

SeeQ®



connect

ORLANDO JUNE 1-3, 2026



Operationalizing Process Hazard Analysis Recommendations with Seeq

Ben Cruz, Aditya Birla Chemicals

Aditya Birla Chemicals PSM Facility in Dalton, Georgia



- Epichlorohydrin
- Flammable solvents
- Formalin



About Me & The Manufacturing Technology Engineer Role



Maintain Process Safety Information



Process Safety Improvements



Hazard Assessment and Safety Audit Site Facilitator.

Uncovered Unknown Safety Interlocks Bypassed During PHA

Popup_Valve_WithLimits_48Lock.gif

XV-1400-21
R1400 BPC R-OH FEED VALVE

OPEN

CLOSE

AUTO

MANUAL

CURRENT MODE: **MANUAL**

ALARMS

INTERLOCK (MORE)

BYPASS

WT-1400-13 REACTOR WEIGHT HIHI
TITZ-1400-06 RX TEMP HI
TITZ-1400-06 RX TEMP HIHI
PROC RXEPICONC ALARM
PROC PRORGANICS ALARM
LIT-1900-03 TANK LEVEL HIHI
PIT-1900-02 TANK PRESSURE HIHI
XS-1900-05 RUPTURE DISK ALARM

☐
☐
☐
☐
☐
☐
☐
☐



OPEN LIMIT OFF
CLOSE LIMIT ON
SOLENOID OFF

CLOSED

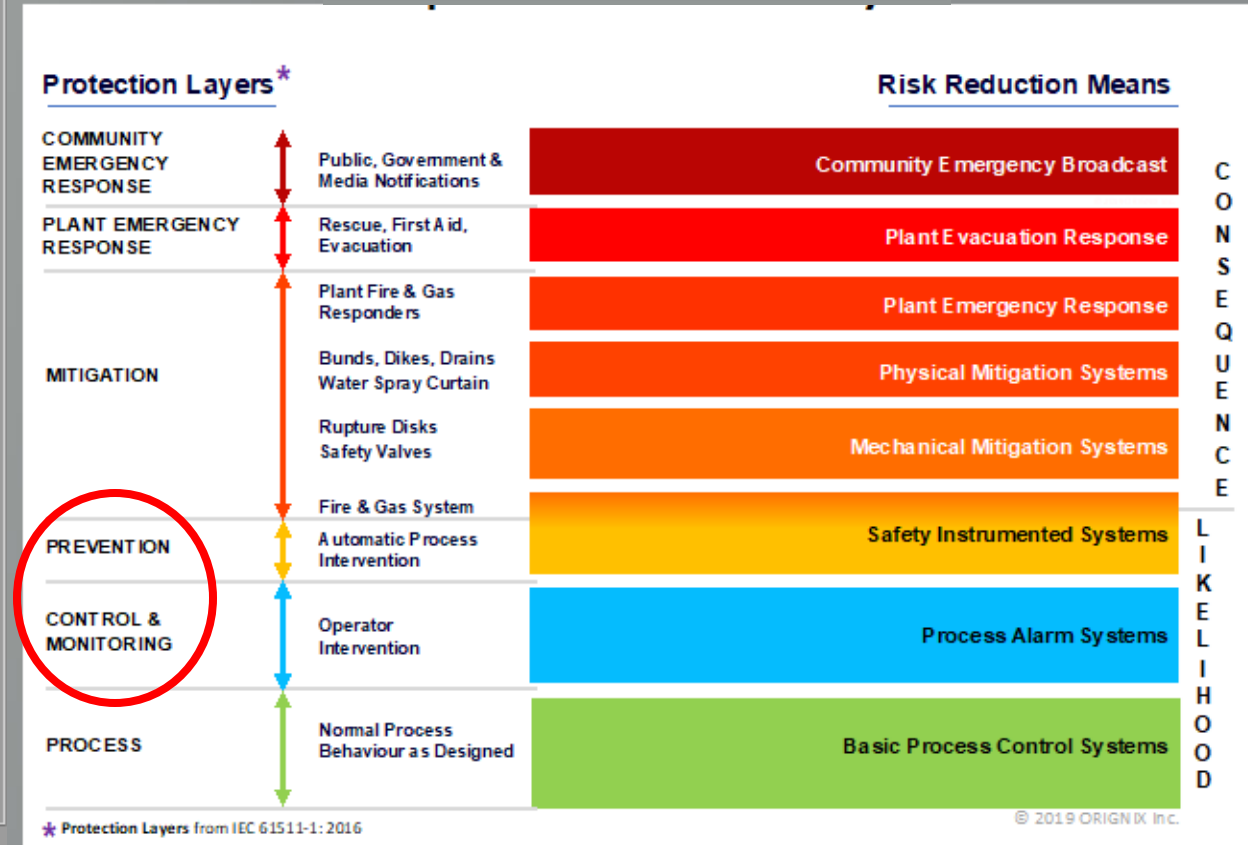


INTERLOCK (HIDE)

