



Monitoring the great value in your Big Data

HANDS-ON WALKTHROUGH · FIELD QUALITY

Field quality monitoring

Two feeds, two paths. A CIM firewall feed checked against Network_Traffic, and a raw JSON business feed whose data dictionary the AI Advisors draft, tighten and review. From an empty tenant to 89 monitored fields.

Captured live on TrackMe 2.4.18 · tenant field-quality



The events keep arriving — the fields inside them quietly break

Data source monitoring sees volume and latency. Detections, dashboards and business reports depend on something else: the values inside each event.

A field stops parsing

A TA or parser change leaves the field empty.
Searches keep running — and silently match less.

A value drifts from the standard

teardown instead of allowed, EURO instead of EUR.
The field exists, the value is wrong, CIM-based content misses it.

A field goes missing on a subset

One region, one host, one message type stops carrying the field. Averages hide it; per-field quality shows it.

What Field Quality Monitoring does

FQM samples the events of a feed, checks every field against a data dictionary (presence, regular expression, optional type), and turns the result into TrackMe entities: one per field plus one @global per context. Each entity has a threshold, a state, a score, history and alerting — exactly like a data source in DSM.

Works for CIM data models and for any raw or custom format — only the data context and its dictionary change.

One collect job, one monitor job, one data dictionary

Fields quality — collect, then monitor

One collect job, two phases: sample and score your events, then turn the results into TrackMe entities and alerts.

PHASE 1 · COLLECT

scheduled search — sampling or head mode

1

Data context + dictionary

CIM data model or raw events

2

Score every field

| trackme:fieldsquality · missing/empty/regex

3

Index the JSON samples

→ sourcetype trackme:fields_quality

PHASE 2 · MONITOR

runs 10 minutes after the collect job

4

Parse the collected events

one source per collect job

Discover the entities

@global + one per field (break-by)

5

Thresholds and alerting

compliance % · stateful alerts · email

JSON samples
+10 min

The data dictionary is generated from a CIM data model or reused from a previous job — dictionaries can be shared between collect jobs.

The same approach applies to CIM and to any raw events: only the data context and its dictionary change.

Thresholds are set automatically on the compliance percentage, so alerting reuses TrackMe's stateful alerts and rich email notifications.

1

Data dictionary

Per field: regular expression, optional type, allow unknown, allow missing / empty. Generated from CIM or drafted by AI, shared between jobs.

2

Collect job — score every field

A scheduled search samples the events and runs trackme:fieldsquality: missing, empty, pattern and type checks, per event.

3

Samples indexed

Results land in the summary index as trackme:fields_quality JSON samples.

4

Monitor job — build the entities

Runs ~10 minutes after the collect job: one entity per field, plus @global per context.

5

Thresholds and alerting

Success % against the threshold drives the state; stateful alerts and emails come for free.

The workflow diagram is built into the wizard — expand "Fields Quality Monitoring Workflow Diagram" at the top of every job wizard.

What we are going to do — two feeds, two dictionary strategies

EXAMPLE 1 • CIM

Cisco ASA firewall logs

```
index=siem* sourcetype="cisco:asa"
```

- Checked against Network_Traffic / All_Traffic
- Dictionary generated from the CIM data model
- Simulate, inspect a regex failure, tune, re-simulate
- Sampling with Auto define, create, run
- 76 entities over 4 regional indexes

EXAMPLE 2 • NON-CIM (JSON)

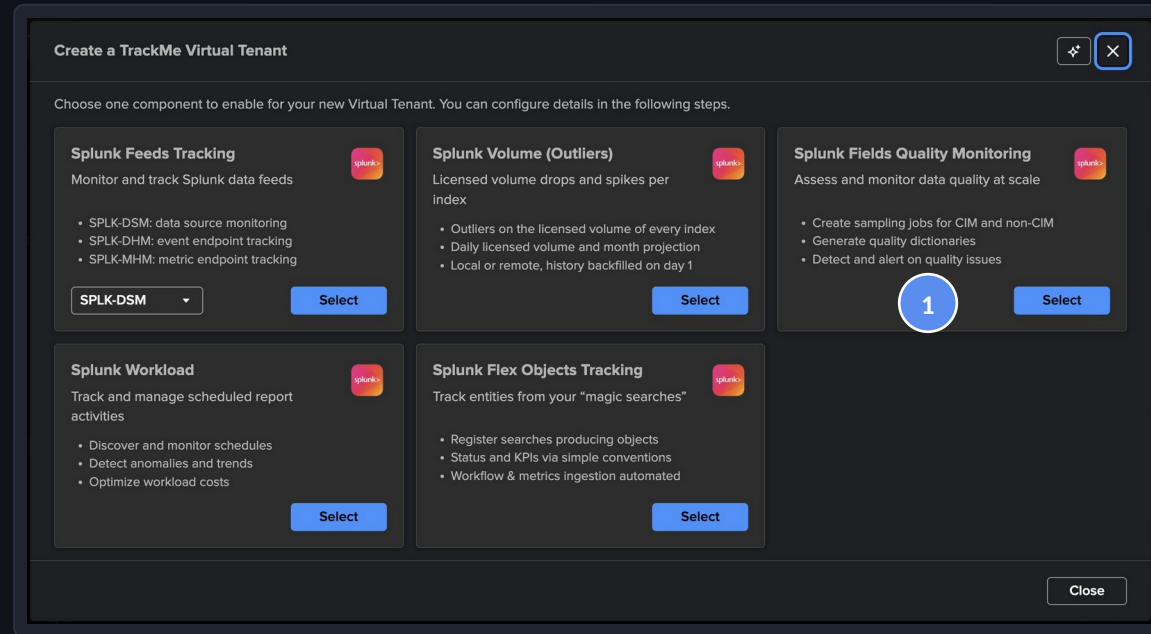
E-commerce order transactions

```
index="business_data"  
sourcetype="business:orders:transactions"
```

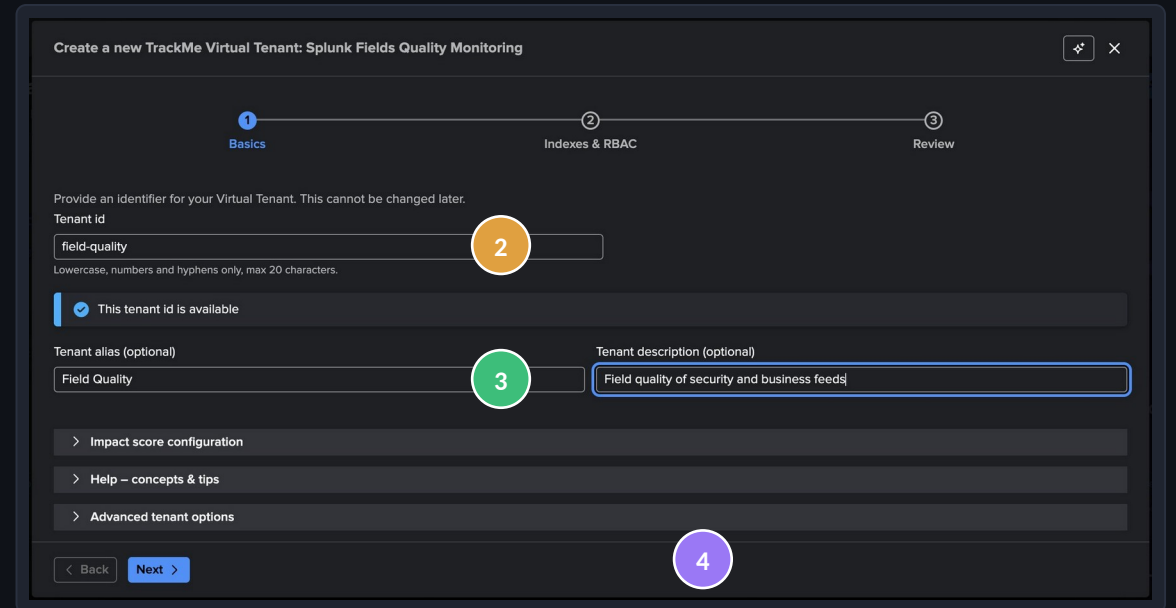
- Custom pseudo data model, raw JSON payload
- Dictionary drafted by the AI dictionary advisor
- Per-field AI advisor tightens currency
- FQM Advisor reviews the failing field
- 13 entities, 28% of orders with a bad currency

Both run in one Virtual Tenant created for the occasion. The last slides show a mature tenant with several months of history.

A dedicated tenant — pick Splunk Fields Quality Monitoring



- 1 Splunk Fields Quality Monitoring**
Sampling jobs for CIM and non-CIM data, quality dictionaries, alerts on quality issues.



- 2 Tenant id — field-quality**
Immutable identifier, checked for availability as you type.
- 3 Alias and description**
What people see on the Virtual Tenants card.
- 4 Next: indexes & RBAC**
Summary, audit, metric and notable indexes, owner and roles — defaults are fine.

The tenant is ready — every FQM action starts from here

The screenshot shows the TrackMe - Tenant Home dashboard. The top navigation bar includes links like /Virtual Tenants, Topology Studio, Configuration, Artificial Intelligence, Maintenance, Search, API & tooling, Collections, Audit & troubleshoot, and License, Help & support. The main header displays 'TrackMe - Tenant Home' and 'Monitoring the great value in your Big Data'. Below this, there are tabs for 'Fields Quality Monitoring (FQM)', 'Investigate Status Flipping', 'Investigate Audit Changes', and 'Tracking Alerts'. The dashboard features two pie charts: 'Monitored entities count by priority' and 'Monitored entities count by state and priority'. Below these are four KPI cards: 'ENABLED FIELDS QUALITY' (5), 'ANY PRIORITY FIELDS QUALITY IN ALERT' (0), 'HIGH PRIORITY FIELDS QUALITY IN ALERT' (0), and 'CRITICAL PRIORITY FIELDS QUALITY IN ALERT' (0). At the bottom, there's an 'Entities Overview' section with a search bar, filters, and a table with columns: Actions, Entity, Status, Score, Labels, Notes, Priority, Enable..., Status description, and Last metrics. A sidebar menu is open on the right, showing options like 'Open in search', 'Refresh all', 'Artificial Intelligence', 'AI Routines', 'Discovery & detections - Quality trackers', 'Execute: hybrid tracker', 'Create a new quality job', 'Manage: quality trackers', 'Manage: data dictionaries', and a 'Features' section with various management options. Numbered callouts 1 through 5 are placed over the sidebar menu items: 1 over 'Execute: hybrid tracker', 2 over 'Create a new quality job', 3 over 'Manage: quality trackers', 4 over 'Manage: data dictionaries', and 5 over the 'ENABLED FIELDS QUALITY' KPI card.

