ENERGY COMPLIANCE, INC. | *Field Reference for Compliance Advisory*

ADVISORY SERIES · EC-WP-701

Why Compliance Programs Get Complicated

How layered decisions, documentation drift, and consultant dependency turn lean programs into unmanageable ones.

Compliance programs do not start complicated. They become complicated, one reasonable decision at a time, until no single decision can be pointed to as the cause.

Energy Compliance, Inc. Professional Reference

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, Regional Entity directive, 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. Compliance with NERC Reliability Standards depends on facts and circumstances unique to each registered entity, and the reader should consult qualified counsel and subject matter experts when making compliance decisions.

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.

Energy Compliance, Inc. assumes no responsibility or liability for errors, omissions, or outcomes resulting from the use of this publication.

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 worked 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.

The entities that do reliability well share a common habit. They take the standards seriously without confusing them with reliability itself. They know that a NERC Reliability Standard is a floor, not a ceiling. They know that compliance is something an auditor evaluates, while reliability is something a system either delivers or does not. They prepare for audits by building programs that survive real questions, not binders that look thick.

That is the perspective these references try to share. Each one takes a single topic, whether a standard family, an operational function, a regulatory framework, or an emerging industry challenge, and walks through how it actually works.

They are written for the compliance manager who wants to understand the system, not just memorize requirements. For the legal counsel who has to brief a board honestly. For the senior operator who has been told that compliance and reliability are the same

thing and suspects they are not. For the new compliance hire who was handed a binder and told good luck.

Energy Compliance exists because much of the consulting offered to registered entities today 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 we apply senior judgment to the work that requires it. If that approach is what you are looking for in a compliance partner, the back of this reference has our contact information.

If not, the reference still belongs to you. Take what is useful. Apply it well. And remember the only test that ultimately matters: when the system needs to perform, does it?

Rob Smith

Founder and Principal, Energy Compliance, Inc.

Executive Summary

Compliance programs do not start complicated. They become complicated. The drift happens slowly, across years, through layered decisions made one at a time by people solving immediate problems. Documentation accumulates. Processes acquire steps that no one can later justify. Consultants leave behind frameworks the entity keeps maintaining after the engagement closes. Ownership moves outside the operators who actually run the work.

By the time the program is unrecognizable, no single decision can be pointed to as the cause. The cause is the sum of every decision. The cure is a different operating discipline, and that discipline is harder than the original program design.

The regulator's own rules explain why this matters. NERC's Sanction Guidelines say that "situations involving multiple serious risk violations or systemic or programmatic failures should typically result in monetary penalties and/or non-monetary penalties."

What follows, in short:

- Complexity is rarely intentional. It accumulates one decision at a time and outlives the people who introduced it.
- A program that grows beyond what its operators can explain has crossed from compliant to fragile.
- In my experience, documentation drift is the costliest accident in compliance, because no one notices it happening.
- Process bloat is a symptom of decisions made for comfort, not for compliance.
- Loss of programmatic ownership is the moment a program stops being defensible from the inside.
- Consultant-driven complexity outlasts the consultant by years.
- Simplification is a strategic discipline. In my view, most programs treat it as housekeeping, and that is a large part of why they grow too complex.

1 Complexity Is Almost Always Accidental

Programs do not become complicated on purpose. They become complicated because no one is responsible for keeping them simple.

1.1 The Clean Original

No compliance program is designed to be unmanageable. The original architecture was usually clean. There was a standard, a set of requirements, an operating reality, and a documentation framework meant to connect them. Whoever built that first framework had the entire program in their head. At the start, the program was smaller than its operators.

1.2 How Accumulation Happens

Then time passed. New standards landed. Audits produced findings. Staff rotated and equipment changed. Each event triggered a local response: a new procedure, a new control, a new evidence artifact in the catalog. None of these decisions was wrong on its own, and none was even unreasonable in context. The problem is structural. Nothing in the standard operating model requires anyone to subtract.