BYPASS

PIT-9500-01 N2 PRESS LOLO
PROC EMERGSTOP ALARM
LSH-1400-60 LEVEL HIGH ALARM
PIT-1400-08 REACTOR PRESSURE HIHI
LSH-1910-01 LEVEL HIGH ALARM
XS-1400-11 RUPTURE DISK ALARM
LIT-1400-38 VAPOR REC LEVEL HIHI
PIT-1400-37 VAPOR REC PRESS HIHI
XV-1400-50 RECEIVER OUTLET NOT CLOSED
XV-1400-53 N2 NOT CLOSED
XV-1400-17B EPI FEED NOT CLOSED
XV-1400-19 CATALYST FEED NOT CLOSED
XV-1400-23 R-OH FEED NOT CLOSED
XV-1400-25 PRODUCT DRAIN NOT CLOSED
XS-1400-03 RUPTURE DISK ALARM
R-1400 MANWAY OPEN ACTIVE

☐
☐
☐
☐
☐
☐
☐
☐
☒
☒
☒
☒
☐
☐
☐
☐



Problem: Interlocks Bypassed on Unknown Number of Output Devices



100 + output devices with potential bypasses applied.

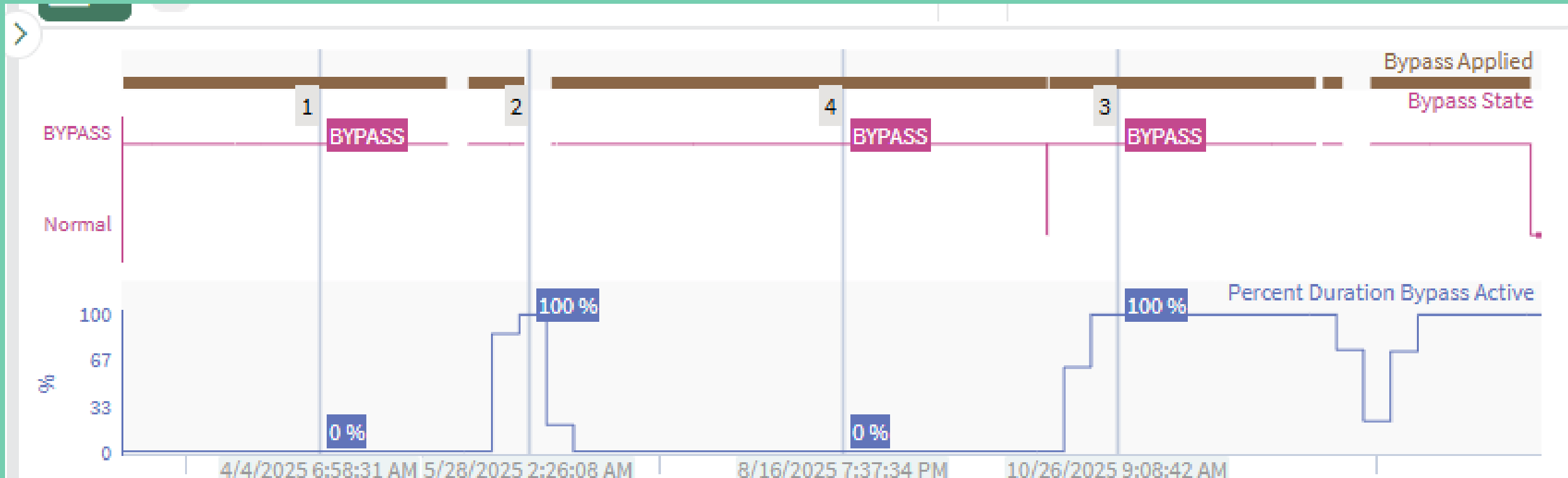


Needed visibility on where and how long bypasses have been applied.