- 1 **Execute: hybrid tracker**
Run any collect or monitor job on demand, with its execution events.
- 2 **Create a new quality job**
Opens the collect & monitor wizard — CIM, CIM/non-CIM, or monitor-only.
- 3 **Manage: quality trackers**
List trackers and their knowledge objects, delete in bulk.
- 4 **Manage: data dictionaries**
Create, import, export, edit dictionaries; see which trackers use them.
- 5 **Empty KPIs**
Enabled fields, fields in alert by priority, fields not monitored.

01

EXAMPLE 1 · A CIM DATA SOURCE

Cisco ASA firewall logs against Network_Traffic

`index=siem* sourcetype="cisco:asa"` — four regional indexes, one sourcetype, one CIM data model. TrackMe generates the dictionary from CIM; we read the results, fix what is expected, and keep what is a real finding.

Steps 1–2 — Tracker type, name and deployment

Fields Quality Monitoring - Create a new collect job for CIM

Follow these guided steps to define, test and create a new Quality collect and monitor job, scroll down within the screen to navigate through the steps

> Fields Quality Monitoring Workflow Diagram

- 1 Select tracker type
- 2 tracker name & deployment
- 3 Datamodel & search
- 4 CIM parsing & dictionary
- 5 Edit the data dictionary

Select the type of quality tracker you want to create:

Tracker type:

Create a new collect job for CIM (generate or re-use a dictionary)

Create a new collect job for CIM (generate or re-use a dictionary) ✓

Create a new collect job for CIM/non-CIM (create or re-use a dictionary)

Create a monitor job only (advanced, re-use events for aggregation)

- 1 **CIM collect job**
Generate the dictionary from a CIM data model, or reuse one.
- 2 **CIM / non-CIM collect job**
Any search, any format — Example 2.
- 3 **Monitor job only**
Advanced: aggregate events collected elsewhere.

Fields Quality Monitoring - Create a new collect job for CIM

Follow these guided steps to define, test and create a new Quality collect and monitor job, scroll down within the screen to navigate through the steps

> Fields Quality Monitoring Workflow Diagram

- 1 Select tracker type
- 2 tracker name & deployment
- 3 Datamodel & search
- 4 CIM parsing & dictionary
- 5 Edit the data dictionary
- 6 Default thresholds

Enter a name for the new object and define if the deployment is local or a remote account:

cisco-asa

New collect job identifier: (will be added as a prefix to entities, and used to group entities)
A unique numeric suffix is appended automatically — the tracker will be created as: **cisco-asa-0938**

This tracker name is valid and does not exist yet in this tenant.

Splunk deployment, either local or a configured remote account:

local

[Open the configuration interface to add or manage accounts](#)

- 4 **Tracker name — cisco-asa**
Prefix of every entity and job; a unique suffix is added (cisco-asa-0938).
- 5 **Local or remote deployment**
The same wizard targets any remote Splunk account configured in TrackMe.

Step 3 — Data model, node and the search that feeds it

< Fields Quality Monitoring Workflow Diagram

Select tracker type

tracker name & deployment

Datamodel & search

CIM parsing & dictionary

Edit the data dictionary

Default thresholds

Collect job strategy

Collect Job main settings

Validate creation

About defining the search and the breakby fields definition:

Search type:
_raw

Datamodel:
Network_Traffic

NodeName:
All_Traffic

Events resulting from this search will be processed by the TrackMe quality management backend, and indexed in the trackme:fields_quality sourcetype:
Enter the Splunk search

Edit SPL Preview

index=siem* source=disco:asa

Comma separated list of fields used to define entities, you can add a custom field available in the data set as metadata.<fieldname>
Breakby fields definition:
metadatasource,metadata.index,metadata.sourcetype,fieldname

When using a custom break by definition, you should also update the Virtual Tenant groupby setting for splk-fqm, you can do so through the Configuration UI or by running the following command: [update to your tenant_id and breakby fields]
Note that the metadata field name is called with an underscore instead of a dot.
[update to your tenant_id] curl -XPOST https://api.trackme.com/v1/configurations/dsl/update --header "Content-Type: application/json" --data '{"tenant_id": "[tenant_id]", "breakby_fields": "[breakby_fields]"}

- 1 Search type**
_raw for an event search; generating for a tstats or custom generating search.
- 2 Data model — Network_Traffic**
The CIM model the events should comply with.
- 3 Node — All_Traffic**
Picked automatically from the model; drives the recommended fields.
- 4 The search**
index=siem* sourcetype="cisco:asa" — four regional indexes at once.
- 5 Break-by fields**
Data model, node, index, sourcetype, field: one entity per field per index.

Because index is part of the break-by, a parsing problem on one region shows up on that region only — it is not averaged away by the other three.

Step 4 — Name the dictionary, simulate on 10,000 events

Fields Quality Monitoring - Create a new collect job for CIM

Follow these guided steps to define, test and create a new Quality collect and monitor job, scroll down within the screen to navigate through the steps

> Fields Quality Monitoring Workflow Diagram

- 1 Select tracker type
- 2 tracker name & deployment
- 3 Datamodel & search
- 4 CIM parsing & dictionary
- 5 Edit the data dictionary
- 6 Default thresholds
- 7 Collect job strategy
- 8 Collect job main settings
- 9 Validate creation

> About CIM parsing preview and dictionary configuration:

AI dictionary generation — complete these steps first
Complete these steps to enable the **Generate dictionary with AI** button:

- ✓ Pick "Create new" in the dictionary option below
- ✓ Enter a name for the new dictionary
- ❑ Click "Simulate the search to continue" to sample fields

Generate dictionary with AI

Dictionary: (create new or use existing) **1** Create new

New dictionary name? **2** Network_Traffic:cisco_asa

Allow unknown by default? **3** False

Allow missing/null by default? False

Event limit: **4** 10000

% success fields threshold (simulation only): **5** 85%

6 Simulate to continue

1 Create new, or reuse

A dictionary is shared: reuse it for every tracker on the same format.

2 Network_Traffic:cisco_asa

The new dictionary name.

3 Defaults for the generated rules

Allow unknown, allow missing/null — false by default.

4 Event limit — 10,000

How many events the simulation scores.

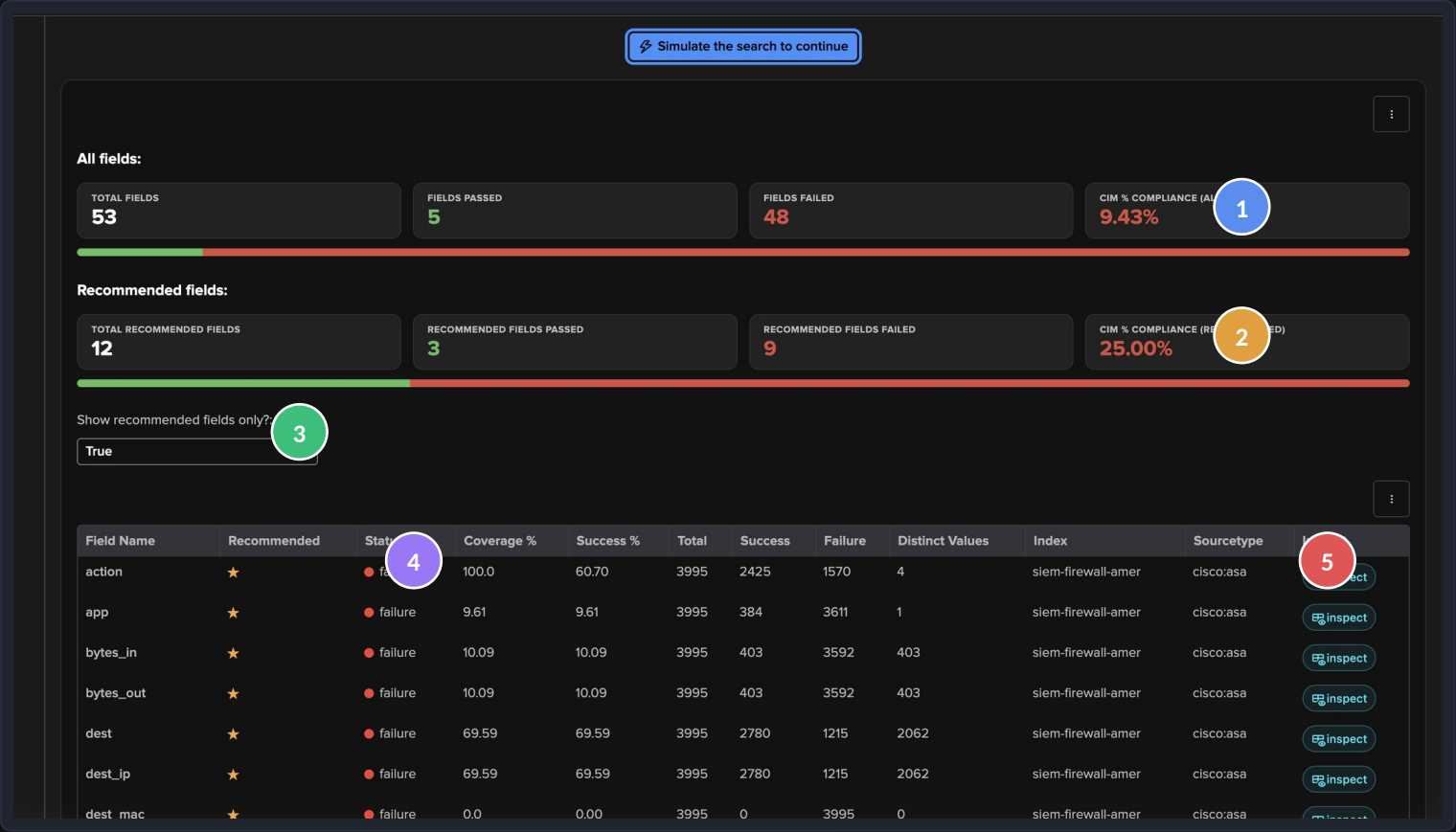
5 Simulation threshold

The success % a field needs to pass — 85 by default.

