MONITORING TRUTH & FLEET RECONCILIATION SERIES · EC-WP-1104

A Machine May Draft It

Switching order integrity: expected state, the equipment the diagram omits, and the one field a drafter must never write.

Machine drafting changes who prepares a switching order. It must not change who approves it or who executes it. Five integrity requirements, each with the failure that motivates it.

Reference Five of Eight in the Monitoring Truth Series

The Procedure Layer

energycomplianceinc.com **OCTOBER 2026**

Disclaimer & Legal Notice

This publication is the work product of Energy Compliance, Inc. and is provided for general information and educational purposes only. It does not constitute legal advice, regulatory guidance, engineering advice, or formal interpretation of any NERC Reliability Standard, FERC order, RTO or ISO protocol, or state commission rule.

This publication may not be used or cited as authority in any audit, enforcement, regulatory, or legal proceeding.

Nothing in this document is a switching procedure, and nothing in it should be used as one. Switching is governed by the operating entity's own approved practices.

On the absence of regulatory citations

This series carries no requirement text, no requirement number, no standard version, and no effective date, and it cites no order, protocol section, or rule. The omission is deliberate. The subject of the series is how a monitoring platform represents measured reality, a question that comes before, and stands apart from, which requirement is being satisfied. Regulatory standard families are named only where the connection to a specific finding is material, and they are named as families, never as citations. Nothing here should be read as a statement of what any standard requires. A reader who needs operative language must obtain it from the current enforceable source for the relevant registered entity and audit period. The technical sources this edition does cite, the PostgreSQL documentation behind the diagnostic queries, are listed under Sources and Authorities.

On the findings

Every defect described here was measured against a live production monitoring platform. The observations are anonymized: no client, site, site code, counterparty, vendor, product, asset tag, or person is identified, and figures are reported at a level of aggregation that does not permit re-

identification. Some of the defects described are in RenewOps, the platform Energy Compliance builds. Those are marked as ours and written up on the same terms as the rest.

On currency

A measurement is a statement about one system at one moment. The register figures in Chapters 3, 6, 7 and 9 were measured on August 24, 2026, and the Chapter 9 library figures on August 26, 2026. The Chapter 5 findings come from the 2026 assessment described in the Foreword. They are reproduced because the shape of the failure is instructive, not because they describe the current state of any system. This edition was reviewed on October 17, 2026. The prose and the diagnostic queries were revised; the August 2026 figures were not re-measured and are presented as they were measured. Energy Compliance, Inc. does not represent that any condition described here persists, or that any condition absent here does not exist elsewhere.

Non-affiliation

NERC, FERC, and the names of Regional Entities are referenced here for descriptive purposes only. Energy Compliance, Inc. is not affiliated with, sponsored by, or endorsed by the North American Electric Reliability Corporation, the Federal Energy Regulatory Commission, or any NERC Regional Entity. PostgreSQL, Supabase and PostgREST are named only to identify the database features the diagnostic queries use. RenewOps and WattOps Compliance are products of Energy Compliance, Inc.

Energy Compliance, Inc. assumes no responsibility or liability for errors, omissions, or outcomes resulting from the use of this publication. Compliance obligations depend on facts and circumstances unique to each registered entity, and the reader should consult qualified counsel and subject-matter experts when making compliance decisions.

Foreword

This professional reference is one of a series Energy Compliance, Inc. publishes for registered entities and the people who run their compliance programs.

I have spent more than thirty years on every side of the bulk electric system. I have operated control centers as a Reliability Coordinator, Transmission Operator, and Power System Operator. I have audited grid facilities and signed off on findings as a senior compliance auditor. I have worked enforcement matters from inside the regulator's process. For the last several years I have advised registered entities directly through the firm I founded.

Here is the thing about a screen. In a control room the screen is not a report on the system. For the eight hours you sit in front of it, the screen *is* the system. You act on it. You hand over on it. You write it into the log, the log becomes the record, and the record becomes the evidence. Everything downstream inherits whatever the screen got wrong, and nothing downstream is equipped to notice.

That is the subject of this series. Not whether a fleet is performing, but whether the number on the screen is a measurement of the fleet at all.

The series came out of an assessment. In 2026 we pointed a defect battery at a live production monitoring stack running a multi-gigawatt utility-scale PV and storage fleet and wrote up everything it found: twenty-plus verified defects, each with a source and a timestamp. What I did not expect was the shape of the result. The defects were not a

list of unrelated bugs. They were six recurring structures, and every one of them was a system reporting an absence as a value. A missing irradiance reading rendered as zero. A denominator that shrank as equipment failed, so a half-dead site scored a perfect hundred. An ingest timestamp wearing an event timestamp's label. A reading four hundred days old displayed as current. A revenue meter an order of magnitude above nameplate rendered live, because nothing in the system had a nameplate to check it against. A voltage band seeded from a plant's own history, indistinguishable on screen from a schedule issued by the entity of record.

None of that was a vendor defect, and I want to be careful here. The easy version of this series is a hit piece on monitoring platforms, and that version would be wrong. A monitoring platform displays what it is given. It was built to display. It was never asked to reconcile what it displays against what a contract requires, what a standard demands, or what physics permits. Nobody built that layer. The gap is not a failure of any product. It is a category that does not yet exist in most fleets.

Some of the defects in the original assessment were in our own platform. We wrote those up the same way and shipped the report with them in it, and they appear in these references marked as ours. I mention it because it is the only credential in this series that is hard to fake. Any vendor can show you a findings list from someone else's product. A vendor who has never found a fault in their own has not looked.