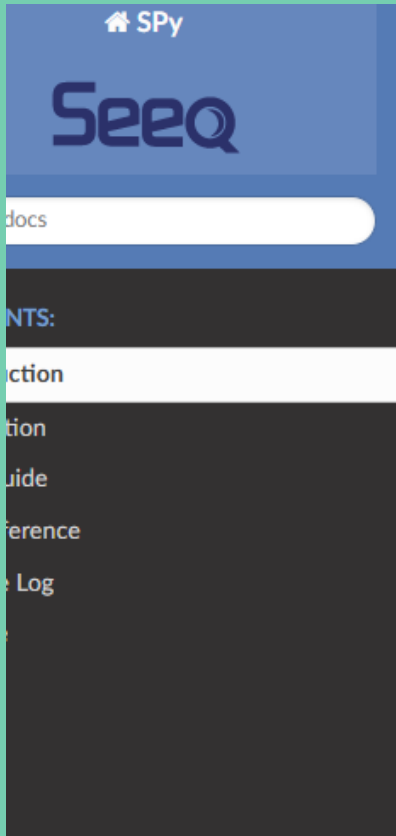


Need dynamic reporting for interlock bypassing on output devices.

Solution (Part 1): Using Seeq Workbench to Determine Methodology



Solution (Part 2): Using Seeq SPy Python Module to Automate Interlock Bypass Analysis



[Home](#) / [Introduction](#)

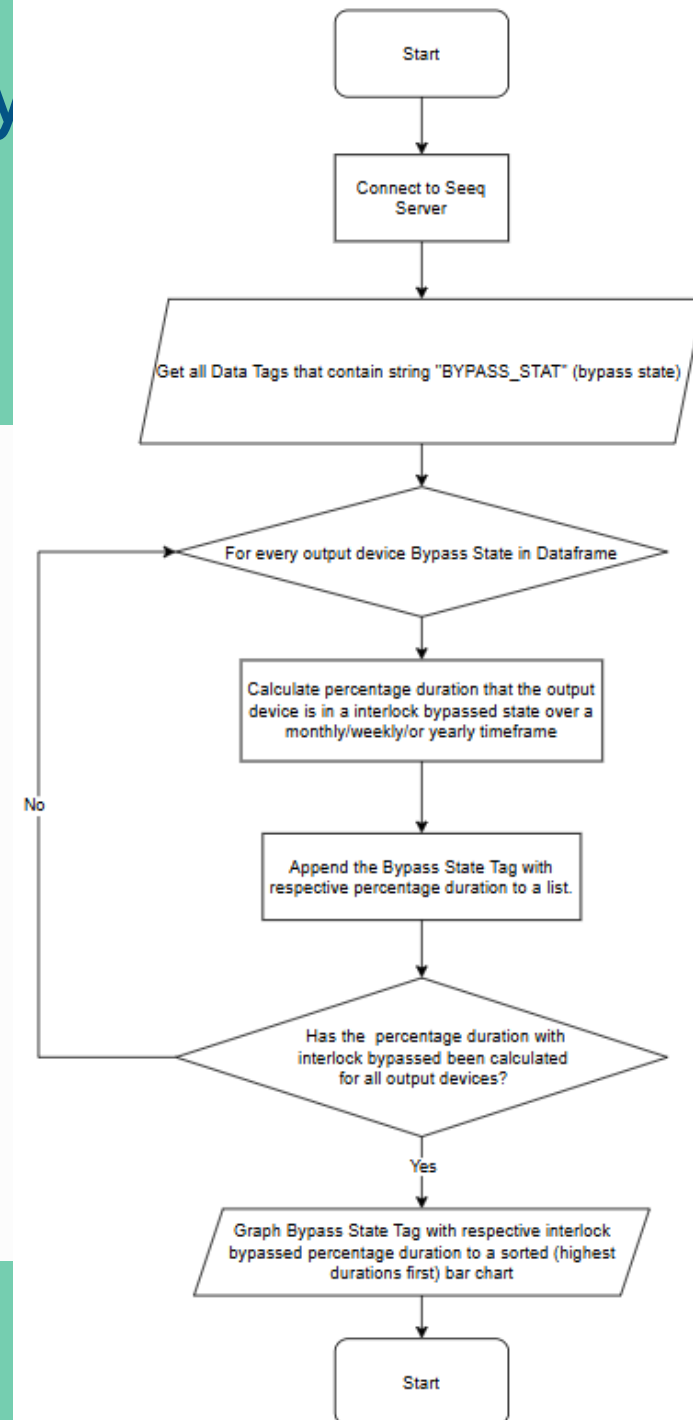
[View page source](#)

Introduction

The Seeq **SPy** module (short for Seeq PYthon) is a friendly set of functions that are optimized for use with [Jupyter](#), [Pandas](#) and [NumPy](#).