6 Simulate the search

One search job: the whole scoring in seconds, nothing written.

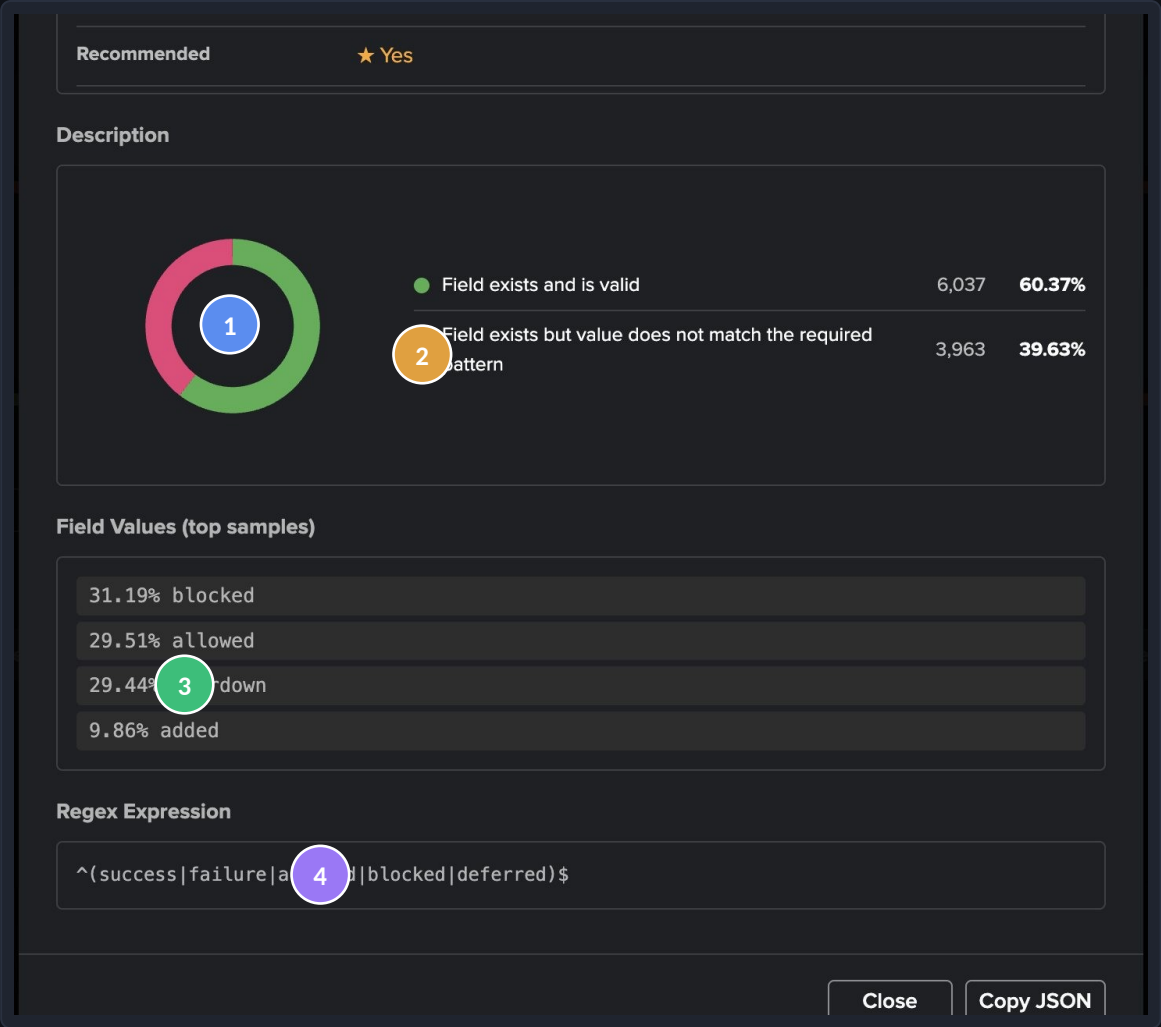
First verdict — 25% CIM compliance on the recommended fields



- 1 All fields: 53 checked, 5 passed**
Every field of the Network_Traffic model — 9.43%. Most are irrelevant for a firewall.
- 2 Recommended fields: 3 of 12**
The fields CIM recommends for All_Traffic — the number that matters: 25%.
- 3 Recommended only**
Filter the table to the 12 recommended fields (star column).
- 4 Per field and per index**
Coverage %, success %, totals, failures, distinct values.
- 5 Inspect**
Opens Field Details for one field — next two slides.

A low first score is normal on CIM. The job now is to separate real compliance defects from fields that simply do not exist on every message type.

A real CIM defect — action fails on the regular expression



- 1

Valid — 60.4%
Present on 100% of events, below the 85% threshold: status failure.
- 2

Does not match the pattern — 39.6%
The field exists, the value is wrong.
- 3

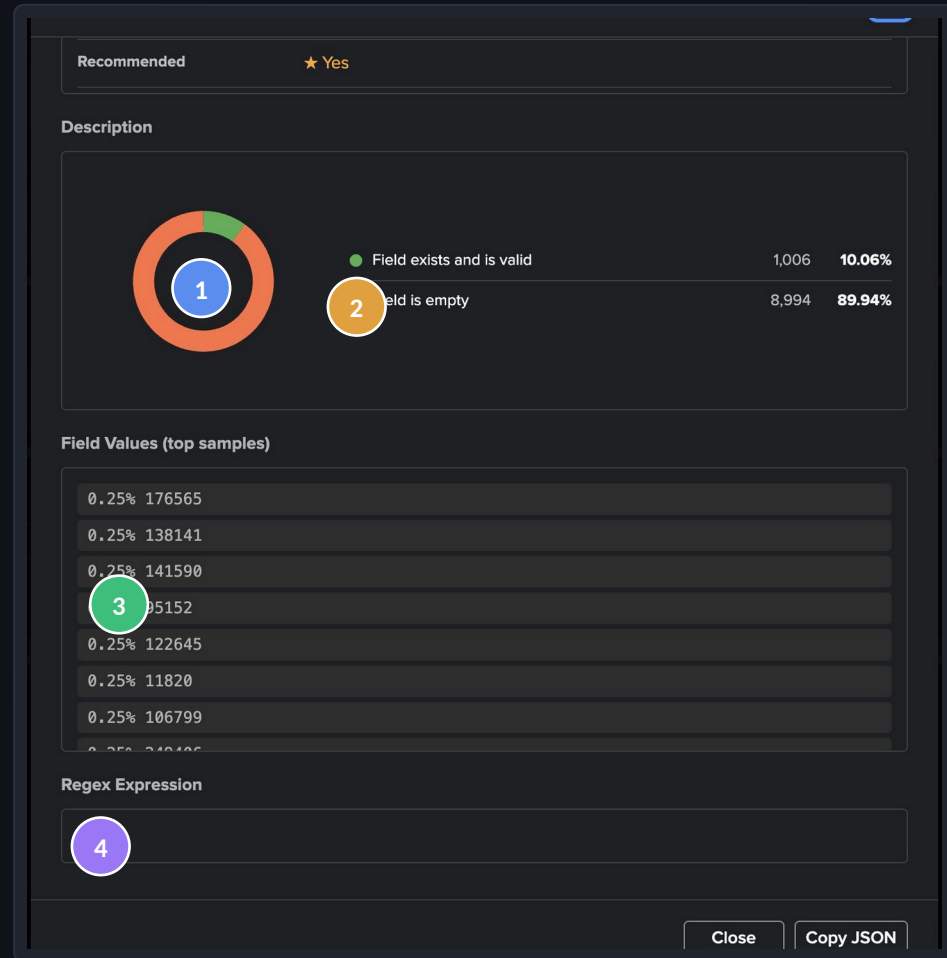
The evidence
blocked 31% · allowed 30% · teardown 29% · added 10%. teardown and added are not CIM actions.
- 4

The rule, from the CIM model
^(success|failure|allowed|blocked|deferred)\$

Why it matters

CIM-based content (correlation searches, dashboards) that filters on action=allowed or action=blocked silently misses 40% of these events.

An expected gap — bytes_in only exists on some messages



- 1 Field exists and is valid — 10%**
Every value present passes `^\\d+$`.
- 2 Field is empty — 90%**
No value on most events — no parsing problem.
- 3 Values are healthy**
Hundreds of distinct byte counts, all numeric.
- 4 Regex `^\\d+$`**
Correct — keep it, and add a type check.

Read the category, not only the score

Two failures, two different answers:

action → value does not match: a real CIM defect, keep the rule strict and alert on it.

bytes_in → value missing on most events by design: allow missing, keep validating the values that are there.

The same reading applies to app, user and dest_mac on this feed.

Step 5 — Edit the data dictionary in place

fine, test and create a new Quality collect and monitor job, scroll down within the screen to navigate through the steps

Workflow Diagram

ant

> About this preview and data dictionaries:

Review and edit the dictionary used by this tracker, including dictionaries generated with AI. Re-run the simulation after saving to check the updated rules.

Restrict to recommended fields?:

True

Unsaved dictionary changes

1

Cancel before continuing.

Save now

Edit, add or remove fields, then save. Type and regex are both checked. Any preserves existing validation.

