



Wondering what is required in federal regulations regarding leak detection on your hazardous liquid pipeline?

What is in the Federal Regulation?



Deadline for Implementation

Applicable pipelines must have “an effective system for detecting leaks” by **October 1, 2024** (or sooner if your line was built within the last 5 years).



Requirements for Design and Implementation of Leak Detection

If you have *computational pipeline monitoring* (a software-based monitoring tool) it must be designed and implemented following guidelines found in **API RP 1130**

Is Your Pipeline Included?

There is a complicated list of types of lines that are covered and exempted.

Essentially, in 2020, **the requirement for leak detection was extended beyond high consequence areas to all regulated, non-gathering hazardous liquid pipelines.**

So what is API RP 1130 and what is in it?

There are many methods that can be used in leak detection – including tools like external patrolling & simple SCADA line balance. API RP 1130 was developed specifically to provide guidelines for software-based leak detection systems that use an algorithmic approach to detect and alert to anomalies. Such methods can enhance the controller’s ability to identify an anomaly, but the added layer of computation warranted some guidelines.

API RP 1130 was developed to assist operators in the selection, implementation, testing, and operation of such a system.

Section 4 – Design Considerations

- A list of capabilities
- Performance metrics

Section 5 – Supporting Infrastructure

- Source Data (Instrumentation, communications, etc.)
- Integration with SCADA, etc.

Section 6 – O&M and Testing

- System Testing (initial and follow-on)
- Change Management
- User Training

Appendices – Background Information

The Flowstate LDS is built to deliver desired performance...

Capabilities

Flowstate LDS offers a suite of tools designed to tackle a variety of leak detection challenges. This includes methods described in API RP 1130 and features listed in its "selection consideration".

Performance Metrics

Each Flowstate LDS solution is built to optimize performance in the key metrics identified by API RP 1130: Sensitivity, Accuracy, Reliability, and Robustness.

Available Leak Detection Methods:

- Volume Balance
- Negative Pressure Wave
- Statistical Analysis
- Deep Learning Model (replaces Real Time Transient Model)

...and enable successful implementation.

System Testing

Digital Leak Simulation

The Flowstate LDS has a built-in ability to simulate a leak by modifying input data. This method is routinely used for **Initial, Commissioning** tests of each segment model that is delivered. It is also available for use in **Periodic Retesting** or **Change-driven testing** as required by API RP 1130.

The LDS also includes a self-test feature that will provide ongoing health monitoring of the system.

Documentation and Training

Detailed Knowledge Base

Detailed user instructions are available on our online Knowledge Base. Support is also always available during business hours.

In-Person or On-Line User Training

Choose your method of training, We offer different levels of in-person training—tailored to your operation. Or, you can select different topics as needed and on your time from a selection of online courses in Flowstate University.

www.flowstatesolutions.ai

Operations

Best in Class Security

The Flowstate LDS is built to meet or exceed all pertinent API Std 1164 and TSA security provisions and is developed under SOC2 compliance.

Change Management

MOC features include things such as:

- Restricted access to system changes, including alarm thresholds
- Change logs for all parameter changes, alarm inhibits, or system modifications

Features Built for Operation, Maintenance and Testing

- ✓ Data Alarming
- ✓ Digital Leak Simulation & Historical Data Rerun
- ✓ Instrumentation List
- ✓ Alarm Report
- ✓ Configuration Change Reports
- ✓ Alarm inhibit logs
- ✓ Segment Summary report
- ✓ Detailed Segment configuration reports (parameter lists)
- ✓ Extensive online user documentation