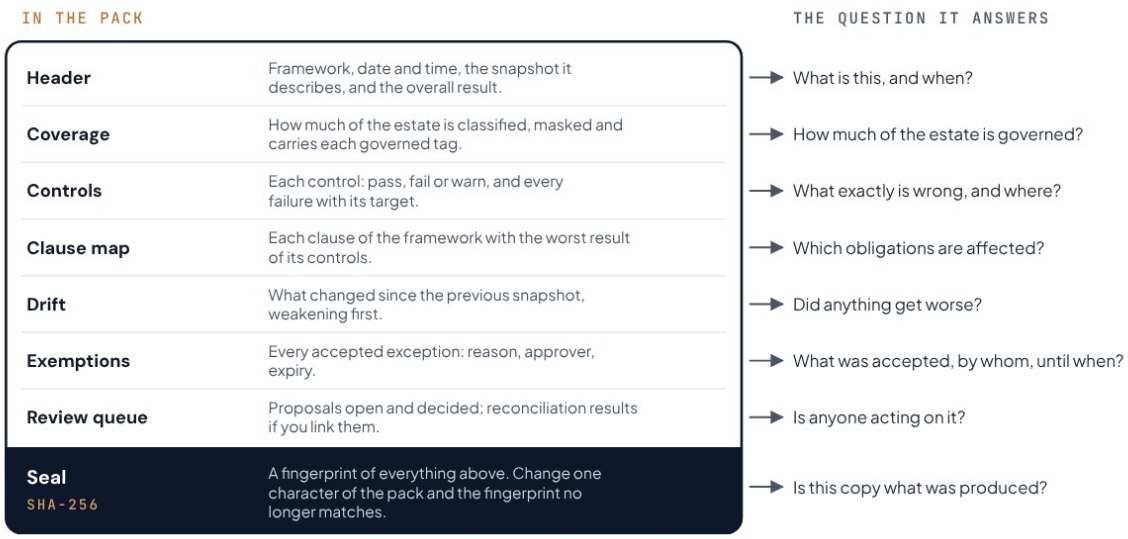


Figure 1 · *What an evidence pack holds, and the question each part answers*

Every pack exists in two forms with the same content. The **readable pack** is a document for people. The **JSON pack** is the same information in a form that GRC tools can load, and it is the form the seal covers.

Reading a pack, part by part

Header. The framework, when the pack was generated, an identifier, and the overall result. It also names the snapshot the pack describes: when it was taken, and how many tables and columns in which databases. Read this first: it tells you the scope the rest of the pack speaks for.

© 2026 Pravidha · Public

Coverage. How many columns were examined, how many carry a classification, how many were detected as sensitive and how many of those are still unlabelled, and what share of labelled columns is masked. For each governed tag, such as retention, it shows how many of the tables that need a value have one, and which values are in use.

Controls. One line per control with its status and counts, then, for every control that did not pass, the list of objects that failed it and why. Up to two hundred failures per control are listed in full by default; if there are more, the pack says how many were left out.

Framework mapping. Each clause of the framework, the controls mapped to it, and a status. A clause takes the worst status of its controls.

Drift. What changed between this snapshot and the one before it. Changes that weaken the position, such as a masking policy removed or read access granted on a sensitive table, are listed first and marked.

Exemptions in force. Every exception your organisation has accepted: the object, the control it is excused from, the reason given, who approved it and when it expires.

Review queue. How many proposed classifications are waiting for a decision, and how many have been approved or rejected.

Seal. A SHA-256 value at the foot of the pack.

The thirteen controls

Each control asks one question. Its identifier never changes, so it can be cited in an audit report.

Control	The question it asks	Grade if it does not hold
CLS-01	Does every column that looks sensitive carry a classification?	Fail
CLS-02	Is every classification a value from the agreed taxonomy?	Fail
CLS-03	Where Snowflake's classifier has an opinion, does it agree with the label?	Warning
CLS-04	Has each sensitive table been classified recently?	Warning
POL-01	Is every classified column masked by the policy chosen for its category?	Fail
POL-02	Does every table that needs a row access policy have it?	Fail
POL-03	Is masking applied only to columns that are classified?	Warning
POL-04	Do the policies the organisation chose exist, and are they in use?	Fail
ACC-01	Are sensitive tables free of read access by broad roles?	Fail
ACC-02	Is every sensitive table owned by an accountable role?	Warning
RET-01	Does every table that needs one carry a retention class?	Fail
RET-02	Is every retention class one the organisation defined?	Fail
DQ-01	Do sensitive tables reconcile? (Only where Vigil is linked)	Fail

© 2026 Pravidha · Public

Your organisation can re-grade a control between fail and warning, switch one off, and add controls of the same kind for other tags it governs by, such as data residency or ownership. A pack always shows the grades in force when it was produced.