Add field

Select	Field	Allow unknown	Allow missing/null	Type	Regex	Advisor
☐	action	No	No	any	^(success failure all ... ked deferred)\$	
☐	app	No	2	any	^[w.-\d\s]+	AI
☐	bytes	No	No	any	^\d+\$	AI
☐	bytes_in	No	Yes	3	^\d+\$	AI
☐	bytes_out	No	Yes	integer	^\d+\$	AI
☐	dest	No	No	any	^[w.-\d\s]+	AI
☐	dest_ip	No	No	any	^\d(1,3)\.\d(1,3)\.\d(1,3)\.\d(1,3)\$	AI
☐	dest_mac	No	Yes	any	^[a-f0-9]{2}[a-f0-9]{2}[a-f0-9]{2}[a-f0-9]{2}[a-f0-9]{2}[a-f0-9]{2}	AI

- 1

Unsaved changes banner
Edits stay local until Save now; steps and simulation are blocked meanwhile.
- 2

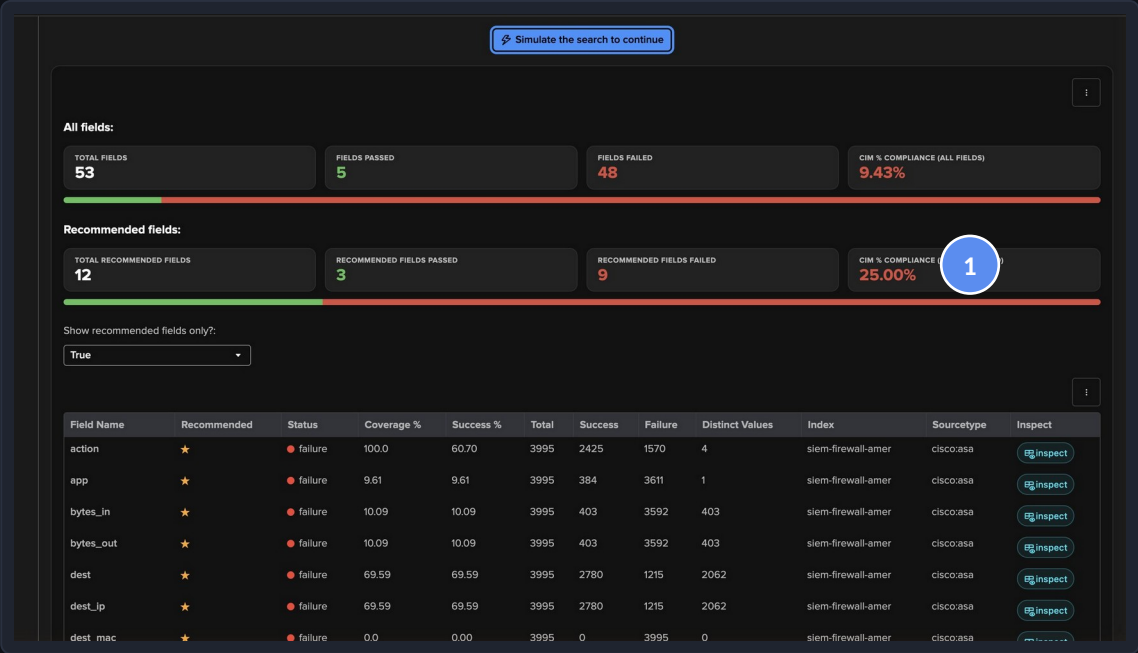
Allow missing/null: Yes
app, bytes_in, bytes_out, user, dest_mac — present only on some message types.
- 3

Type: integer
bytes_in and bytes_out: regex and type are both checked.
- 4

action stays strict
The CIM regex is kept — it is a real finding.
- 5

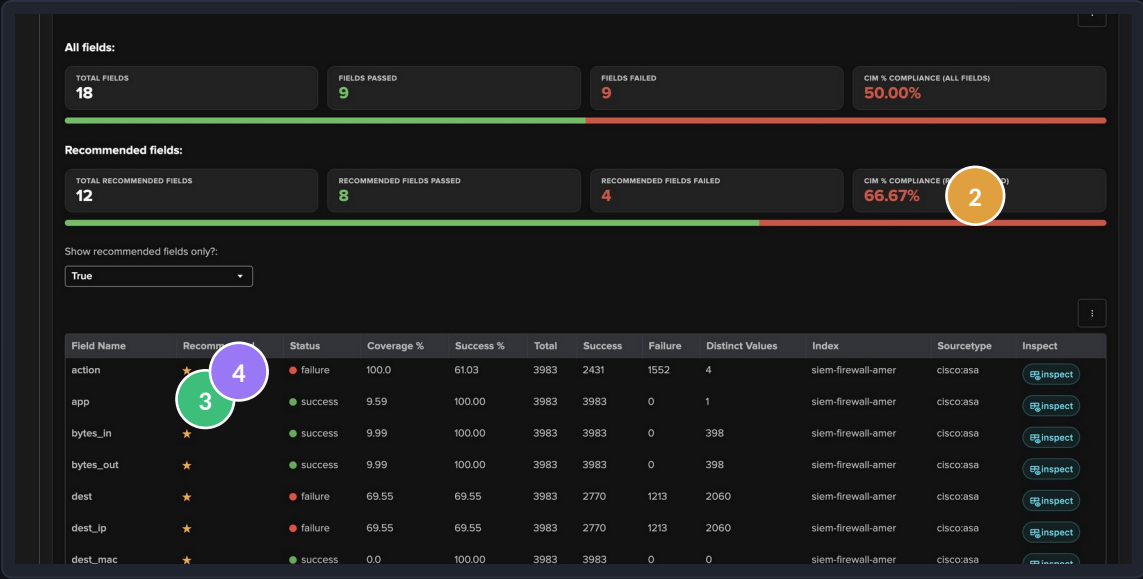
Per-field AI
Ask the AI field advisor about one rule — used in Example 2.

Before and after — 25% to 66.7%, and the right fields still fail



BEFORE — dictionary as generated from CIM

- 1 25.00% — 3 of 12 recommended fields pass
Expected gaps and real defects mixed together.



AFTER — 5 fields allow missing, 2 typed

- 2 66.67% — 8 of 12 pass
Expected gaps are now accepted.
- 3 bytes_in / bytes_out / app pass
Values still checked when present.
- 4 action still fails
The real CIM defect stays visible.

Step 6 — Default thresholds and impact score

Fields Quality Monitoring - Create a new collect job for CIM

Follow these guided steps to define, test and create a new Quality collect and monitor job, scroll down within the screen to navigate through the steps

> Fields Quality Monitoring Workflow Diagram

- 1 Select tracker type
- 2 tracker name & deployment
- 3 Datamodel & search
- 4 CIM parsing & dictionary
- 5 Edit the data dictionary
- 6 Default thresholds**
- 7 Collect job strategy
- 8 Collect job main settings
- 9 Validate creation

> About thresholds in TrackMe fields quality:

Default thresholds for fields:

When discovering fields, TrackMe will preset a threshold value for each field, which can be easily adjusted in the entity modification screen.

The percentage of success represents the amount of times the field passed the quality controls in the sampled events. (formula: count of success/count of total * 100)

Default threshold for fields: **85%**

0 **1** 100

Impact score for fields:

The Impact score (0-100) assigned when a field threshold is breached. An entity is green if the accumulated score is 0, orange if >0 and <100, red if >=100.

Default impact score for fields: **100**

0 **2** 100

Default thresholds for the global entity:

TrackMe creates a **@global** entity for each context, called @global, which is the aggregation of all fields.

The default threshold **3** is the minimum percentage of passed fields. (formula: passed fields / total fields checked * 100)

Default threshold for the global entity: **100%**

- 1 Field threshold — 85%**
Minimum success % per field. Adjustable per entity later.
- 2 Impact score — 100**
Score added when a field breaches: 100 = red, below 100 = orange.
- 3 Global threshold — 100%**
The @global entity: share of fields that must pass in the context.

Thresholds reuse TrackMe's hybrid scoring: a field below threshold adds its impact score to the entity, and the usual states, stateful alerts and emails follow — no FQM-specific alerting to learn.

Step 7 — Sampling, sized by Auto define

Measures event counts across the full search window. Targets 95% of the hard limit to reduce truncation. This search may take time on large datasets.

1 Suggested 1:5: approximately 8,154 sampled events from 40,772 events across 49 populated time buckets.

Sampling is random and future volume can change. **1** Run the benchmark after applying to verify time coverage.

Use suggested value **2**

Benchmark sampling mode

3 7.30 seconds runtime

4 8,060 events retrieved during the benchmark — 80.6% of the hard limit (10,000)

5

events_count

- 1** Auto define

Counts every event of the search window, per time bucket, and targets 95% of the hard limit.
- 2** Suggested 1:5

40,772 events in 49 buckets → ~8,154 sampled. One click applies it as a custom ratio.
- 3** Benchmark: 7.3 s

The real collect search with the new ratio.
- 4** 8,060 events — 80.6% of the limit

Close to the target, with headroom for volume growth.
- 5** The whole 24 hours

Every bucket of the day is represented.

Deterministic

Auto define is deterministic — a count per bucket, no AI, no guess. Re-run it when the volume of the feed changes significantly.

Why sampling — head mode only sees the last few hours

Event limit:

10000

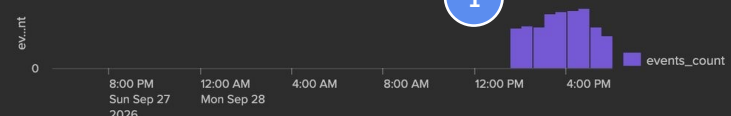
Benchmark head mode

5.21 seconds runtime

Events distribution over time

10,000 events retrieved during the benchmark — 100% of the event limit (10,000)

1



index	sourcetype	count
siem-firewall-amer	cisco:asa	3985
siem-firewall-apac	cisco:asa	1994
siem-firewall-emea	cisco:asa	2031
siem-firewall-latam	cisco:asa	1990

4

performance.