These references carry no regulatory citations, and that is deliberate; the reasoning is in the Disclaimer. They are also not marketing material dressed as content. I write them in the same voice I use in a control room or in front of a Regional Entity audit team: direct, grounded in evidence, and honest about what a measurement can and cannot tell you.

Machine drafting of switching orders is arriving whether or not the industry is ready, and the argument about it is being held at the wrong altitude. The question is not whether a machine can write a competent order. It can. The question is which fields it is permitted to write, and there is exactly one it must never touch.

Energy Compliance, Inc. exists because most of the consulting offered to registered entities is structured for billable hours rather than for outcomes. Every engagement is overseen by a senior practitioner. We do not bring five people to a meeting that needs one. We automate the work that should be automated and apply senior judgment to the work that requires it. If that matches what you are looking for, the back of this reference tells you how to reach us.

If not, the reference still belongs to you. Take what is useful and apply it. And keep asking the only question that matters about a number on a screen: what would have to be true for this to be wrong, and would I be able to tell?

Rob Smith

Founder and Principal, Energy Compliance, Inc.

Bloomington, Minnesota

How to Use This Reference

This reference is one of eight in the Energy Compliance Monitoring Truth and Fleet Reconciliation Series. The series is organized layer by layer. Each reference stands on its own and assumes none of the others, at the cost of a little repeated framing where two layers meet.

EC-WP-1100 · The Measurement Layer: Missing Is Not Zero
The six-class failure taxonomy and the five design rules that answer it. The foundational reference for the series; every paper that follows takes one class apart on a live system.

EC-WP-1101 · The Alarm Layer: The Board That Deleted Its Own Alarms
Alarm management treated as an information system: load, priority, latching, and the monitor that stops. Seven failure modes, each with the arithmetic that exposes it.

EC-WP-1102 · The Operator Layer: What the Incoming Operator Cannot See
Situational awareness as a testable property of the record rather than a quality of a dashboard. Turnover, attribution, and identity, measured five ways.

EC-WP-1103 · The Rules Layer: A Rules Engine That Measures Its Own Usefulness
The not-needed rate, and how a rules engine earns the right to keep a rule. The measurement that breaks the one-way growth of an alarm catalogue.

EC-WP-1104 · The Procedure Layer: A Machine May Draft It (this reference)
Switching order integrity when a machine prepares the draft. Five integrity requirements, each with the failure that motivates it, and the one field a drafter must never write.

EC-WP-1105 · The Contract Layer: Six Ways to Compute 99%
Availability constructs reconciled against executed agreements. Computability as a per-site position (ready, partial, blocked, or undefined) with the blocker named.

EC-WP-1106 · The Evidence Layer: Displayed Is Not Citable
Provenance as the property that separates a number you can show from a number you can cite. The tautology trap, the seeded schedule, and the obligation nobody could mark done.

EC-WP-1107 · The Platform: The Reconciliation Layer
RenewOps itself: what the platform does, what it deliberately refuses to produce, and what it does not claim. The capability reference for the layer the other seven diagnose the need for.

How to read a figure in this series

Read every number with its source, and with its measurement date where the text gives one. This series does not make claims about systems it has not measured. Where a value could not be verified, it is marked as deferred with the specific question that would resolve it, rather than softened into an estimate.

What these tests cannot tell you

Every diagnostic in this series tests the *record*, not the plant. A clean result means the platform's representation of the fleet is internally consistent, sourced, and current. It does not mean the equipment is healthy, the revenue is correct, or the program is compliant. The reverse also holds: a failed test is evidence that a number cannot be

defended, not evidence that the underlying asset is in trouble. Several of the worst findings in the original assessment sat on top of plants that were running perfectly well. Conflating the two is the most common misreading of this work, and it produces both false alarm and false comfort.

Specific to this reference: the integrity requirements here govern how an order is prepared, reviewed, and recorded. They are silent on whether a given switching sequence is electrically correct for a given plant. No document can carry that. It belongs to the person whose name goes on the approval line, and the whole point of the requirements is to make certain there is still such a person.

The findings in this reference are drawn from an assessment of one fleet. They are offered as a taxonomy to test your own fleet against, not as a statement about it.

Executive Summary

A machine may draft it. Only a person may sign it. Switching order drafting is one of the strongest applications for machine assistance in a control room, and one of the easiest to get catastrophically wrong. The line between the two is not model quality. It is which fields the machine is permitted to write.

A switching order is a sequence of state changes on energized equipment, executed by a person, verified at each step, and relied on by whoever works on the isolated plant afterward. In my view it is the most consequential document a control room produces. It is also highly structured, highly repetitive, and heavily dependent on a one-line diagram. That is exactly the profile machine drafting suits.

> **The paradox**
>
> The value of a drafted order is that it is complete and consistent. The danger of a drafted order is that it is complete and consistent, which is also what a correct one looks like.

#	Integrity requirement	Because
1	**Every step carries an expected state**	A step without a verifiable outcome cannot be checked, only performed. See Ch. 3.
2	**The drafter never writes an identity field**	Prepared-by, approved-by and executed-by are assertions about people. See Ch. 5.
3	**A refusal is a refusal, never a document**	A model's decision not to draft must not be saved as the order. See Ch. 6.
4	**Equipment absent from the one-line is still equipment**	A drafter that only knows the diagram omits what the diagram omits. See Ch. 4.
5	**The lifecycle is enforced, not decorative**	A register in which nothing ever reaches "executed" has not shown that it works. See Ch. 6.

