LOGZILLA DOCUMENTATION

# Rewrite Rule Walkthrough

Worked example of a LogZilla rewrite rule that drops keepalive noise and tags surviving events with site and device_role for filtering

Data Transforms · Generated June 12, 2026 · logzilla.ai/docs/data-transforms/rewrite-rule-walkthrough

# Rewrite Rule Walkthrough

This use case shows how to reduce event noise and add business context with a focused rewrite rule. The example remains vendor-neutral and applies to common keepalive or heartbeat patterns.

## Scenario

- Devices emit periodic keepalives that clutter dashboards.
- Operations needs those dropped.
- Remaining events need searchable context tags such as `site` and `device_role` for filtering and reporting.

## Goal

- Drop well-defined noise events early in the pipeline.
- Add stable, low-cardinality tags to useful events.

## Prerequisites

- Permission to manage rules and view parser statistics.

## Plan

Define a clear match criterion for noise (for example, "keepalive").
Identify stable tags to add (for example, `site`, `device_role`).
Author a small rewrite rule (YAML recommended; JSON is also supported).
Validate, reload, and verify behavior.

## Procedure

Prepare representative events for testing.

- For ad hoc tests, post two events via the HTTP Event Receiver:

```
# Noise event candidate (should be dropped)
curl -H 'Content-Type: application/json' \
     -H 'Authorization: token YOUR_GENERATED_TOKEN' \
     -X POST -d '{
       "events": [ {
         "host": "lab-router-01",
         "program": "netd",
         "message": "keepalive: OK",
         "user_tags": {"site": "west-dc", "device_role": "edge"}
```

```
      } ] }' \
     'http://lzserver.company.com/incoming'

# Useful event candidate (should be retained and tagged)
curl -H 'Content-Type: application/json' \
     -H 'Authorization: token YOUR_GENERATED_TOKEN' \
     -X POST -d '{
       "events": [ {
         "host": "lab-router-01",
         "program": "netd",
         "message": "interface ge-0/0/1 up",
         "user_tags": {"site": "west-dc", "device_role": "edge"}
       } ] }' \
     'http://lzserver.company.com/incoming'
```

The following YAML rewrite rule example matches the noise pattern and drops those events. The rule ensures stable tags are present on retained events.

```
- rewrite_rules:
  - match:
    - field: message
      op: "=*"
      value:
      - "*keepalive*"
    drop: true
  - match:
    - field: program
      op: eq
      value:
      - "netd"
    tag:
      site: west-dc
      device_role: edge
```

The first rule drops events whose `message` contains "keepalive". The second rule adds `user_tags` when `program` equals `netd`.

## Validate and Load the Rule

```
# Validate rule file
logzilla rules validate /path/to/noise-reduction.yaml

# Add the rule and assign a recognizable name
logzilla rules add /path/to/noise-reduction.yaml --name "Noise Reduction"

# Reload rules so changes take effect
logzilla rules reload
```

## Troubleshooting

- If events are not dropped, verify the match operator and value: use `op: "=*"` with `"*keepalive*"` to match wildcard contains.
- Confirm the event fields being matched (for example, `message`, `program`) by inspecting values:

```
logzilla events values --scope fields --limit 50
```

- Check for rule errors:

```
logzilla rules errors
```

- Ensure rules were reloaded after changes:

```
logzilla rules reload
```

## Notes

- Rewrite rules are best for simple, deterministic actions:
  - Normalize a core field, add a tag, replace controlled text, parse a simple key=value pair, or drop low-value events.
- Escalate to Lua when multiple extractions, conditional logic, external lookups, or message reformatting are required.

## Related reading

- Rewrite rules (https://www.logzilla.ai/docs/data-transforms/rewrite-rules)
- Lua rules (https://www.logzilla.ai/docs/data-transforms/lua-rules)
- Testing and verification (https://www.logzilla.ai/docs/data-transforms/testing-and-verification)
- Command Line Tools —- Data Commands (https://www.logzilla.ai/docs/command-line-tools/data-commands)