The SPy module is the best choice if you're trying to do any of the following:

- Search for signals, conditions, scalars, assets, etc.
- Pull data out of Seeq
- Import data in a programmatic way (when Seeq Workbench's *CSV Import* capability won't cut it)
- Calculate new data in Python and push it into Seeq
- Create an asset model
- Create content using Workbench and Organizer templates
- Manage permissions on many items
- Run automated jobs on a regular schedule



Solution (Pt 3a): Overview of the Simplistic code using Seeq SPy Module

```
1 from seeq import spy
2 import pandas as pd
3
4 # Set the compatibility option so that you maximize the chance that SPy will remain compatible with your notebook
5 spy.options.compatibility = 198
6
7
8 # Log into Seeq Server if you're not using Seeq Data Lab:
9 spy.login(url='https://adityabirlachemicals.seeq.tech/', credentials_file='../credentials.key', force=False)
```

Connect to
Seeq Database

```
1 all_tag_names = spy.search({
2     'Name': 'BYPASS_STAT'
3 })
4
5 all_tag_names
```

Query successful

	Name	Time	Count	Pages	Result
0	BYPASS_STAT	00:00:00.27	192	1	Success

	ID	Name	Description	Type	Value	Unit Of Measure	Datasource Name	Archived
0	0F047B8D-AA92-EC80-AA7F-70BDF008F0CC	DALT:XV-1300-25_BYPASS_STAT	XV-1300-25 BYPASS STATUS	StoredSignal		string	OSDALT03MP	False
1	0F047B8D-AA9C-E8C0-A15A-272D9516026F	DALT:XV-1300-45_BYPASS_STAT	XV-1300-45 BYPASS STATUS	StoredSignal		string	OSDALT03MP	False
2	0F047B8D-AAA6-7500-B3AC-CF045B62DE21	DALT:XV-1300-50_BYPASS_STAT	XV-1300-50 BYPASS STATUS	StoredSignal		string	OSDALT03MP	False

Get bypass
state tags

Solution (Pt 3b): Overview of the Simplistic code using Seeq Spy Module

```
1 from datetime import datetime
2 from dateutil.relativedelta import relativedelta
3 # Current date and time
4 first = datetime.now()
5
6 # Subtract one month
7 last_month_start = first - relativedelta(months=1)
```

Define Periodic Condition
("Monthly")