Chapters 1 and 2 define the order and the drafter's boundary. Chapters 3 through 6 take the five requirements with the failures that motivate them. Chapter 7 covers the registers that must exist alongside. Chapter 8 is ten questions for your own records. Chapter 9 states what our own platform enforces: as of August 2026, none of the five requirements was yet an enforced control in RenewOps. One related control was.

1 What a Switching Order Actually Is

A sequence of state changes, each verifiable, executed under a named authority, producing a defined end state that other people's safety depends on.

1.1 The Four Properties

Property	Means	Fails as
Ordered	Steps execute in sequence. Order is load-bearing: opening in the wrong sequence can be the difference between isolation and an arc.	Reordering without re-approval
Verifiable	Each step states the state expected afterward, so the operator confirms rather than assumes.	Null expected state (Ch. 3)
Attributed	Prepared by, approved by, executed by. Three roles, three people, recorded.	Caller-supplied identity (Ch. 5)
Terminal	It produces a defined end state that a clearance can be issued against.	No isolation register (Ch. 7)

1.2 The Three Roles, and Why They Are Three

Preparation, approval and execution are separate acts by separate people, and the separation is the control. The person who wrote the sequence is not the person who checks it, and neither of them is the person standing at the switchgear.

Machine drafting changes exactly one of those three. It changes *who prepares*. It does not, and must not, change who approves or who executes. Any design in which machine drafting collapses two of the three roles has removed a control, whatever it has added.

What machine drafting is good at

Completeness against a device list. Consistent step phrasing. It does not skip the verification step or forget the restoration sequence because it is tired. It produces the same quality of order at 03:00 as at 14:00. These strengths are real, and they line up with the failure modes of tired humans.

What it is not good at

Knowing what is not on the diagram. Knowing what changed on site last week. Knowing that the labeling at this particular substation departs from the convention everywhere else. Chapter 4 takes that up.

2 Drafting, and Where the Machine Must Stop

The drafter's job is to produce a complete, ordered, fully specified candidate sequence and hand it to a human with every assumption visible.

2.1 What the Drafter Receives

A drafting request is not a prompt. It is a structured object, and the structure is what makes the output checkable.

A well-formed request carries:

- the site, as a canonical identifier rather than free text;
- the objective;
- the target equipment, as identifiers drawn from the asset register;
- the specific revision of the one-line diagram the draft is to be prepared against;
- a reference to the entity's own approved switching practice; and
- the identity of the requester, resolved from an authenticated session and never supplied as text.

Two of those fields do more work than they appear to. **The one-line revision** makes the draft reproducible and makes it obvious when a draft was prepared against a superseded diagram. **The operating practice reference** ties the draft to the document the order must conform to, so the approver can check conformance rather than plausibility.

2.2 What the Drafter Produces

Every step must be complete before the order can be submitted, and completeness is specific. Each step needs an ordered position, a device identified from the asset register rather than invented, the action, **the expected state after the action**, the method by which the operator confirms that expected state, and the provenance of the step. Where the step was inferred rather than read, it also needs the assumption behind it, written out. A step missing any of those is not a terse step. It is an incomplete one, and the register should refuse it.

Labels are not function. The asset-population finding in EC-WP-1105 applies here: **naming conventions are not reliable across sites.** A drafter inferring function from a tag pattern will be right at most sites and wrong where the convention differs. Function comes from the asset register, never from the shape of a name.

Provenance is the discipline of EC-WP-1100 applied to a switching order. A step derived from the one-line diagram, a step derived from the entity's approved practice, a step added because the equipment class normally requires it, and a step the drafter *inferred* carry four different

levels of evidence. An approver reading an order in which all four look the same is checking a document that has hidden the one thing they most need to see.

> **The design rule**
>
> Every inferred step is flagged and carries its assumption in plain language: "Assumed DC combiner fuses present per equipment class; not shown on one-line revision C." The approver can then confirm or delete it. With the assumption visible, that is a thirty-second decision. Without it, the omission cannot be detected.

2.3 The Four Things the Drafter May Never Write

1. **Any identity field.** Prepared-by is the requesting operator, resolved server-side from the session, not from anything the drafter produces. Approved-by and executed-by are written only by the approval and execution paths.
2. **Any approval state.** A draft is created in draft and can reach no other state by drafting.
3. **An order number that implies issuance.** Numbering is a sequence controlled by the register and allocated at submission, not at drafting. Otherwise abandoned drafts burn numbers out of a sequence that later has to be explained.
4. **Its own refusal, as a document.** See Chapter 6.

3 Expected State: The Field That Makes a Step Verifiable

On the register we measured on August 24, 2026, five drafts carried complete step arrays targeting real devices. **Expected state was null on every step of every one.** That register is ours (see Chapter 9).

3.1 What That Means Operationally

A step reading *"Open the feeder breaker"* is an instruction. A step reading *"Open the feeder breaker. Expected state: OPEN. Verify by position indication and voltage absent on the load side"* is a control.

The difference is not documentation quality. The second step can fail. An operator who performs the first step and finds an ambiguous indication has no defined test of whether the step succeeded. An operator performing the second either gets the expected state or stops.

Without expected state	With expected state
The step was performed.	The step was performed *and confirmed*.
Failure is detected at the next step that depends on it, or not at all.	Failure is detected at the step that caused it.
The completed order evidences intent.	The completed order evidences outcome.
Post-event reconstruction relies on memory.	Post-event reconstruction relies on the record.