Sampling value:

Custom

Hard limit value:

10000

Custom value:

5

2

fine

Measures event counts across the full search window. Targets 95% of the hard limit to reduce truncation. This search may take time on large datasets.

i

Suggested 1:5: approximately 8,154 sampled events from 40,772 events across 49 populated time buckets.

Sampling is random and future volume can change. Run the benchmark after applying to verify time coverage.

Use suggested value

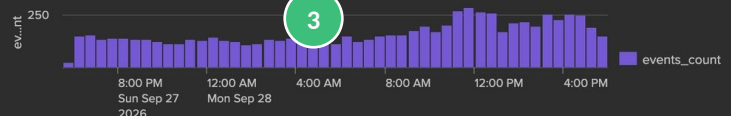
Benchmark sampling mode

7.30 seconds runtime

Sampled events distribution over time

8,060 events retrieved during the benchmark — 80.6% of the hard limit (10,000)

3



- 1

Head mode: 10,000 events
100% of the limit, all in the last hours of the window.
- 2

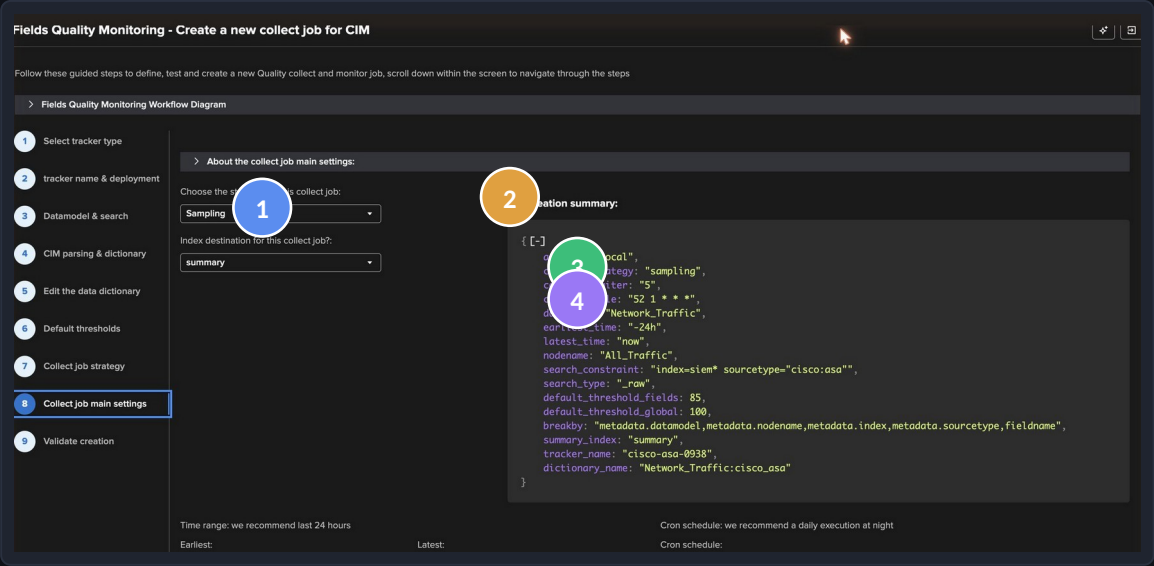
Sampling 1:5 (Auto define)
Custom ratio applied from the suggestion.
- 3

Sampling: 8,060 events
Evenly spread across the 24 hours.
- 4

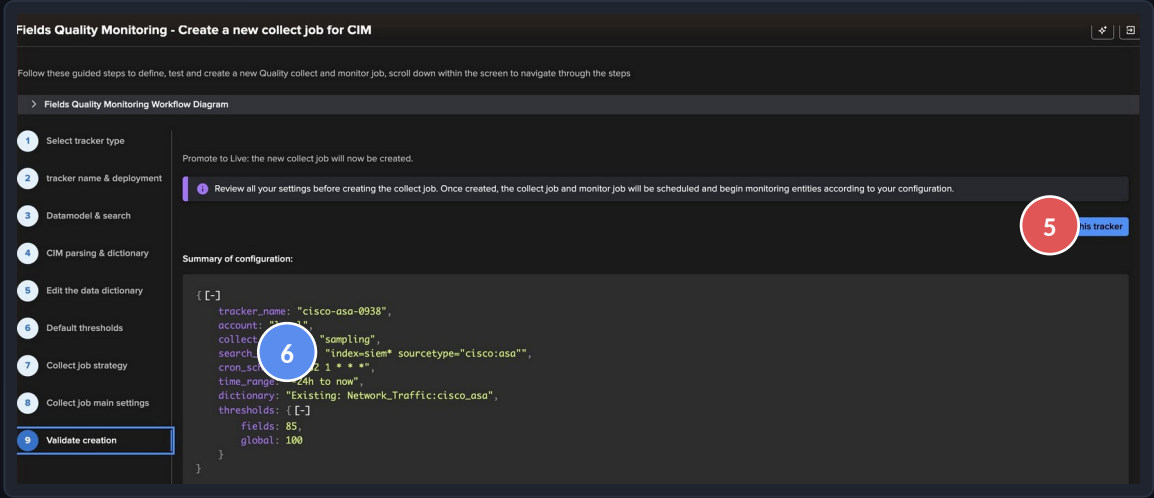
Per-index breakdown
Each benchmark lists events per index and sourcetype.

Same search, same window, same limit — only the collect strategy differs.

Steps 8–9 — Main settings, review, create



- 1 Strategy: Sampling**
Carried from step 7 — collect_limiter 5.
- 2 Job creation summary**
The exact definition that will be created.
- 3 collect_strategy: sampling, limiter 5**
The ratio chosen with Auto define; last 24 hours window.
- 4 Daily cron at night**
52 1 * * * here; the monitor job follows ~10 minutes later.



- 5 Create this tracker**
Creates and schedules the collect and monitor jobs.
- 6 Final summary**
Tracker cisco-asa-0938, sampling, dictionary, thresholds 85 / 100.

No need to wait for the night — run the jobs now

The screenshot shows the 'Trackers: hybrid trackers' window. It has a title bar with a maximize button and a close button. The main content area is divided into two sections. The top section, labeled 'Choose an active hybrid tracker and click', contains a dropdown menu with the selected option 'cisco-asa-0938_30e1b (trackme_fqm_collect_cisco-asa-0938_30e1b_tracker_tenant_fi...' and a blue button labeled 'Run tracker now'. The bottom section, labeled 'Choose the hybrid tracker:', contains a table titled 'Execution Events (auto-refresh 15s)'. The table has two columns: 'Time' and 'Event'. It displays two log entries. The first entry is from 9/28/2026 at 5:57:33.344 PM, showing a successful execution of the 'trackme_fqm_collect_cisco-asa-0938_30e1b_wrapper_tenant_field-quality' job, generating 427 tenant IDs and scoring 8192 events. The second entry is from 9/28/2026 at 5:57:22.693 PM, showing a successful execution of the 'trackme_fqm_collect_cisco-asa-0938_30e1b_wrapper_tenant_field-quality' job, generating 383 tenant IDs and scoring 76 entities. A 'Close' button is located at the bottom right of the window.

1 Execute: hybrid tracker

Pick the collect job, then the monitor job.

2 Run tracker now

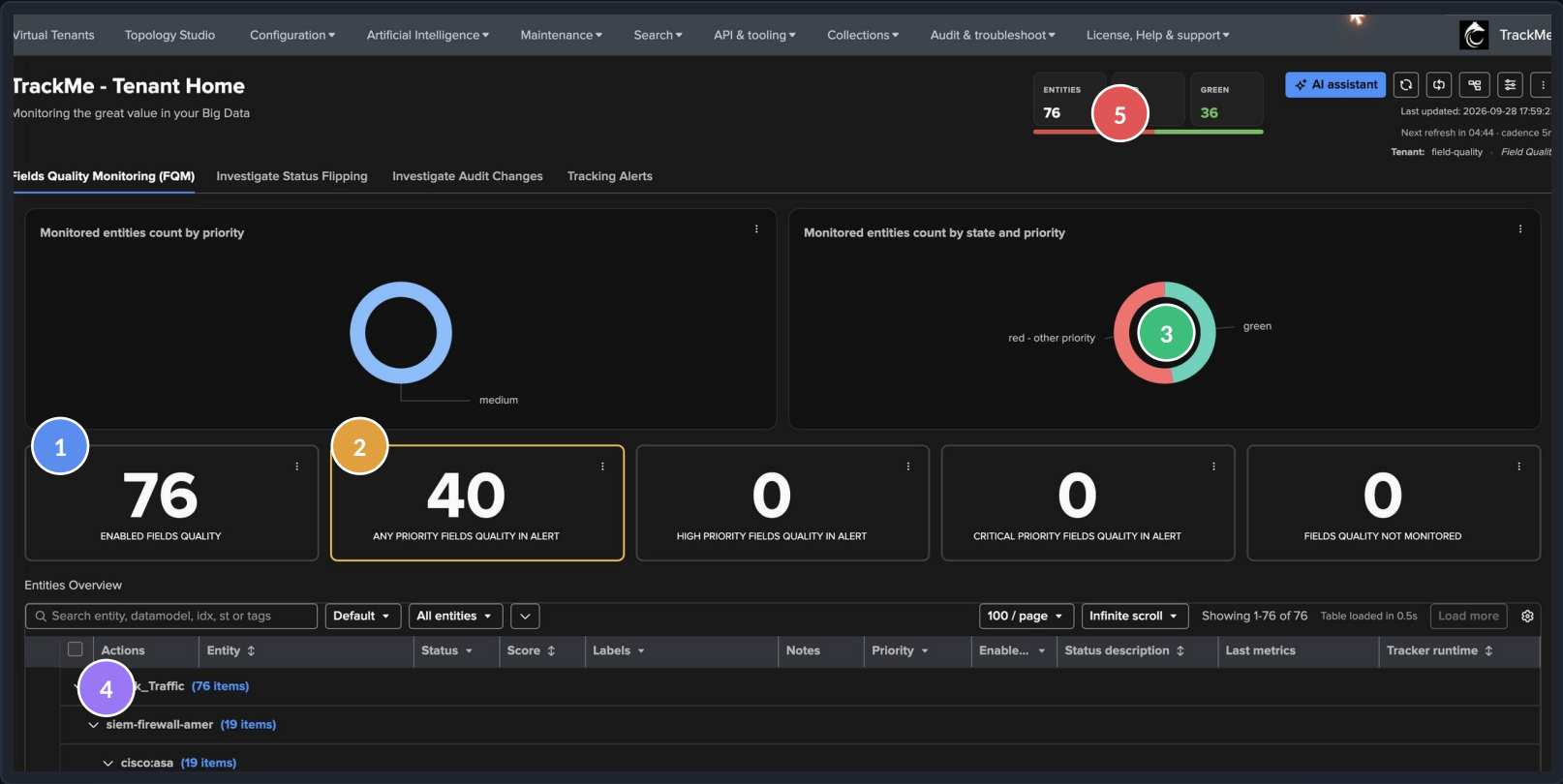
Runs the job as the scheduler would, with its time window.