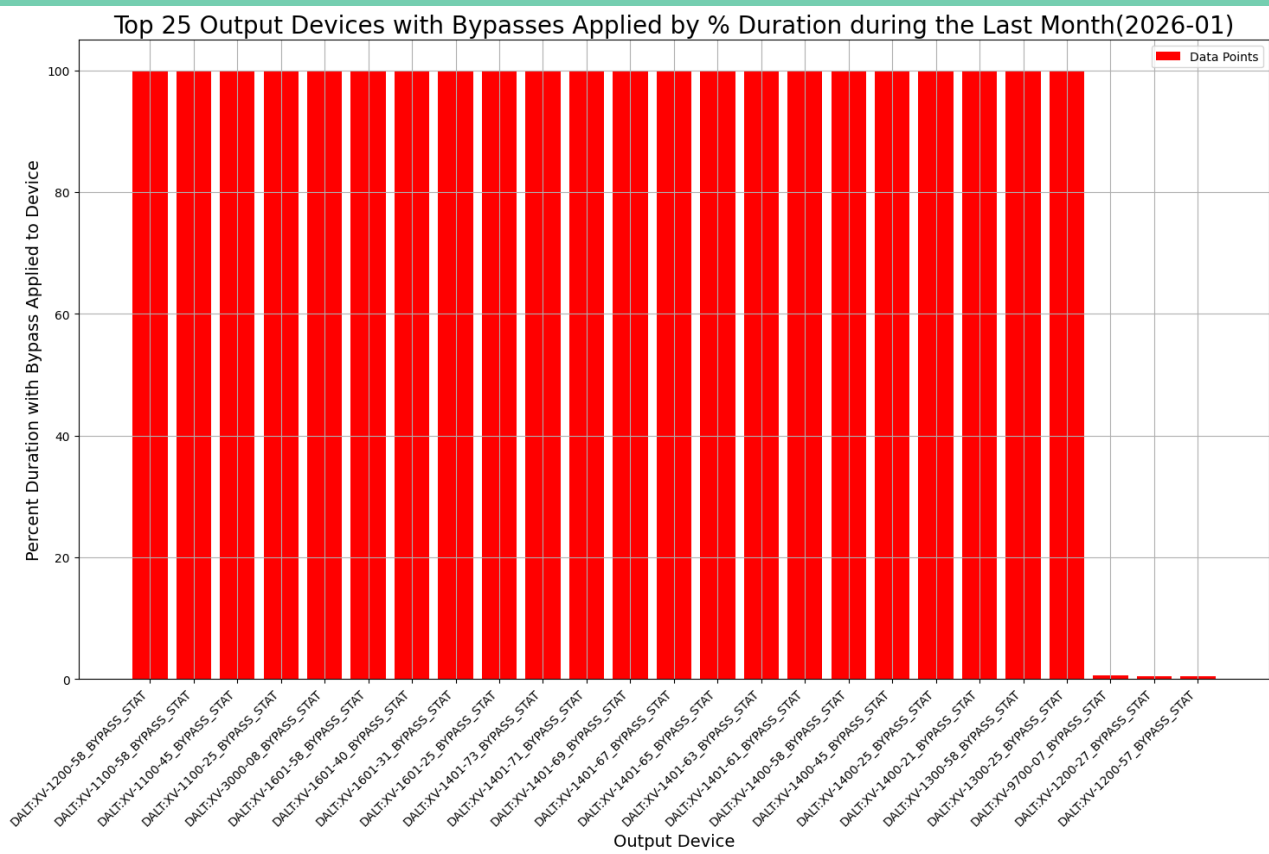
```
1 duration_array = []
2 i = 0
3 for tagname in bypass_stat_full_tags['Name']:
4     mytag = spy.search({'Name':tagname})
5     myresult = spy.pull(mytag,
6                         start = last_month_start.strftime('%Y-%m-%d'),
7                         end   = first.strftime('%Y-%m-%d'),
8                         calculation = '($signal == "BYPASS").setMaximumDuration(30day).aggregate(percentDuration(), months("US/Eastern"),middleKey())',
9                         grid = None)
10    myduration_val = myresult.to_numpy()
11    myduration_tag = bypass_stat_full_tags.iloc[i]['Name']
12    duration_array.append([myduration_val.item(0), myduration_tag])
13    i = i+1
14    print(i)
15 #duration_array
```

For each bypass state, calculate the percentage duration that an output device has an interlock bypassed monthly