The last row is where this becomes a compliance question as well as a safety one. An executed switching order with expected states and confirmations is a contemporaneous record of what the plant was doing at each moment of the sequence. Without them it records only what somebody meant to do.

3.2 Refuse Incomplete Steps at Submission

The query below returns every order that should not be submittable. It is written for PostgreSQL against an illustrative schema (switching_order, switching_step, asset_register); rename the tables and columns to match yours.

Calculation 1 · Step completeness

```
-- Every row returned is an order that must not be submittable.
SELECT o.order_id, o.site_id, COUNT(s.order_id) AS steps,
COUNT(*) FILTER (WHERE s.order_id IS NOT NULL AND s.expected_state IS NULL) AS missing_expected,
COUNT(*) FILTER (WHERE s.order_id IS NOT NULL AND s.verification_method IS NULL) AS missing_verification,
COUNT(*) FILTER (WHERE s.order_id IS NOT NULL AND NOT EXISTS (SELECT 1 FROM asset_register a WHERE a.device_id = s.device_id)) AS unknown_device,
COUNT(*) FILTER (WHERE s.source = 'inferred' AND s.assumption IS NULL) AS unexplained_inference
FROM switching_order o LEFT JOIN switching_step s ON s.order_id = o.order_id
GROUP BY o.order_id, o.site_id
HAVING COUNT(s.order_id) = 0
OR COUNT(*) FILTER (WHERE s.order_id IS NOT NULL AND (s.expected_state IS NULL OR s.verification_method IS NULL OR NOT EXISTS (SELECT 1 FROM asset_register a WHERE a.device_id = s.device_id) OR (s.source = 'inferred' AND s.assumption IS NULL))) > 0;
```

Two details in that query are deliberate. It uses NOT EXISTS rather than NOT IN for the unknown-device test, because in PostgreSQL a NOT IN test with no matching value returns null rather than true when the list contains a null, and a single null device identifier in the asset register would silently hide every unknown device. And the left join returns orders with no steps at all, which an inner join would never show you.

Enforce the test as a check at the submission boundary, not as a report somebody reads later. A report describes orders that already exist, and by then the incomplete one may have been executed.

4 The Equipment the Diagram Does Not Show

A drafter working from a one-line diagram produces an order complete with respect to the diagram. Plants are not complete with respect to their diagrams.

4.1 The Gap

One-line diagrams are abstractions built for a purpose, such as protection coordination or an interconnection study, and they legitimately omit equipment that does not matter to that purpose but matters a great deal to isolating a circuit safely. Direct-current combiner fusing on a photovoltaic site is the standard example. In my experience it is often absent from the one-line and often exactly what must be addressed to make a section safe to work on.

A drafter that knows only the diagram will produce an order that reads as complete and leaves it out. That is more dangerous than an obviously incomplete order: the reviewer's attention goes to what is present rather than to what is missing.

> **The requirement this creates**
>
> The drafter must draw on two sources, not one: **the site-specific model** (one-line, asset register, tag list) and **a class-standard equipment model** that knows what equipment normally accompanies a given configuration, whether or not the diagram shows it. Steps from the second source are flagged standard_equipment with their assumption stated, so the approver confirms presence rather than discovering absence.

4.2 The Three-Source Model

Source	Supplies	Confidence
Site model	Devices, connectivity, labels, revision.	High, but only as current as its last update.
Approved practice	Required sequence patterns, mandatory verification steps, tagging conventions.	High. It is the governing document.
Class standard	Equipment normally present for this configuration and not shown.	Assumption. Always flagged.

5 Identity: The One Thing Content Guards Cannot Guard

A set of privileged functions guarded their content well. One demanded a directive reference before writing a schedule of record. One refused an evaluation interval under five minutes. One refused a priority change with no written reason. Not one of them could guard identity.

5.1 The Finding

Across the platform assessed in 2026, the fields naming who did something (set_by, approved_by, changed_by, responded_by) were **strings supplied by the caller**.

The functions were careful about everything except the one thing that makes an approval an approval. A function that refuses to write a setpoint without a directive reference, and then accepts approved_by = 'anyone' from whoever called it, has secured the content of the record and left its authorship open.

It got worse. Twenty of those functions ran with elevated privilege, and twelve were called by nothing in the application at all, including one that allocates switching order numbers.

> **The point**
>
> Content guards test whether the record is well-formed. Only the identity path can test whether it is true.

5.2 The Rule

An identity field is never a parameter.

It is resolved server-side from an authenticated session, checked against the roster, and written by the platform. A function that accepts an actor name as an argument cannot produce an attributable record, however carefully it validates everything else.

For a switching order this is not a hardening preference. **Approval is the control.** An order approved by a name the system accepted on the caller's word has not been approved. It has been annotated.

5.3 Find the Functions That Accept an Identity

In PostgreSQL, "elevated" means a function declared SECURITY DEFINER, which runs with the privileges of the user that owns it rather than the user that calls it. PostgreSQL also grants EXECUTE on every new function to PUBLIC by default, so, unless default privileges have been changed, a privileged function is callable by every role that can reach its schema until that grant

is revoked. The query below lists the functions in the public schema whose arguments look like an actor.