The result is accumulation. Twelve years in, the entity is maintaining controls that no longer map cleanly to any current requirement. It keeps evidence formats created for an audit four cycles old, and review cadences everyone follows because they have always been followed. The program has grown larger than the people who operate it, and no one can point to the moment it crossed that line.

1.3 Make Subtraction Part of the Job

Recognizing the pattern matters because the cure is not another process to manage complexity. The cure is making subtraction part of the operating discipline. Programs that retire as much as they add stay legible. Programs that only add eventually become structures their operators describe rather than understand.

> **From the Field**
>
> Programs become unmanageable because no one is paid to keep them simple. The standard operating model has a structural bias toward addition. Subtraction has to be intentional, or it does not happen. A program that has grown larger than the people who operate it has crossed a line the entity rarely notices in real time.

2 The Layered-Decision Problem

Each individual decision was reasonable in context. The aggregate is unreasonable, and no one is responsible for the aggregate.

2.1 Every Fix Adds a Layer

Layered decisions are the most common engine of compliance complexity. A finding closes with a new procedure. A regulatory change closes with a new control. A leadership concern closes with a new review cycle. Each addition is justified at the moment of the decision, and the justification is often still on file, sometimes in the same document as the procedure itself.

The decisions are rarely revisited. Once the new procedure is in the binder, nobody asks again whether it is still required. The next finding triggers the next procedure; the next regulatory change triggers the next control. After ten years the program holds the residue of every compliance decision the entity ever made, with no built-in way to retire the layers that have outlived their purpose.

2.2 The Accountability Gap

The accountability gap makes it worse. Each individual decision had an owner. The aggregate program does not. No single role inside the entity is responsible for asking whether the program as a whole is still legible, still defensible, and still executable by the people who run it. The program drifts because the architecture is ownerless even when every component is owned.

2.3 The Counter-Discipline

Mature programs build a counter-discipline. They review the program annually as a whole, with explicit authority to retire requirements that have become ceremonial. They treat the architecture as a living artifact that one person owns end to end. They accept that subtraction is a senior judgment call and give it to a senior person. In my view that discipline is uncommon, and its absence is why most programs end up layered rather than coherent.

> **From the Field**
>
> Each layered decision was reasonable. The aggregate is the problem, and the aggregate has no owner. Annual whole-program review with authority to retire ceremonial requirements is the discipline that prevents this. It is not common.

3 Documentation Drift and the Compounding Effect

Documentation that is not actively maintained begins drifting on day one. The drift compounds, and so does the audit exposure.

3.1 Drift Is the Default State

Every procedure begins drifting from operational reality the moment it is signed. Equipment changes, software updates, people rotate. The procedure stays put. Eighteen months later it describes an operation that no longer exists. After three years it can describe an operation that nobody now working at the entity has ever performed.

3.2 From One Finding to a Pattern

The compounding effect is the dangerous part. A single drifted procedure is a finding. A pattern of drifted procedures starts to look like a programmatic failure, which the Sanction Guidelines say should typically draw a penalty. Prior programmatic failures involving the same or similar Reliability Standards and Requirements, if still ongoing within five years of the start date of the new violation, can also aggravate its monetary penalty. The audit team documents the findings. Enforcement staff decide whether enough of them add up to a programmatic failure.

3.3 How Drift Happens, and the Discipline That Stops It

The mechanism is mundane. Procedure updates are work. They get scheduled, then deferred. The deferred update becomes the missed update, which becomes the never-updated procedure. Operations runs on tribal knowledge and the documentation becomes ceremonial. Auditors spot ceremonial documentation quickly, because it does not survive comparison with what the operators say in interviews.

Programs that survive audits keep documentation current as a continuous discipline. Every operational change triggers a documentation review. Each quarter, someone compares the documents to actual practice. Once a year, the SMEs who execute each procedure walk through it. None of this is exotic. It is what separates programs that survive audits from programs that explain afterward why they did not.