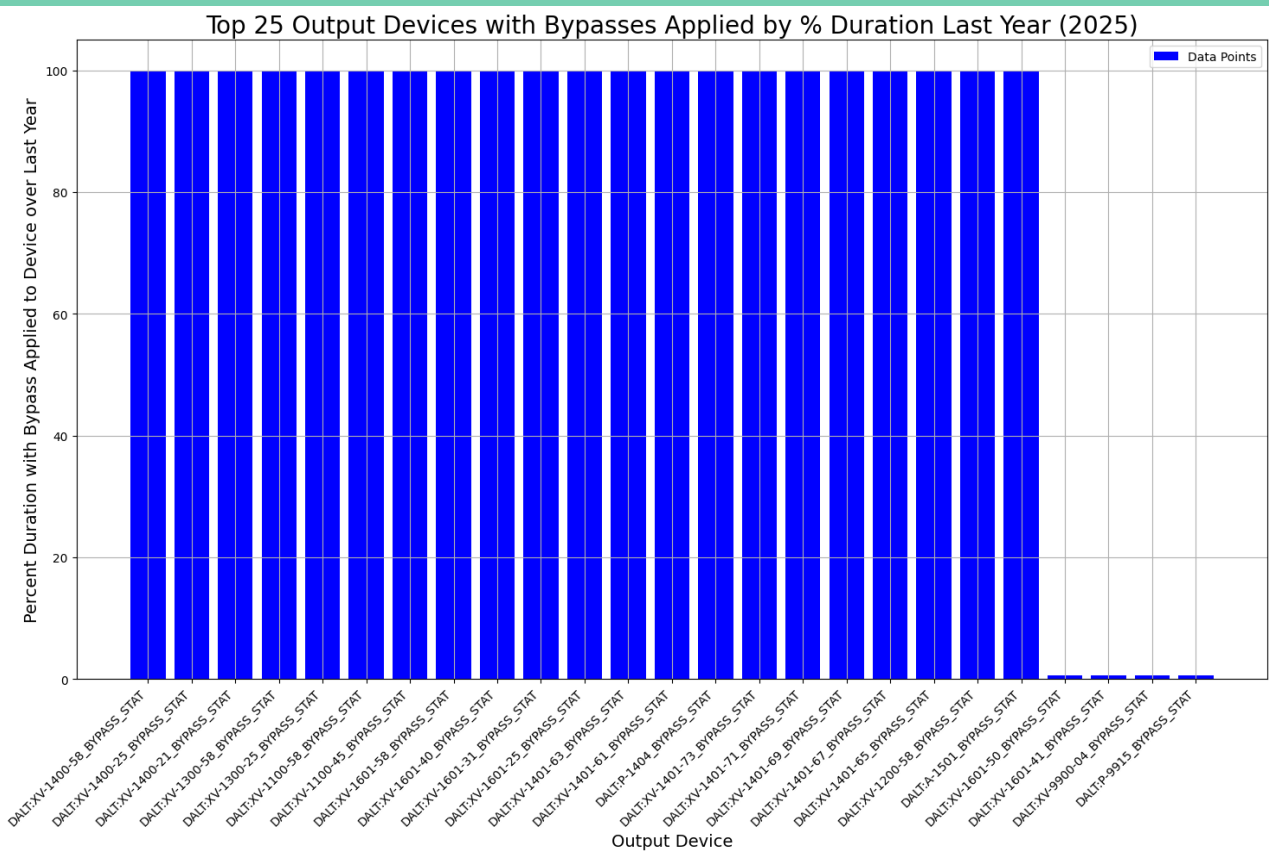
Pull successful from 2026-03-16 00:00:00-04:00 to 2026-04-16 00:00:00-04:00

ID	Type	Name	Time	Count	Pages	Data Processed	Result
0 0F047B8F-7593-EAA0-B06A-11680F95457E	StoredSignal	DALT:XVZ-3000-35_BYPASS_STAT	00:00:00.10	2	1	5 KB	Success

Solution (Pt 4a): Automate the Interlock Bypassed Report Visualization



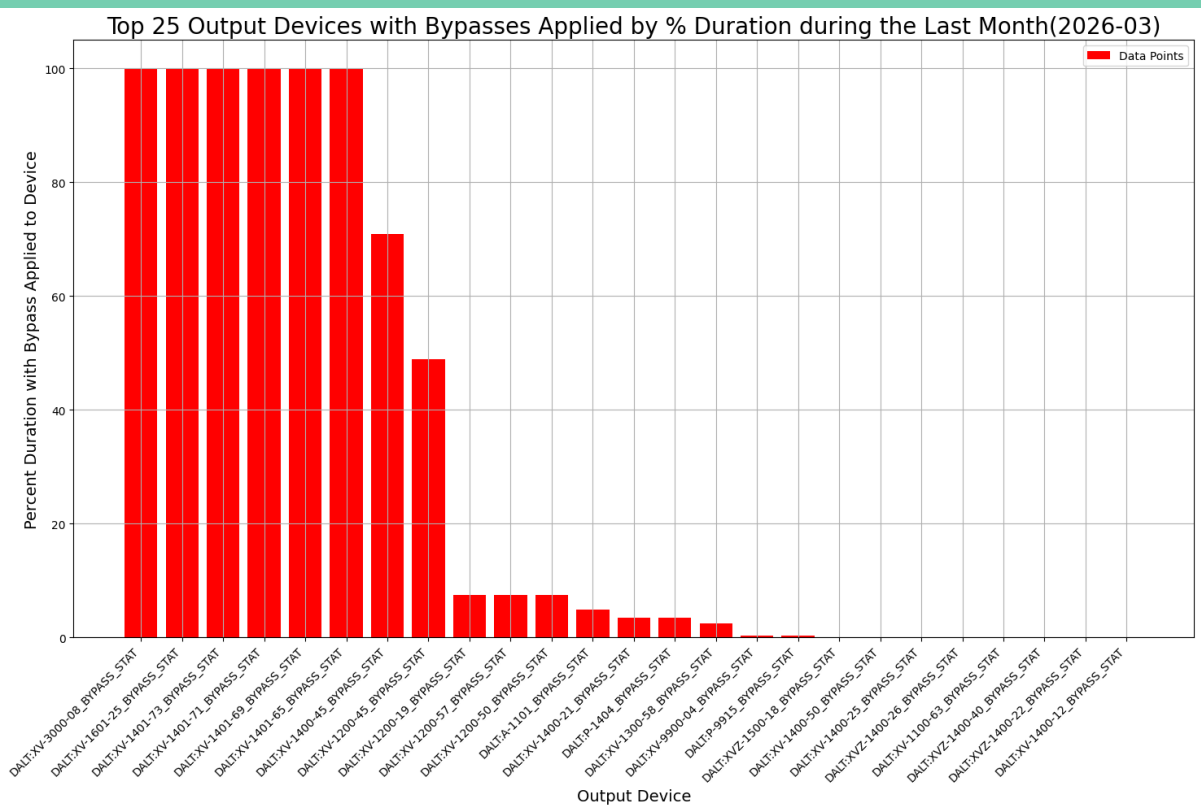
Output Devices with Interlocks Bypassed
January of 2026