Calculation 2 · Privileged functions that accept an identity

```
-- Any function taking an actor-shaped argument is an attribution risk.
-- 'anon' is Supabase's role for unauthenticated access; on a plain PostgREST stack, use the role named in db-anon-role. Substitute the role your platform uses for unauthenticated callers.
SELECT p.proname, pg_get_function_arguments(p.oid) AS arguments,
p.prosecdef AS runs_elevated,
has_function_privilege('anon', p.oid, 'EXECUTE') AS callable_anonymously
FROM pg_proc p JOIN pg_namespace n ON n.oid = p.pronamespace
WHERE n.nspname = 'public'
AND pg_get_function_arguments(p.oid) ~* '(approved_by|set_by|changed_by|responded_by|executed_by|operator|actor|user_id)'
ORDER BY p.prosecdef DESC, callable_anonymously DESC;
```

Read the result in three passes. Functions that are *both* elevated and callable anonymously are the immediate exposure. Functions accepting an identity argument at all are the design problem. Functions in that list that nothing in the application calls should have their EXECUTE grants revoked rather than be left available.

5.4 The Three-Role Separation, Enforced

Role	Written by	Must be refused if
Prepared by	The drafting path, from the authenticated session.	The session identity is not on the roster.
Approved by	The approval path only.	Same identity as prepared-by; or the order is not in submitted; or any step fails the completeness check.
Executed by	The execution path only, step by step.	Same identity as prepared-by or approved-by; or the order is not approved; or steps are confirmed out of sequence.

The self-approval refusal in the second row is the cheapest control in this document and, in my experience, the one most often absent. A preparer approving their own order defeats the separation entirely, and the test is a single equality check.

6 The Lifecycle, and What an Empty Register Tells You

Measured on August 24, 2026, one switching order register held twenty rows: thirteen cancelled, seven draft, and **none in submitted, approved or executed.** This is the RenewOps register, and these findings are ours.

6.1 What the Register Contained

Observation, as of August 24, 2026	Count
Total records	20
Cancelled	13
Draft	7
Currently submitted, approved or executed	0
Drafts titled test / Test / Sample / TEST	6 of 7
Drafts with complete step arrays targeting *real* devices	5
Steps carrying an expected state	0
Records whose body was a model's refusal to draft	1

Read the fifth and sixth rows together, because that combination is the interesting one. Six of seven drafts were titled as tests, and five of them carried complete step sequences addressing real switchgear at real sites. A test-titled record containing real device operations sits in an ambiguous position a register should never permit. It is not clearly an exercise and not clearly operational, and nobody reading the table later can tell which it was.

6.2 The Refusal Saved as a Document

One cancelled record contained, as its order body, **a language model's refusal to produce the order.**

The model behaved correctly. It declined. The defect is entirely in the calling code, which treated a refusal as a successful response and wrote it into the record where an order belongs.

This is the same shape as every failure in this series: a failure that looked like a result, here a declined generation read as a document.

The rule

A drafting call has three outcomes: *an order*, *a refusal*, or *an error*. They are three different types with three different destinations. A refusal goes back to the requester with its reason and creates no record. Only a validated order is saved, and validation means passing the completeness check in Chapter 3.

6.3 Lifecycle Health

The first query shows the distribution of current states. The second asks whether anything has completed.

Calculation 3 · Lifecycle health

```
SELECT state, COUNT(*) AS n,
ROUND(100.0 * COUNT(*) / SUM(COUNT(*)) OVER (), 1) AS pct,
MIN(created_at) AS oldest, MAX(created_at) AS newest
FROM switching_order GROUP BY state ORDER BY n DESC;
-- Has anything completed? Reads current state only; "ever" requires the state-change history.
SELECT COUNT(*) FILTER (WHERE o.state = 'executed') AS executed_now,
COUNT(*) FILTER (WHERE o.title ~* '(test|sample|demo)' AND EXISTS (SELECT 1 FROM
switching_step s JOIN asset_register a ON a.device_id = s.device_id WHERE s.order_id = o.order_id))
AS test_titled_real_devices
FROM switching_order o;
```

6.4 What an Empty Register Means

A register in which nothing reaches a terminal state is telling you one of three things, and it is worth finding out which:

1. **The capability is not in use.** Switching happens on paper; the register is decorative.
2. **The lifecycle cannot be completed.** A transition is blocked or unimplemented, so orders pile up in draft.
3. **It is genuinely new** and has not yet carried real work.

All three are legitimate answers. Only the third is safe to leave alone. The first is the most dangerous: a visible but unused safety register creates an impression of control the organization does not have. Somebody asking for switching records will find it, and the answer will be that the real records are elsewhere.

The judgment from the assessment stands unchanged: **build it properly or remove it from the navigation. A safety register that reads empty is worse than an absent one.** There is no acceptable middle position, and the middle position is where these features tend to sit for years.

7 The Registers That Must Exist Alongside

A switching order produces an isolated state. Something has to record who is relying on that state, and on the system we measured, nothing did. These findings are also ours.

7.1 What Was Missing

Register	State as of August 24, 2026
Isolation / lockout control	No register existed.
Site access	Zero rows, and no columns for arrival, departure, status, escort, badge or clearance. It could not have answered "who is inside the fence" even if populated.
Work orders	No backing table found under eighteen probed names.

The second row is the instructive one. An empty table can be filled. A table *whose schema cannot express the question* cannot. No amount of data entry makes a site access register without arrival and departure columns capable of telling you who is on site.

That distinction is worth carrying generally: **before asking whether a register is populated, ask whether its schema can answer the question it exists for.** The second failure is invisible in every completeness metric, because a table with no rows and a table with no relevant columns look identical in a row count.