> **From the Field**
>
> Drift is not a future problem. It is the default state. One drifted procedure is a finding. A pattern starts to look like a programmatic failure, and enforcement staff decide whether it is one. Documentation maintenance is a continuous operating discipline, not a periodic project. Treating it as a project guarantees it gets deferred.

4 Process Bloat: When Steps Outnumber Reasons

Programs accumulate process steps faster than they accumulate reasons for those steps. The result is execution friction that no one can justify.

Process bloat is layered-decision complexity in operational form: steps that each had a purpose once, and together now take longer than the underlying work.

4.1 An Illustration: The Sixteen-Step Self-Report

Picture a Self-Report procedure that has grown to sixteen steps. Eight are signoffs. Three are review cycles. One is a legal hold. One is a board notification. Each answered a real concern once. Now the procedure takes six weeks to run, so a short-fuse Self-Report either skips steps or arrives late.

Late costs something. Under the Sanction Guidelines, NERC or the Regional Entity may reduce a penalty for a Self-Report made within a reasonably prompt time and before the issue is detected by, or the entity is notified of, a compliance monitoring engagement. The Guidelines also say a Self-Report should go in as soon as practical, typically within three months of discovery. Once an audit notice covering the requirement arrives, that credit is generally off the table until the engagement ends. A procedure slower than the calendar is a liability.

4.2 Workarounds Become the Operation

Bloat also has a cultural effect. Operators stop trying to follow the procedure exactly, because it cannot be followed exactly inside the operational rhythm. The workarounds become the actual operation, and the documented procedure becomes fiction. The audit surfaces both, and the entity faces two issues where one would have been bad enough.

4.3 Cutting Steps Takes Seniority

The cure is periodic review of process steps, with explicit authority to remove the ones whose purpose has expired. The reviewer needs enough seniority not to be overruled by the step's original owner, who almost always argues that someone might someday need it.

> **From the Field**
>
> Every step had a reason once. Bloat is what is left when the reasons expire and the steps stay, and the workarounds operators build around it become the audit finding.

5 The Loss of Programmatic Ownership

Ownership tends to migrate away from the operators over time. Once it leaves, the program becomes harder to defend and easier to misalign.

5.1 What Ownership Means

Programmatic ownership is what lets someone inside the entity speak for the program at audit. The owner does not need to be the only person working on the program. They do need to be the one person whose understanding of it is comprehensive and whose accountability for it is unambiguous.

5.2 How Ownership Migrates Outward

Ownership drifts outward over time. A consultant comes in to handle one workstream and ends up as its institutional memory. The compliance manager rotates, and the new manager inherits a structure they did not build. Documentation once maintained by operators is now maintained by outside advisors who know the program better than the people who run it.

5.3 The Brittleness Shows Up at Audit

Once ownership has left the operating staff, the program carries a brittleness that stays invisible until it is tested. The auditor asks why the program is structured the way it is, and the operator on the call does not know. The consultant who knows is not on the call. The answer comes back two days later, in writing, after consultation, and the auditor draws a conclusion about ownership that the entity will struggle to refute.

5.4 Reclaiming It

Reclaiming ownership is achievable, but it takes deliberate effort. The entity has to pull institutional knowledge back inside, which often means rebuilding documentation in language the operators wrote and can defend. It means scheduling internal program walk-throughs with no outside advisors in the room. And it means accepting a program that is temporarily smaller and rougher in exchange for one that is owned from the inside.

> **From the Field**
>
> Programmatic ownership is what lets an operator answer audit questions without calling an outside advisor. Ownership migrates outward by default. Reclaiming it is the work of deliberate years, not weeks. An owned program survives audits. A program owned by people outside the audit room survives until the auditor asks the wrong person.