3 Execution events

Collect: success in 11.1 s, 8,192 events scored. Monitor: success in 24 s, 76 entities.

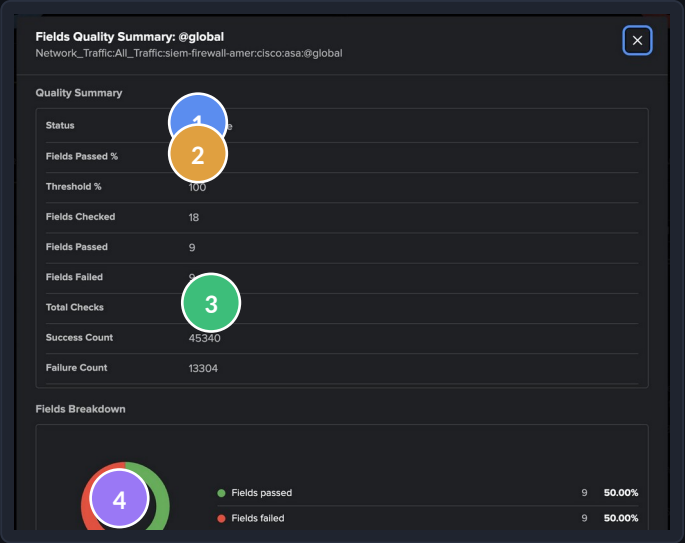
76 entities = 4 indexes × (18 dictionary fields + one @global). From now on both jobs run daily on their own.

76 field entities — grouped by data model, index and sourcetype



- 1 **76 enabled fields**
Every field of the dictionary, per index.
- 2 **40 in alert**
Fields below threshold — and one @global per index.
- 3 **State by priority**
Classify with priority, tag and SLA policies like any entity.
- 4 **Hierarchy**
Network_Traffic → index → sourcetype → field.
- 5 **Header KPIs**
Entities · red · green, always visible.

The @global entity — the context in one view



Metadata	
Index	siem-firewall-amer
Sourcetype	cisco:asa
Datamodel	Network_Traffic
Node Name	All_Traffic

Fields (18)

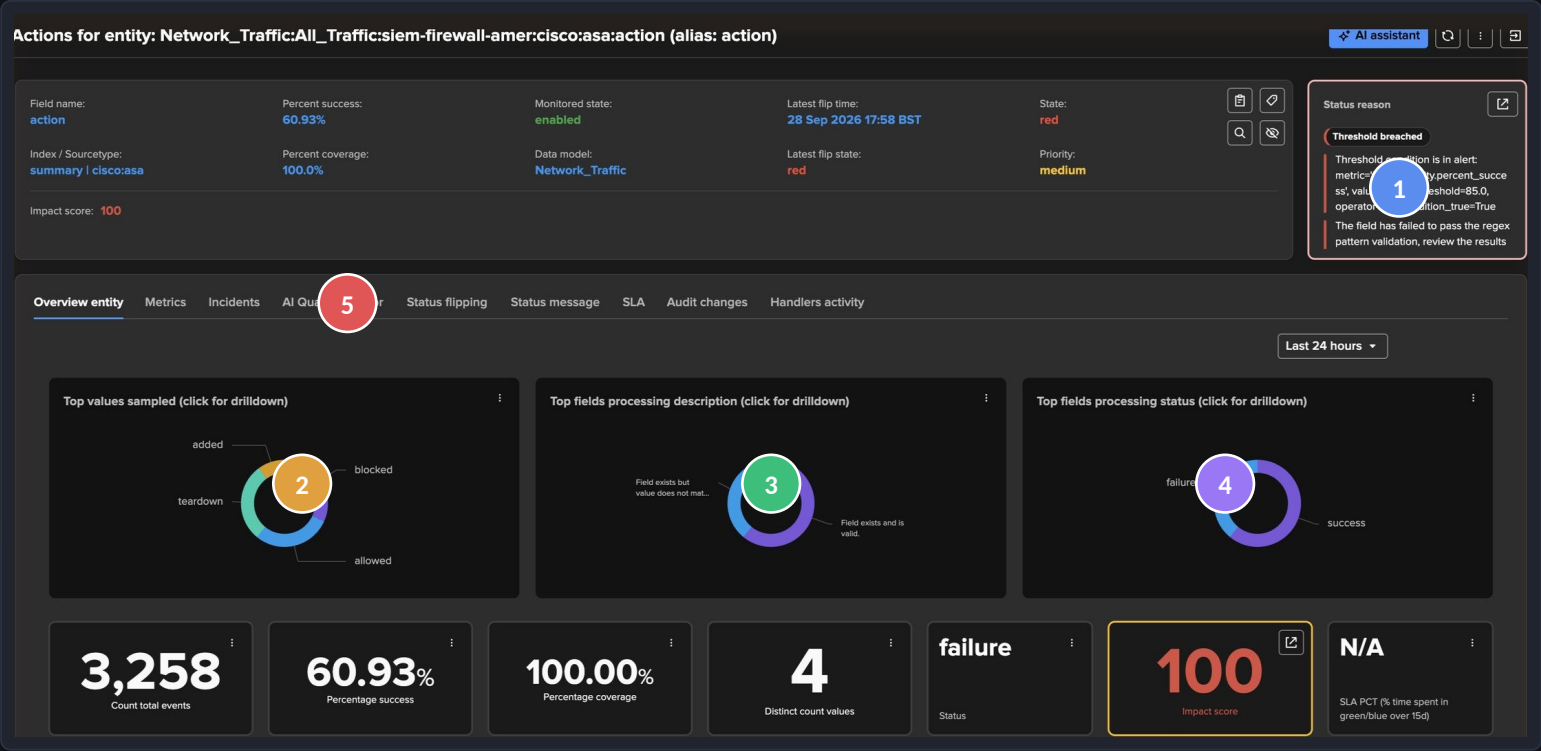
Q Filter fields

Field	Status	Success %	Coverage %	Details
action	Failure	60.93	100	Details
bytes	Failure	30.66	30.66	Details
dest	Failure	68.54	68.54	Details
dest_ip	Failure	68.54	68.54	Details
dest_port	Failure	68.54	68.54	Details
protocol	Failure	68.54	68.54	Details
rule	Failure	10.37	10.37	Details

Raw JSON Copy JSON Close

- 1 Status: failure**
9 of 18 fields pass — 50% against a 100% threshold.
- 2 Fields passed %**
The @global metric the threshold is evaluated on.
- 3 58,644 checks**
Every field of every sampled event: 45,340 successes, 13,304 failures.
- 4 Fields breakdown**
Passed vs failed at a glance.
- 5 Filter fields**
Find one field in large dictionaries.
- 6 Failed first**
action, bytes, dest, dest_ip, dest_port, protocol, rule...
- 7 Drill down**
Opens Field Details of that field.

The entity — action, red, and why



- 1 **Status reason**
Threshold breached: percent_success 60.93 against 85 — the field failed the regex validation.
- 2 **Top values sampled**
blocked · allowed · teardown · added — click to drill down.
- 3 **Processing description**
Valid vs value does not match the pattern.
- 4 **Processing status**
Success vs failure over the period.
- 5 **AI Quality Advisor tab**
The FQM Advisor, per entity — Example 2.

The three charts read the collected samples for this field only, so they stay fast on large trackers.