7.2 The Chain That Has to Close

Link	Record	What it holds
1	**Switching order**	A defined isolated state, produced step by step and verified.
2	**Isolation record**	Binds that state to a physical control, with who applied it and when.
3	**Clearance**	Issued to a named person against that isolation, with scope and duration.
4	**Site access**	Who is on site, under which clearance, when they arrived, and whether they have left.
5	**Restoration**	Clearance surrendered, isolation removed, switching order reversed, end state verified.

Every link is a record. A chain with a missing link cannot answer the question the whole chain exists for: *is it safe to re-energize?* On the assessed system, no isolation register existed, no clearance register was examined, and the site access register could not record who was on site, so an executed switching order would have ended in a state nothing downstream could reference.

7.3 Identity, One More Time

The switching order identifier was **one of six incompatible namespaces addressing the same sites**. The other five were an internal site key, a forecast slug, a document code, a free-text name and the upstream platform's object identifier.

For a safety register this is not a reporting inconvenience. It means a switching order and the alarms for the same substation may not be joinable, so the question *what was alarming while this order was being executed?* cannot be answered by query. That is precisely the question an event investigation asks.

8 Ten Questions for Your Own Switching Records

#	Question	What a bad answer means
1	Does every step carry an expected state and a verification method?	Without them the order evidences intent, not outcome.
2	Can any function write an approver's name from a caller-supplied string?	Then approval is annotation, and the control does not exist.
3	Can the preparer approve their own order?	A single equality test closes this. Without it the separation is gone.
4	Has anything in your register ever reached executed?	Work out which of the three explanations in Chapter 6 applies before leaving it in the navigation.
5	Do any test-titled records contain real device operations?	Ambiguous records in a safety register cannot be reconstructed later.
6	What happens when a drafter declines to produce an order?	If the refusal is saved as the body, refusals are being read as results.
7	Does the drafter know about equipment absent from the one-line?	It will produce an order complete with respect to the diagram and incomplete with respect to the plant.
8	Is every inferred step flagged with its assumption?	Unflagged inference cannot be told apart from documented fact.
9	Does an executed order tie to an isolation record and a clearance?	Otherwise the chain ends in a state nothing references.
10	Can your site access register answer "who is inside the fence"?	Check the schema, not the row count. Missing columns cannot be filled in.

The principle

A machine may draft it. A person prepares it, a different person approves it, and a third executes it step by step. The machine's contribution is completeness. Everything that makes it an order rather than a document is human, and must be written by the platform, not by the caller.

9 What RenewOps Does About This

RenewOps is the reconciliation layer. It reads your monitoring stack, checks it against ratings, agreements and the documents of record, and declines to publish what it cannot defend.

This chapter is shorter than its counterparts in the rest of the series, and the reason is the finding. As measured in August 2026, none of the five integrity requirements in this reference was yet an enforced control in RenewOps. One related control was enforced: controlled document provenance. A second, order numbering from a controlled sequence, was built, but when measured no order was in a submitted state and the register held no evidence that allocation had run. The five requirements are what RenewOps is being built to. A document arguing for provenance discipline does not get to blur that line.

9.1 The Controlled Procedure Library

As measured on August 26, 2026, RenewOps held 121 controlled documents (operating practices, alarm response guides, switching practices), live and retrievable, filed by document type, with post-load reconciliation reporting nothing missing. Three behaviors in that library matter more than the count.

A company procedure is never returned as the standard it implements. The document type says what the document *is*. The link to a standard family is carried in a separate field. Every field is supplied explicitly on ingest rather than auto-detected, because auto-detection, in an earlier version of the register, misfiled eight documents as standards. An operator who searches for the standard and receives somebody's internal procedure has been handed the wrong authority, and will not necessarily notice.

Supersession runs on ingest. A replacement retires its predecessor. When the library was reloaded, the entire prior load retired. There is no state in which two versions of a controlled document are both live because nobody remembered to withdraw one.

Ingest re-verifies the operator against a roster on every call and writes an audit row per ingest. An unrostered code was rejected rather than accepted and reconciled later.

> **The finding raised against the library itself**
>
> 99 of the 121 controlled documents still carried a *placeholder* document number, and the approver and wet-signature register could not be loaded. Both were raised ahead of the library's effective date rather than after it. A controlled document library whose documents are not properly numbered is a filing system with aspirations.

9.2 Order Numbers Come From a Controlled Sequence

An order number is **allocated by the register at submission**. A drafter does not type it, and drafting does not mint it. As measured on August 24, 2026, no order was in a submitted state (Table 6.1), so the register held no evidence that allocation had run, and the control is reported as specified, not enforced. Once exercised, it prevents the Chapter 2 failure of abandoned drafts burning numbers out of a sequence somebody later has to explain.