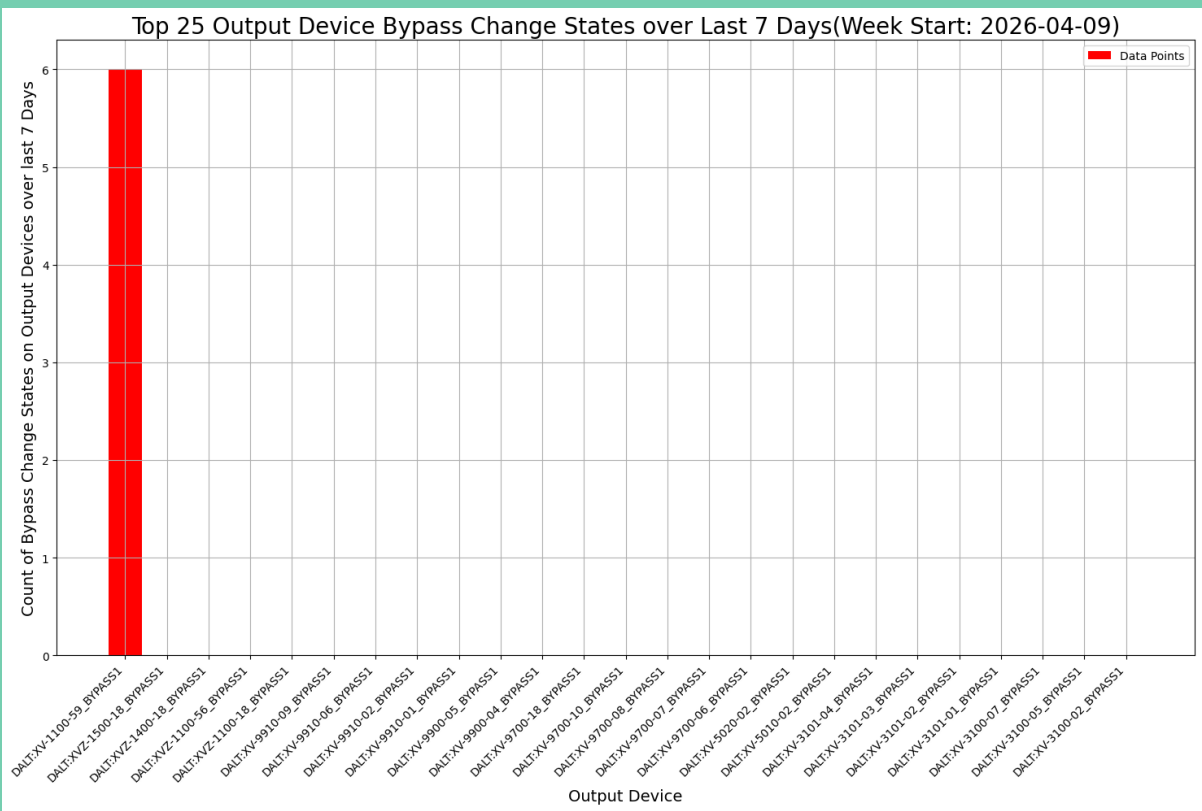
Output Devices with Interlocks Bypassed
All of 2025