What the statuses mean

Status	Applies to	Meaning
`PASS`	A control	Everything examined met the control
`FAIL`	A control or one object	The control is not met, and it is graded as a failure
`WARN`	A control or one object	The control is not met, and it is graded as a warning
`EXEMPT`	One object	The object does not meet the control and an exemption is in force. It is counted and listed, and it does not fail the control
`SKIPPED`	A control	The control could not be evaluated, and its note says why. For example, no retention tag has been named, or Vigil is not linked

The **overall** result of a pack is `FAIL` if any control fails, `WARN` if none fails and any warns, and `PASS` otherwise.

A skipped control is neither a pass nor a failure. A pack with skipped controls is saying that part of the picture is missing, and that is worth asking about.

How controls map to clauses

The mappings below ship with Pramana. They are Pramana's reading of which technical control is evidence for which clause, offered as a starting point. Your organisation can change them, and the pack always prints the mapping it used.

A pack is evidence about technical controls on one platform. It supports a compliance position. It is not a legal determination, and it says nothing about obligations that are not technical, such as lawful basis or notices.

UK GDPR (`GDPR_UK`)

Clause	Title	Controls
Art. 5(1)(d)	Accuracy	CLS-03, DQ-01
Art. 5(1)(e)	Storage limitation	RET-01, RET-02
Art. 5(1)(f)	Integrity and confidentiality	CLS-01, ACC-01
Art. 24	Responsibility of the controller	CLS-04, ACC-02
Art. 25	Data protection by design and by default	POL-01, POL-03
Art. 30	Records of processing activities	CLS-01, CLS-02, RET-01
Art. 32	Security of processing	POL-01, POL-02, POL-04, ACC-01

© 2026 Pravidha · Public

DORA (DORA)

Clause	Title	Controls
Art. 8	Identification (classification of information assets)	CLS-01, CLS-02
Art. 9	Protection and prevention	POL-01, POL-02, POL-04
Art. 9(3)(b)	Integrity and availability of data	DQ-01
Art. 9(4)(c)	Access management policies	ACC-01, ACC-02

ISO 27001 (ISO27001)

Clause	Title	Controls
A.5.9	Inventory of information and other associated assets	ACC-02
A.5.12	Classification of information	CLS-01, CLS-03
A.5.13	Labelling of information	CLS-02
A.5.15	Access control	POL-02, ACC-01
A.5.33	Protection of records	DQ-01, RET-02
A.8.10	Information deletion	RET-01
A.8.11	Data masking	POL-01, POL-03, POL-04

Packs are produced for UK GDPR and DORA unless your administrator changes the list. ISO 27001 is mapped and is switched on by adding it to that list; a framework of your own is added the same way. The Reference has the detail.

How far a pack can be relied on

A pack is a faithful record of what Pramana could see. These are the limits of what it could see.

- **It describes metadata, a few hours old.** Pramana reads Snowflake's account metadata, which is typically up to two or three hours behind the account. A change made shortly before a run appears in a later pack.
- **Newly attached policies can take longer to appear.** When an account attaches policies for the first time, Snowflake can take many hours to list them. A pack produced in that window reports those columns as unprotected. Pramana notes this in the affected controls when it can tell.
- **It proves a policy is attached, not what the policy does.** POL-01 and POL-02 confirm that the chosen policy is in place. Whether the policy's rule is the right one is a separate review.
- **It speaks only for what is in scope.** Databases that were not placed in scope are not in the pack. The header names the ones that are.
- **Detection has two inputs.** A column is recognised as sensitive when Snowflake's classifier has examined its table, or when its name matches a rule. CLS-04 shows where the classifier has not been run.
- **Long lists are shortened.** Beyond the configured number of failures per control, the pack gives the count and not every item.

© 2026 Pravidha · Public

Exemptions

An exemption is your organisation's decision to accept a finding: this column is a one-way hash and need not be treated as personal data; this table is masked on purpose although it is not classified. Each exemption names the object and the control, and carries a reason, an approver and, normally, an expiry date.

Exempted objects are counted and listed in every pack. They never disappear from view, and an exemption that has expired stops applying without anyone having to remove it.

When a data steward rejects a proposed classification, Pramana treats that as a standing decision: the column is counted as exempt under the control from then on, and the rejection is counted in the review figures. A rejection is not itemised with a reason, so where the reason matters to an auditor, record an exemption as well.