6 Consultant-Driven Complexity

Some complexity is added by the entity. Some is added by the consultants the entity hired to make things simpler.

6.1 The Deliverable That Stays Behind

Not all complexity starts inside the entity. In my experience, a meaningful share of it is consultant-introduced, and usually with good intentions. A consultant arrives to fix a specific problem, designs a methodology that solves it, and leaves the methodology behind as a deliverable. The methodology becomes part of the program. Six months later the entity is maintaining a framework more elaborate than the standard ever required, one no current operator can fully explain, from a firm whose engagement closed long ago.

The pattern repeats with every engagement. Each brings its own framework, naming convention, classification scheme, and review cadence, and none is retired when the engagement ends. The entity ends up running a program built from the residue of every consulting engagement it ever commissioned, often with no one responsible for reconciling the layers.

6.2 Why It Is Hard to Remove

Consultant-introduced complexity is harder to remove than the home-grown kind because it carries borrowed credibility. It is documented. It looks professional. It came from a firm with a recognizable name, and removing it feels like throwing away advice that was paid for. That feeling is usually misplaced. If the framework is more elaborate than the standard requires and the entity cannot explain why it exists, it is mass without purpose, and removing it is the right call.

6.3 The Structural Defense

The defense is structural. An entity should adopt no consultant-introduced framework without three things: an owner inside the entity, an explanation of why it exists written in the operators' own language, and a stated retirement criterion. Without those, the framework persists by default. In my view, default-persistent frameworks are the most expensive kind of complexity, because the entity pays for them indefinitely and never revisits the decision.

> **From the Field**
>
> In my experience, a meaningful share of program complexity originates with the consultants hired to reduce it. Consultant-introduced frameworks survive long after the engagement ends because no one is responsible for retiring them. Adopt no external framework without internal ownership, an internal explanation, and an explicit retirement criterion.

7 When Complexity Becomes Risk

Complexity does not stay neutral. At a certain density, it becomes the source of the audit findings the program was supposed to prevent.

7.1 The Threshold

Below a certain threshold, complexity is tolerable. Controls execute, evidence gets generated, and the audit passes. The complexity is overhead, but it does not drive outcomes. Above the threshold the math changes, and complexity starts causing the failures the program was built to prevent.

7.2 How Errors Aggregate

The mechanism is easy to recognize. When the program is too complex for any operator to hold in their head, decisions get made on partial understanding. Controls get executed without a clear grasp of the requirement behind them. Documentation gets updated without checking whether the change carries through to related artifacts. Evidence generated for one purpose gets used for another. Each error is small. Together they produce findings the entity cannot fully explain, because no one fully understood the program in the first place.

7.3 Supportable Complexity Moves

The threshold varies by entity, by function, and by the seniority of the staff. A program that works with twenty experienced operators can become unmanageable when half of them rotate out. A program that passed its last audit because the SMEs carried the institutional memory turns into a finding generator when that memory walks out the door. The complexity did not change. The complexity the staff could support did.

7.4 Reading the Late Signal

The threshold is hard to see in advance and easy to see in hindsight. If the program has produced a string of unexpected findings across several audit cycles, and those findings cluster around documentation drift, inconsistent evidence, and SMEs contradicting each other, it has crossed the line. The answer is not more controls. It is structural simplification, done with senior authority, across the whole program rather than one workstream at a time.

> **From the Field**
>
> Supportable complexity drops sharply when experienced operators rotate. Programs built around institutional memory become fragile with every personnel change. A string of unexpected findings around drift, inconsistency, and contradiction is the late signal. Listen earlier or pay more later.

8 Simplification as a Strategic Discipline

Reducing complexity is not housekeeping. In my experience, it is the highest-return compliance investment most programs never make.

8.1 Why Housekeeping Fails

Most entities treat simplification as housekeeping. The cleanup happens between audits, gets pushed aside when other work arrives, and never gets the senior attention it needs. Everyone agrees in principle that simplification matters, but agreement alone does not change a program unless someone schedules the work and owns it.