Solution (Pt 4b): Automate the Interlock Bypassed Report Visualization

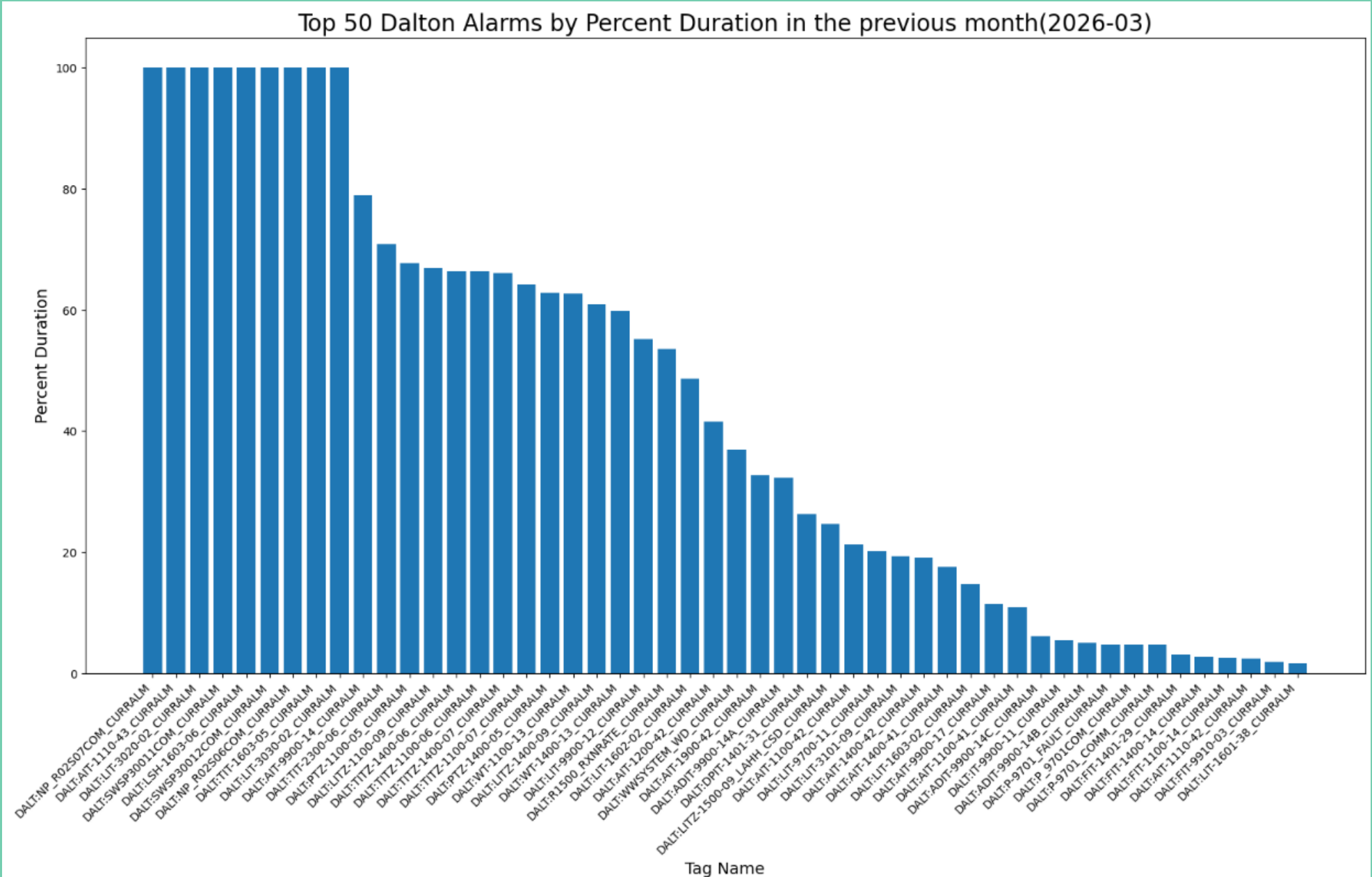


Report Ran on 4/16/2026



Report Ran on 4/16/2026

Bonus Solution: Automate Active Alarm Visualization



Summary

Dynamic reporting using Seeq that can also be automatically scheduled and pushed to an Organizer Report.

Re-usable code that can be utilized in more cases than just bypasses applied (alarm rationalization, equipment faults outs, etc.)

Questions?