A pack for a past date

Because packs are kept, the position on a past date is a lookup, not a reconstruction. Anyone with the auditor role can run:

```
SELECT PACK_ID, GENERATED_AT, OVERALL, PACK_SHA256
  FROM PRAMANA.GOV.GOV_EVIDENCE_PACK
 WHERE FRAMEWORK = 'GDPR_UK'
   AND GENERATED_AT::DATE = '2026-09-30';
```

Packs stay until an administrator runs the retention purge. The purge removes what is older than 1,100 days, about three years, unless your organisation has set another period, and it never removes the two most recent snapshots.

Taking a pack away

In the console, open **Evidence**, choose the pack, and use **Take it away**. Or in a worksheet, with the pack's identifier:

```
-- the readable pack
SELECT PACK_MD FROM PRAMANA.GOV.GOV_EVIDENCE_PACK WHERE PACK_ID = '<pack id>';

-- the JSON pack, as text: this is the form the seal covers
SELECT TO_JSON(PACK) FROM PRAMANA.GOV.GOV_EVIDENCE_PACK WHERE PACK_ID = '<pack id>';

-- the seal, as the installation holds it
SELECT PACK_SHA256 FROM PRAMANA.GOV.GOV_EVIDENCE_PACK WHERE PACK_ID = '<pack id>';
```

Save the JSON as a file, for example `pack.json`. In Snowsight, run the second statement and download the result. From a terminal, the Snowflake CLI writes the file in one step:

```
# the JSON pack of one pack, saved as pack.json (put the pack's own identifier in place of <pack id>)
snow sql --format json -q "SELECT TO_JSON(PACK) AS PACK FROM PRAMANA.GOV.GOV_EVIDENCE_PACK WHERE PACK_ID = '<pack
id>';" > pack.json
```

The identifier has to be a real one, from the first query in this section or from `GOV_EVIDENCE_LATEST_V`. A statement that matches no pack still writes a file, an empty one, and the check below will tell you it holds no pack.

© 2026 Pravidha · Public

Checking that a pack has not been altered

The seal is a SHA-256 computed over the JSON pack. Anyone holding a copy can recompute it. Pravidha publishes a small script for this, `verify_pack.py`, on the documentation page. It needs Python 3 and nothing else, and it reads only the file you give it.

```
# recompute the seal from your copy, and compare it with the value the installation holds
python3 verify_pack.py pack.json --expect <PACK_SHA256>
```

It prints the seal the pack carries, the seal it computed and the value you supplied, and one of two verdicts:

```
SEAL HOLDS: this copy says what Pramana wrote, and it is the pack the installation holds
SEAL DOES NOT HOLD: this copy differs from what Pramana wrote, or it is not the pack that value belongs to
```

What a seal that holds tells you:

- the copy has the content Pramana wrote, to the character;
- with `--expect`, the copy is the pack the installation holds under that identifier. Ask for that value from someone who can read the installation, separately from whoever gave you the file.

What it does not tell you:

- that the readable pack you were handed matches. The seal covers the JSON. The readable pack is produced from it and prints the same seal, so compare a readable copy against the JSON you have verified;
- that the account was configured sensibly. Scope and exemptions are choices, and the pack shows them so that they can be questioned.

For anyone who wants to reproduce the calculation without the script: remove the `integrity` block from the pack; write every whole number without a decimal point; serialise as JSON with keys sorted, `", "` between items and `": "` between a key and its value, and non-ASCII characters escaped; take the SHA-256 of the UTF-8 bytes. The script's header states the same.

Feeding a GRC tool

Pramana exposes one view built for this purpose, `PRAMANA.CORE.PRAMANA_EVIDENCE_V`: one row per pack, with coverage figures, the count of weakening changes, open proposals, the controls and the clause map, and the pack itself. Grant the tool's service role `PRAMANA_EVIDENCE_READER`; it can read that view and nothing else.

Answering an audit request

1. **Agree the date and the framework.** Find the pack with the query under "A pack for a past date".
2. **Hand over both forms** of the pack, and the seal.
3. **Let the auditor verify** the JSON with `verify_pack.py`, using a seal value they obtained themselves.
4. **Walk through the exemptions.** They are the part of the pack that records decisions, and the part most often asked about.
5. **Where a control failed,** show the later pack in which it passes. The drift section of that pack shows the change that closed it.

Questions about this guide: **info@pravidha.com**.

© 2026 Pravidha · Public