

Control Register Template

A control register is the single list that tells you what you have promised to control, who owns it, how it is tested, and when it last passed. This template gives you the columns, the conventions, and the review rhythm.

The columns that matter

Registers fail by being either too thin to be useful or so wide that nobody maintains them. These columns are the working minimum.

- Control ID and short name — stable, referenced everywhere else.
- Source — the policy, standard, regulation, or framework objective it satisfies.
- Risk addressed — plain language, one sentence.
- Control type — preventive, detective, or corrective; manual, automated, or hybrid.
- Process and activity — where in the flow it actually applies.
- Owner and approver — named people, not teams.
- Test method and frequency — how you know it works and how often you check.
- Last tested, result, and evidence link.
- Status — effective, degraded, failed, or retired.

Conventions that keep it alive

A register only survives if the rules for filling it in are boring and consistent.

- One control, one row. If a row has 'and' in it twice, it is two controls.
- Never delete a retired control — mark it retired with a date and a reason.
- If a control cannot be tested, it is a policy statement; move it out of the register.
- Owners are individuals. A team cannot be called at 7am.

Linking the register to live data

The difference between a register that is read and one that is filed is whether the test column is automatic. Where a control maps to a governance rule over an event log, the last-tested date and result should be written by the platform, not by a person.

Manual tests still belong in the register — just mark them clearly so you can see what proportion of your control environment depends on someone remembering.

The review rhythm

Monthly: exceptions and failed tests only. Quarterly: ownership, thresholds, and anything degraded. Annually: full re-walk against the source policies, including retirement of controls that no longer address a live risk.

Video walkthrough — key points

1. Control Register Template

- What you control
- Who owns it
- When it last passed

A control register is the single list that tells you what you have promised to control, who owns it, how it is tested, and when it last passed. This template gives you the columns, the conventions, and the review rhythm.

2. The columns

- ID · source · risk · type
- Process · activity · owner
- Test method · last result · status

Registers fail by being too thin to be useful, or so wide that nobody maintains them. The working minimum is: a stable identifier, the source policy or objective, the risk in one sentence, the control type, the process and activity it applies to, a named owner and approver, the test method and frequency, the last test and its evidence, and a status.

3. Conventions

- One control, one row
- Retire, never delete
- Untestable means it is a policy
- Owners are people

A register only survives if the filling-in rules are boring and consistent. One control, one row: if a row contains the word and twice, it is two controls. Never delete a retired control, mark it retired with a date and a reason. If a control cannot be tested, it is a policy statement. And owners are individuals, because a team cannot be called at seven in the morning.

4. Link it to live data

- Automatic last-tested dates
- Manual tests clearly flagged

The difference between a register that is read and one that is filed is whether the test column fills itself in. Where a control maps to a governance rule over an event log, the last tested date and result should be written by the platform. Keep manual tests, but flag them, so you can see how much of

your control environment depends on someone remembering.

5. Review rhythm

- Monthly: exceptions
- Quarterly: owners and thresholds
- Annually: full re-walk

Then set the rhythm. Monthly, look at exceptions and failed tests only. Quarterly, review ownership, thresholds, and anything degraded. And annually, re-walk the whole register against the source policies, retiring the controls that no longer address a live risk.

© Automated Governance Systems. Shared for use inside your organisation. For a walkthrough against your own processes, book a session at ags-apip.lovable.app/book-demo.