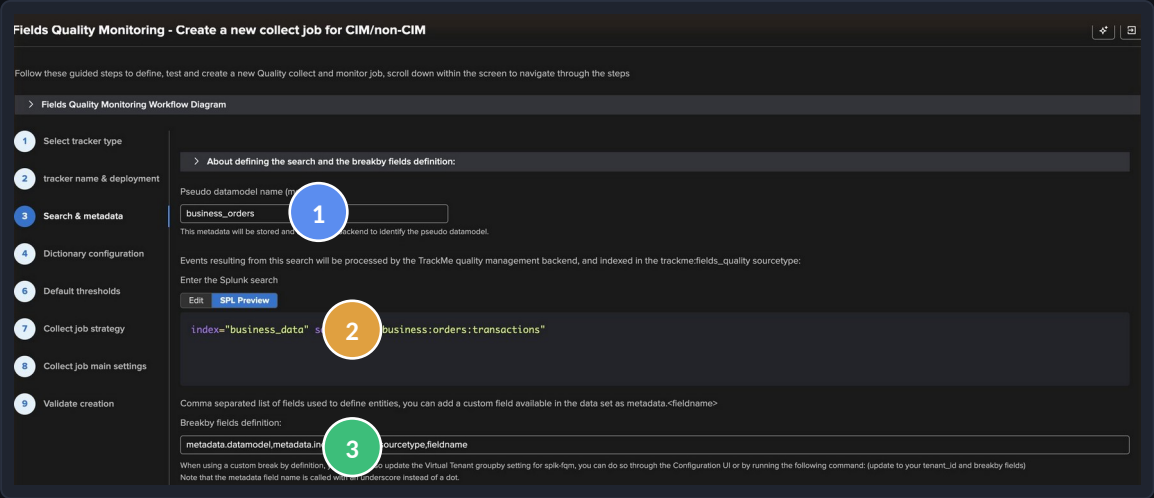
02

EXAMPLE 2 · A NON-CIM DATA SOURCE

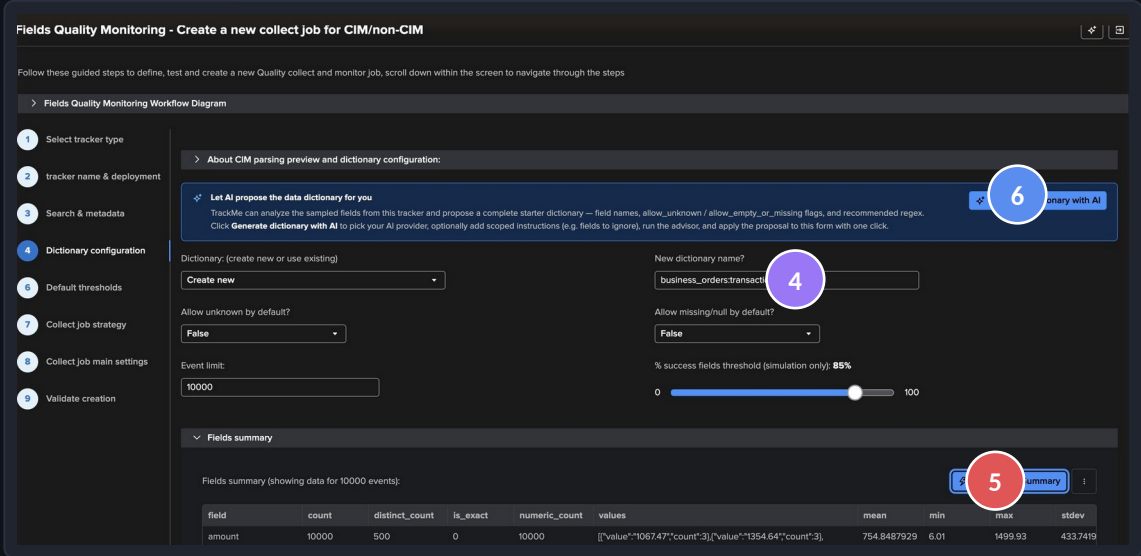
E-commerce orders — a dictionary drafted by AI

`index="business_data" sourcetype="business:orders:transactions"` — JSON order events, no CIM model. The AI Advisors draft the dictionary from the sampled fields, tighten one rule on request, and review the verdict.

Option 2 — any search, a pseudo data model, the sampled fields



- 1 **Pseudo data model — business_orders**
Top of the entity hierarchy, like a CIM model.
- 2 **The search**
index="business_data" sourcetype="business:orders:transactions"
- 3 **Break-by**
Same identity as CIM: model, index, sourcetype, field.



- 4 **New dictionary — business_orders:transactions**
Created by this job, reusable by others.
- 5 **Load fields summary**
fieldsummary over 10,000 events.
- 6 **Generate dictionary with AI**
Enabled once the fields summary is loaded.

Generate the dictionary with AI — scoped by your instructions

Generate dictionary with AI

The AI will analyze **16 dictionary-eligible fields** from **orders** (raw) and propose a starter data dictionary — field names, allow_unknown / allow_empty_or_missing flags and recommended regex and value type.

Splunk metadata and internal fields (e.g. index, source, sourcetype, host, *) are excluded, so this can be fewer than the tracker's total field count.

AI PROVIDER

google_gemini (gemini-pro-latest)

ADDITIONAL INSTRUCTIONS (optional)

Steer the proposal — e.g. fields to ignore, coverage thresholds, or naming preferences. Leave empty to let the AI propose from all sampled fields.

Only propose these fields, all present in the JSON payload: amount, channel, currency, customer_id, items, order_id, payment_method, region, shipping, status, timestamp, transaction_id. Ignore any other field.

Generate dictionary

Generate dictionary with AI

The AI will analyze **16 dictionary-eligible fields** from **orders** (raw) and propose a starter data dictionary — field names, allow_unknown / allow_empty_or_missing flags and recommended regex and value type.

Splunk metadata and internal fields (e.g. index, source, sourcetype, host, *) are excluded, so this can be fewer than the tracker's total field count.

AI PROVIDER

google_gemini (gemini-pro-latest)

ADDITIONAL INSTRUCTIONS (optional)

Steer the proposal — e.g. fields to ignore, coverage thresholds, or naming preferences. Leave empty to let the AI propose from all sampled fields.

Only propose these fields, all present in the JSON payload: amount, channel, currency, customer_id, items, order_id, payment_method, region, shipping, status, timestamp, transaction_id. Ignore any other field.

Analyzing the sampled fields and drafting the dictionary... (101s)

Agent runtime started

Running reasoning step 2

- 1

16 dictionary-eligible fields

From the sampled events; metadata excluded.
- 2

AI provider

Any provider configured in TrackMe.
- 3

Additional instructions

amount, channel, currency, customer_id, items, order_id, payment_method, region, shipping, status, timestamp, transaction_id.
- 4

Generate dictionary

Starts an AI Advisor run.
- 5

Live progress

Agent runtime and reasoning steps, streamed.

The advisor proposes — nothing is written until you apply.

The proposal — a reasoned rule per field, applied in one click

1

SUMMARY

The data source is a raw tracker for business orders transactions (index="business_data", sourcetype="business:orders:transactions"). The sample shows well-structured e-commerce events with clean enums for channel, payment_method, region, shipping, and status, alongside numeric fields for amount and items. However, fields like currency and transaction_id contain significant data quality issues (empty strings, 'NULL', 'ERROR', 'N/A'), requiring permissive rules to avoid excessive false positives.

2

PROPOSED DICTIONARY ENTRIES (16)

amount

Type: numerical

Field is purely numeric (10000 of 10000 values). Proposed type numerical with no regex.

3

channel

Type: any

regex: ^(mobile_android|mobile_ios|web|api|kiosk)\$

Observed 5 distinct values across 10000 events. Proposed enum regex.

4

currency

Type: any

Observed 21 distinct values including empty strings and dirty values (e.g. 'ERROR', 'N/A'). Proposed type any with allow_unknown and allow_empty rules.

About this preview and data dictionaries:

Review and edit the dictionary used by this tracker, including dictionaries generated with AI. Re-run the simulation after saving to check the updated rules.

No existing trackers use this dictionary.

Edit, add or remove fields, then save. Type and regex are both checked. Any preserves existing validation.

Add field

Select	Field	Allow unknown	Allow missing/null	Type	Regex	AI
☐	amount	No	No	any	.	AI
☐	channel	No	No	any	^(mobile_android mobile_ios web api kiosk)\$	AI
☐	currency	Yes	Yes	any	.	AI
☐	customer_id	No	No	any	^CUST-\d+\$	AI
☐	items	No	No	numerical	.	AI
☐	order_id	No	No	any	^ORD-\d{4}-\d+\$	AI
☐	payment_method	No	No	any	^(paypal debit_card google_pay apple_pay bank)	AI

What the advisor got right

Enumerations became strict regexes, numeric fields got a type, IDs got recogniser patterns (ORD-\d{4}-\d+\$, CUST-\d+\$). Minutes of work instead of an afternoon of reading events.

- 1 **Summary of the data**
Enumerations, numerics — and dirty currency and transaction_id values.
- 2 **amount — numerical**
Purely numeric in 10,000 of 10,000 values.
- 3 **channel — enumeration**
^(mobile_android|mobile_ios|web|api|kiosk)\$
- 4 **currency — permissive**
Allow unknown and empty "to avoid false positives" — we will challenge that.
- 5 **Applied to the wizard**
Saved as a real dictionary, 12 fields kept.
- 6 **AI per field**
Every row can ask the AI field advisor.

Tighten one rule — ask the AI field advisor about currency

AI field advisor — currency

Review the current rules for **currency** using any available sample statistics. Describe the improvement you want below. Without samples, suggestions are based on the rules and your instructions.

AI PROVIDER

google_gemini (gemini-pro-latest)

ADDITIONAL INSTRUCTIONS (optional)

Steer the proposal — e.g. fields to ignore, coverage thresholds, or naming preferences. Leave empty to let the AI propose from all sampled fields.

Currency must be a valid ISO 4217 code (USD, EUR, GBP, JPY, CAD, AUD). Empty, NULL, ERROR or placeholder values are data quality defects and must fail. Do not allow unknown or missing values.

Same pattern as the dictionary advisor: propose, explain, apply only on consent.

Steer the proposal — e.g. fields to ignore, coverage thresholds, or naming preferences. Leave empty to let the AI propose from all sampled fields.

Currency must be a valid ISO 4217 code (USD, EUR, GBP, JPY, CAD, AUD). Empty, NULL, ERROR or placeholder values are data quality defects and must fail. Do not allow unknown or missing values.

SUMMARY

Refined the currency field based on operator instructions to enforce strict ISO 4217 compliance. The sample data contains a mix of valid currency codes and numerous data quality defects (empty strings, NULL, ERROR, placeholders) which will now be correctly flagged as failures.

PROPOSED DICTIONARY ENTRIES (1)

currency

Rule: any

Regex: ^{(USD|EUR|GBP|JPY|CAD|AUD)}\$

Sample shows 10,000 events with 21 distinct values, including valid codes (CAD, GBP, EUR, AUD, JPY, USD) and many invalid placeholders (NULL, ERROR, empty strings). Per operator instructions, restricted to valid ISO 4217 codes and disabled empty/unknown allowances to flag data quality defects.

CONFIDENCE NOTES

The regex strictly enforces the 6 requested ISO codes. Note that a significant portion of the sampled data (e.g., empty strings, 'NULL', 'ERROR', '\$\$') will fail validation, which aligns with the instruction to treat these as data quality defects.