9.3 What the Register Actually Contained

The register counts are in Table 6.1; the one row that table combines splits by state as follows, because 13 cancelled plus 7 draft accounts for all 20 records:

State, as of August 24, 2026	Count
Currently submitted	0
Currently approved	0
Currently executed	0

9.4 Where the Requirements Stand

Control	Status	Evidence (August 24 and 26, 2026)
Order numbering from a controlled sequence (related control)	Specified (built; no allocation evidenced)	Register allocates at submission; when measured, no order was in a submitted state (Table 6.1), so the register held no evidence that allocation had run.
Controlled document provenance: a procedure is not a standard (related control)	**Enforced**	Explicit typing on ingest; supersession; roster check; audit row.
Every step carries an expected state (Requirement 1)	Specified	Null on every step on record. Not yet enforced at write.
The drafter never writes an identity field (Requirement 2)	Specified	When measured, no order was in submitted, approved or executed state, so the register held no evidence that the separation had run.
A refusal is a refusal, never a document (Requirement 3)	Specified	One cancelled record holds a refusal as its body.
Equipment absent from the one-line is still equipment (Requirement 4)	Specified	No order on record has step provenance or an inferred-step assumption populated.
The lifecycle is enforced, not decorative (Requirement 5)	Specified	Nothing in the register was in submitted, approved or executed.

> **Design, not control**
>
> An integrity control with no evidence that it has run is a design, not a control.

9.5 Why the Honest Version Is the Useful One

A capability claim in this domain can be checked, and somebody with the authority to look will check it. If a reference claimed enforced separation of drafter, approver and executor, the first competent question at a demonstration would be: *show me an order approved by somebody other than the person who drafted it.* On the register as measured on August 24, 2026, there was no such order to show, because there was no approved order at all.

The requirements in Chapters 3 through 6 are correct requirements. They are the ones this platform is being built to, and the ones we would test any other platform against. What they are not, yet, is a description of controls in force. The distinction between a specification and a control is precisely what this reference exists to teach.

9.6 Adjacent Registers, and a Standing Judgment

As measured on August 24, 2026, the site access register held no rows and lacked the fields to answer who was inside the fence, no lockout and tagout register existed, and no work order backing table was found under eighteen probed names (Section 7.1); the standing judgment in Section 6.4 applies to all three.

Register figures measured August 24, 2026; library figures August 26, 2026, on a live production instance. Every status statement above is dated because a register measured today may read differently tomorrow, and the direction of travel on this one is the point.

Glossary of Terms

Term	Meaning in this paper
Approval path	The part of a platform that records an approval, writing the approver's identity from the authenticated session. Used in this paper to distinguish it from the drafting and execution paths.
Approved practice	The operating entity's own approved switching practice: the document that governs sequence, mandatory verification steps and tagging conventions, and that an order must conform to.
Asset register	The platform's authoritative list of devices, with their identifiers and attributes. In this paper, the only valid source of a device in a switching step.
Class-standard equipment model	A model of the equipment types that normally accompany a given configuration, whether or not a site's diagram shows them. Steps drawn from it are flagged standard_equipment.
Clearance	In this paper, the record issued to a named person against an isolation, with scope and duration, that permits work on the isolated equipment.
Drafter	The machine component that prepares a candidate switching order. It prepares; it does not approve or execute.
Execution path	The part of a platform that records execution, step by step, writing the executor's identity from the authenticated session and refusing steps confirmed out of sequence.
Expected state	The state a device should be in after a step is performed (for example, OPEN), stated on the step so the operator confirms the outcome rather than assumes it.
Inferred step	A step the drafter added from reasoning rather than reading it from the one-line diagram, the approved practice, or the equipment class. It must carry its assumption in plain language.
Isolation record	The record binding an isolated state to a physical control (for example, a lock or tag), with who applied it and when.
Lifecycle	The sequence of states a switching order passes through in a register: draft, submitted, approved, executed, or cancelled. Enforced when the platform refuses transitions that skip a state or a check.
One-line diagram	A single-line electrical diagram of a plant or substation. Built for a purpose such as protection coordination or an interconnection study, it can omit equipment that matters for isolation.
Provenance	The recorded origin of a value or a step, which tells the reader how much evidence stands behind it. The subject of EC-WP-1100 and EC-WP-1106.

Term	Meaning in this paper
Register	A table or system of record that holds a class of safety or operational records: switching orders, isolations, site access, work orders.
Refusal	A language model's decision not to produce the requested output. In this paper, one of three distinct outcomes of a drafting call, alongside an order and an error.
SECURITY DEFINER	A PostgreSQL function option under which the function runs with the privileges of the user that owns it rather than the user that calls it. "Elevated" in this paper means a function with this option.
Site model	The site-specific sources a drafter reads: the one-line diagram at a stated revision, the asset register and the tag list. High confidence, but only as current as the last update.
Switching order	A sequence of state changes on energized equipment, each verifiable, executed under a named authority, producing a defined end state that other people's safety depends on.
Three-role separation	Preparation, approval and execution recorded as separate acts by separate people. The platform refuses an approval by the preparer and an execution of an unapproved order.
Verification method	The way the operator confirms a step's expected state, such as position indication or voltage absent on the load side.

How Energy Compliance Can Help

If machine drafting is on your roadmap, or already in your control room, the questions in Chapter 8 are where to start. We can help you answer them on your own records.

Gap and risk assessment of your switching and safety records. We test your switching order register, isolation and site access records against the requirements in this reference: expected state on every step, identity written by the platform, refusals kept out of the record, a lifecycle that completes. You get each gap with its evidence.

Audit preparation. An executed switching order with expected states and confirmations is a record an auditor can follow. We help you build the evidence trail around it so the record you show is the record you kept.

Monitoring Truth Assessment. The full defect battery against your live monitoring stack. Every finding returns with a source, a timestamp, and the question that resolves it, whether or not you buy anything afterward. Nobody buys a reconciliation layer before they know they need one, and the only honest way to find out is to look.