8.2 What the Successful Programs Do

Programs that succeed at simplification schedule the work at the senior level, on a calendar, with explicit authority to retire requirements that no longer trace cleanly to a current standard. They measure the program by what it has removed as well as what it has added. Any framework whose purpose no one inside can explain is a candidate for retirement, whoever introduced it. And they accept that the program will be smaller and less impressive on a slide in exchange for being more defensible at audit.

8.3 The Economics and the Deliverable

In my experience the economics favor simplification. A simpler program costs less to operate, less to audit, less to mitigate when something goes wrong, and less to defend if enforcement escalates. None of that shows in a single quarter. Across an audit cycle, all of it does.

The regulator's rules point the same way. The Sanction Guidelines let NERC or the Regional Entity reduce a penalty for an internal compliance program that is generally effective in preventing and detecting violations. A program its own people can explain is far easier to show as effective.

Simplification is not glamorous. What it produces is a smaller binder, a clearer program, an operator who can answer the audit question without help, and a position that holds up under regulatory scrutiny. Programs that skip it tend to become the case study someone else cites.

> **From the Field**
>
> Every retirement should leave a record of why, so the next reviewer does not rebuild what you removed. Programs measured by what they have removed, not just what they have added, stay legible. The rest do not. Budget for simplification the way you budget for audit prep. Work with no line item gets done last, if it gets done at all.

Glossary

Term	Meaning in this paper
Compliance Audit	A systematic, objective review and examination of records and activities to determine whether a Registered Entity meets the Requirements of applicable Reliability Standards (NERC Rules of Procedure, Appendix 2).
Compliance Enforcement Authority (CEA)	NERC or the Regional Entity in their respective roles of monitoring and enforcing compliance with the NERC Reliability Standards (Appendix 2).
Compliance Monitoring and Enforcement Program (CMEP)	The NERC program in Appendix 4C to the Rules of Procedure, or a Regional Entity's Commission-approved program (Appendix 2).
Documentation drift	EC's term for the growing gap between what a procedure says and how the work is actually done, which begins the day the procedure is signed.
Internal compliance program	An entity's own program to prevent and detect violations. The Sanction Guidelines treat its presence and quality as a factor that may reduce a monetary penalty (Appendix 4B).
Mitigation Plan	An action plan developed by the Registered Entity to correct a noncompliance with a Reliability Standard and prevent its re-occurrence (Appendix 2).
Programmatic ownership	EC's term for the condition in which one person inside the entity understands the whole program and is plainly accountable for it, so the entity can explain its own program at audit.
Registered Entity	An owner, operator, or user of the Bulk Power System, or the entity registered as its designee for compliance, that is included in the NERC Compliance Registry (Appendix 2).
Regional Entity	An entity having enforcement authority pursuant to 18 C.F.R. § 39.8 (Appendix 2).
Self-Logging	A process by which Registered Entities found eligible by a Compliance Enforcement Authority, after a formal review of internal controls, record potential noncompliance on a log instead of submitting individual Self-Reports (Appendix 2; Appendix 4C, Section 4.5A).
Self-Report	A report by a Registered Entity stating that it believes it has, or may have, violated a Reliability Standard (Appendix 2). Under Appendix 4C, Section 4.5, it includes a risk assessment, the extent and cause of the noncompliance, and the mitigating actions.
Subject matter expert (SME)	The person who performs or owns a task and can explain it to an auditor. A working term, not a NERC-defined one.

How Energy Compliance Can Help

This reference argues for smaller, owned programs. We work that way too: a senior practitioner overseeing every engagement, and no new framework left behind.

Three services fit these problems:

- **Regulator-perspective program reviews.** A senior, auditor's-eye read of your program, including what can be retired.
- **Compliance program design and improvement.** Rebuilding the program so the people who run it can explain and own it.
- **Mitigation planning and Self-Report development.** A Self-Report path short enough to keep pace with the calendar.