Evidence Provenance Review. Your compliance-relevant signals traced to source and sorted into citable and not, with the seeded, the derived, and the self-referential separated out from the issued.

Fleet Exposure Read. Your availability constructs reconciled against your executed agreements, returning a per-site computability position (ready, partial, blocked, or undefined) with the specific blocker named for each site that cannot compute.

WattOps Compliance is our NERC compliance evidence platform (wattopscompliance.com). We will run one standard at one facility through WattOps at no cost. CIP standards are excluded from that offer.

RenewOps is our renewable operations monitoring and analytics platform, with a Data Center module. It is the reconciliation layer this series describes, and Chapter 9 states plainly where it stood against these requirements in August 2026.

Tell us which of the ten questions you could not answer cleanly, and we will start there.

Website: energycomplianceinc.com · Email: rsmith@complianceforenergy.com · Phone: 763.438.4427

About the Author

Rob Smith is the Founder and Principal of Energy Compliance, Inc., with more than thirty years in the industry on every side of the system. He began in control-room operations as a Reliability Coordinator, Transmission Operator, and Power System Operator across RTO/ISO and utility control centers. On the regulator side, he was a senior compliance auditor and subject matter expert for NERC Reliability Standards. He audited grid facilities, evaluated mitigation adequacy, participated in risk-based oversight of utility mitigation activities, and supported the development of violation notifications and settlements in FERC-directed enforcement actions. He conducted a regulatory audit in the Sultanate of Oman against its Sector Law and Grid Code.

Education: MBA, Energy Management, University of Mary; MSL, Compliance Law, Fordham Law University; BAS, Energy Management, Bismarck State College; BAS, Mechanical Engineering, Metallurgy minor, University of Florida; BAS, Criminology, St. Petersburg College (State Law Enforcement Certification); AAS, Power Plant Technology, Bismarck State College; AAS, Electrical Transmission System Technology, Bismarck State College.

Certifications: NERC Lead Auditor training; previously NERC-certified at the Reliability Coordinator level; 1st Class A Boiler Engineer License, Minnesota.

Auditing teaches one habit that never leaves: before you believe a number, ask what it would look like if it were wrong.

The views expressed in this publication do not represent the views of NERC, FERC, or any Regional Entity.

About Energy Compliance, Inc.

Energy Compliance, Inc. is an independent regulatory compliance and advisory firm serving the energy sector, with a focus on NERC, FERC, and RTO/ISO compliance and a particular concentration in the ERCOT and Texas PUCT markets. The firm helps registered entities and prospective registrants through the full compliance lifecycle: registration, program design, evidence development, RSAW production, mitigation, and audit defense.

The approach is research-analyst-first. Every conclusion is tied to an authoritative source, every narrative is backed by evidence, and every deliverable is built to survive regulator scrutiny. Every engagement is overseen by a senior practitioner with regulator-side experience. We do not staff for billable hours. We staff for outcomes. Where automation can replace manual work, we build the automation. Where senior judgment is required, the senior is in the room.

That discipline extends to EC's compliance technology: WattOps Compliance, our NERC compliance evidence platform, and RenewOps, the reconciliation layer described throughout this series.

Energy Compliance is not affiliated with, sponsored by, or endorsed by the North American Electric Reliability Corporation, the Federal Energy Regulatory Commission, or any Regional Entity.

Website: energycomplianceinc.com · Email: rsmith@complianceforenergy.com · Phone: 763.438.4427

Sources and Authorities

First-party measurements

- Energy Compliance, Inc., measurements of the RenewOps switching order register and adjacent registers (August 24, 2026) and controlled procedure library (August 26, 2026) on a live production instance, as reported in the August 2026 edition. Not re-measured for this edition.
- Energy Compliance, Inc., 2026 assessment of a live production monitoring stack on a multi-gigawatt utility-scale PV and storage fleet (anonymized), the source of the Chapter 5 findings and the series taxonomy.

PostgreSQL 18, Supabase and PostgREST documentation (accessed October 17, 2026)

- PostgreSQL Global Development Group, *System Catalogs: pg_proc* (prosecdef). https://www.postgresql.org/docs/18/catalog-pg-proc.html
- PostgreSQL Global Development Group, *System Information Functions and Operators*. https://www.postgresql.org/docs/18/functions-info.html
- PostgreSQL Global Development Group, *CREATE FUNCTION* (SECURITY DEFINER). https://www.postgresql.org/docs/18/sql-createfunction.html
- PostgreSQL Global Development Group, *Privileges*, Section 5.8 (default EXECUTE grant to PUBLIC). https://www.postgresql.org/docs/18/ddl-priv.html
- PostgreSQL Global Development Group, *Row and Array Comparisons*, Section 9.25.2. https://www.postgresql.org/docs/18/functions-comparisons.html
- Supabase, *Postgres Roles* (anon and authenticated). https://supabase.com/docs/guides/database/postgres/roles
- PostgREST, *Configuration*, db-anon-role. https://docs.postgrest.org/en/v12/_sources/references/configuration.rst.txt

Companion references in this series

- Energy Compliance, Inc., EC-WP-1100, *Missing Is Not Zero*. https://energycomplianceinc.com/whitepapers/ec-wp-1100-missing-is-not-zero.html
- Energy Compliance, Inc., EC-WP-1105, *Six Ways to Compute 99%*. https://energycomplianceinc.com/whitepapers/ec-wp-1105-six-ways-to-compute-99.html

© Energy Compliance, Inc. · EC-WP-1104 · Status as of October 17, 2026