Services Provided

Energy Compliance services include, but are not limited to:

- NERC reliability and compliance advisory support
- Reliability governance and program assessments
- Registration and applicability analysis
- Operational and engineering reliability alignment
- Compliance program design and improvement
- Audit and enforcement support (non-advocacy)
- Mitigation planning and Self-Report development
- Training and executive briefings on reliability frameworks
- Regulator-perspective program reviews

WattOps Compliance

WattOps Compliance is EC's NERC compliance evidence platform (wattopscompliance.com). It keeps your evidence in your hands, not a consultant's files. EC will run one standard at one facility through WattOps at no cost (CIP standards excluded).

RenewOps

RenewOps is EC's renewable operations monitoring and analytics platform, with a Data Center module. For renewable and storage fleets, it puts the operations data behind the compliance program in front of the people who have to explain it.

If your program has grown past what your own people can explain, start with Chapter 8, then call us.

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 the Sultanate's Sector Law and Grid Code.

He founded Energy Compliance, Inc. to deliver senior regulator-side compliance authority without the layered staffing and billable-hour overhead of larger firms.

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.

About Energy Compliance, Inc.

Energy Compliance, Inc. is an independent consulting and advisory firm based in Bloomington, Minnesota, focused exclusively on electric reliability, cybersecurity reliability, and regulatory compliance for organizations connected to the North American Bulk Electric System. Our work supports registered entities, including Generator Owners and Operators, Transmission Owners and Operators, Reliability Coordinators, Balancing Authorities, and Distribution Providers. We work across NERC Reliability Standards, FERC orders, RTO/ISO market participation rules, Regional Entity oversight, and state regulatory frameworks.

We do this work differently than larger consulting firms. Engagements are overseen by a senior practitioner with regulator-side experience. We do not staff for billable hours. We staff for outcomes. Our deliverables are written to be operationally executable and audit-defensible, not to manufacture activity. Where automation can replace manual work, we build the automation. Where senior judgment is required, the senior is in the room.

The practice is organized around four areas: **NERC Compliance** (program support, interpretation, and audit preparation), **Senior Advisory** (direct engagement on complex reliability questions), **Industry Engagement** (standards development and working-group participation), and **Audit Defense** (Notice of Penalty response and settlement posture). Each engagement is scoped to the entity's role, function, and bulk system impact. 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. Learn more at energycomplianceinc.com.

Sources and Authorities

- North American Electric Reliability Corporation, Rules of Procedure, Appendix 2, Definitions Used in the Rules of Procedure (effective June 27, 2024). https://www.nerc.com/globalassets/who-we-are/rules-of-procedure/appendix-2.pdf
- North American Electric Reliability Corporation, Rules of Procedure, Appendix 4B, Sanction Guidelines of the North American Electric Reliability Corporation (effective January 19, 2021), Section 2.1 (Initial Determination of Whether Monetary and/or Non-Monetary Penalties are Necessary) and Section 3.3 (aggravating and mitigating factors, including Repetitive Violations and Compliance History, Presence and Quality of Entity's Internal Compliance Program, and Disclosure of the Violation Through Self-Reporting and Voluntary Mitigating Activities by the Entity). https://www.nerc.com/globalassets/who-we-are/rules-of-procedure/appendix-4b.pdf
- North American Electric Reliability Corporation, Rules of Procedure, Appendix 4C, Compliance Monitoring and Enforcement Program (effective May 19, 2022), Sections 4.1 (Compliance Audits), 4.5 (Self-Reports), 4.5A (Self-Logging), and 6.0 (Mitigation of Noncompliance with Reliability Standards). https://www.nerc.com/globalassets/who-we-are/rules-of-procedure/appendix-4c.pdf

© Energy Compliance, Inc. · EC-WP-701 · Status as of October 25, 2026