Reasoning trace

Apply to field

- 1

One field, one question
AI field advisor — currency, with the sample statistics.
- 2

The instruction
Valid ISO 4217 codes only; empty, NULL, ERROR are defects and must fail.
- 3

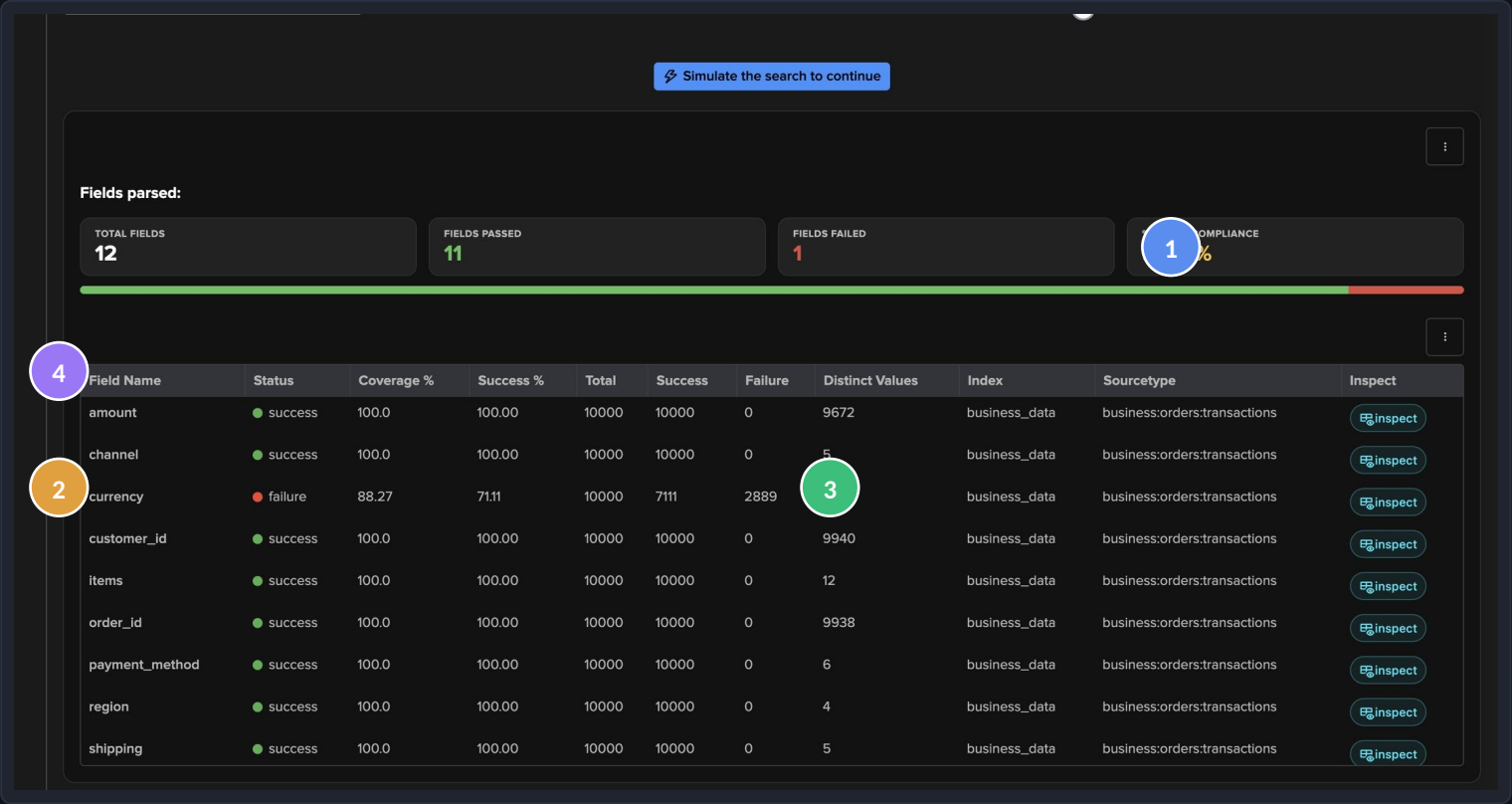
Summary
Why the rule changes.
- 4

The rule
^{USD|EUR|GBP|JPY|CAD|AUD}\$ — allow unknown and missing off.
- 5

Confidence notes
The advisor states the impact on the sample up front.
- 6

Apply to field draft
Updates the row; saved with the dictionary.

11 of 12 fields pass — currency is the business finding



- 1 **91.67% field compliance**
12 fields checked, 11 passed, 1 failed.
- 2 **currency — failure**
Coverage 88.3%, success 71.1%.
- 3 **2,889 failures in 10,000 orders**
Almost 29% of the orders carry an unusable currency.
- 4 **Everything else is clean**
IDs, enumerations, amounts and timestamps: 100%.

Why it matters

For finance and reporting teams this is not a parsing detail: revenue by currency is wrong for more than a quarter of the orders.

currency — wrong codes, spelled-out words, placeholders, blanks



- 1 Field exists and is valid — 71.1%
CAD, EUR, GBP, AUD, JPY, USD: 12.7% to 14% each.
- 2 Value does not match — 23.3%
Present, but not an ISO 4217 code.
- 3 The failing values
G · US · EU · XYZ · DOLLAR · \$\$ · EURO · NULL · - - —
each 1-3% of the orders.
- 4 Field is empty — 5.6%
No currency at all.

Same view everywhere

The same view exists in the wizard (simulation → inspect) and in Tenant Home (Last metrics → inspect) once the tracker runs.

Auto define again — then 13 entities in the tenant

Measures event counts across the full search window. Targets 95% of the hard limit to reduce truncation. This search may take time on large datasets.

1

Suggested 1:9: approximately 8,637 sampled from 77,730 events across 49 populated time buckets.

Sampling is random and future volume can change. Run the benchmark after applying to verify time coverage.

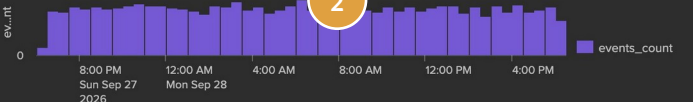
Use suggested value

Benchmark sampling mode

3.19 seconds runtime

Sampled events distribution over time

8,534 events retrieved during the benchmark — 85.3% of the hard limit (10,000)



index	sourcetype	count
business_data	business:orders:transactions	8676

- 1

Suggested 1:9
77,730 events → ~8,637 sampled, 49 buckets.
- 2

8,534 events, 3.2 s
Spread over the full 24 hours.

89
ENABLED FIELDS QUALITY

42
ANY PRIORITY FIELDS QUALITY IN ALERT

0
HIGH PRIORITY FIELDS QUALITY IN ALERT

0
CRITICAL PRIORITY FIELDS QUALITY IN ALERT

0
FIELDS QUALITY NOT MONITORED

Entities Overview

business_orders

Default

All entities

100 / page

Infinite scroll

Showing 1-13 of 13

Table loaded in 0.3s

Load more

Actions	Entity	Status	Score	Labels	Notes	Priority	Enable...	Status description	Last metrics	Tracker runtime
	business_orders (13 items)									
	business_data (13 items)									
	business:orders:transactions (13 items)									
	@global		100	+ label		medium		% success: 91.67, fields passed: 11, fields failed: 1	inspect	28 Sep 2026 17:27 UTC
	amount		0	+ label		medium		% success: 100, % coverage: 100	inspect	28 Sep 2026 17:27 UTC
	channel		0	+ label		medium		% success: 100, % coverage: 100	inspect	28 Sep 2026 17:27 UTC
	currency		100	+ label		medium		% success: 71.97, % coverage: 88.7	inspect	28 Sep 2026 17:27 UTC
	customer_id		0	+ label		medium		% success: 100, % coverage: 100	inspect	28 Sep 2026 17:27 UTC
	items		0	+ label		medium		% success: 100, % coverage: 100	inspect	28 Sep 2026 17:27 UTC
	order_id		0	+ label		medium		% success: 100, % coverage: 100	inspect	28 Sep 2026 17:27 UTC

- 3

business_orders → business_data → business:orders:transactions
Same hierarchy as CIM.
- 4

@global — 91.67%
11 of 12 fields pass.
- 5

currency — red
success 71.97%, coverage 88.7%.

Ask about the entity — the Assistant hands over to the FQM Advisor

AI Assistant —
business_orders:business_data:business:orders:transactions:currency
(alias: currency)

google_gemini (gemini-pro-late)

Entity: business_orders:business_data:business:orders:transactions:currency (alias: currency) (FQM)

You

Which currency values are failing, what share of the orders is affected, and what should the application team fix?

AI Assistant

Based on the Fields Quality Monitoring (FQM) analysis for the currency field, here is the breakdown of the issue:

1. Which currency values are failing? The field expects strict 3-letter ISO currency codes based on the configured regex pattern: `^([UR|GBP|JPY|CAD|AUD])$`. The values currently failing this validation include:

- Truncated/Malformed codes: G, US, EU
- Non-standard formats: DOLLAR, EURO, \$\$
- System/Error values: XYZ, ERROR, NULL
- Empty fields: 489 events have no value at all.

2. What share of the orders is affected? A total of **28.03%** of the orders are currently failing quality checks (2,415 out of 8,616 total events).

Ask Assistant

Concierge actions

Ask questions and get guidance. The Assistant may suggest Concierge actions.

AI Assistant —
business_orders:business_data:business:orders:transactions:currency
(alias: currency)

google_gemini (gemini-pro-late)

Entity: business_orders:business_data:business:orders:transactions:currency (alias: currency) (FQM)

You

Which currency values are failing, what share of the orders is affected, and what should the application team fix?

AI Assistant

Based on the Fields Quality Monitoring (FQM) analysis for the currency field, here is the breakdown of the issue:

1. Which currency values are failing? The field expects strict 3-letter ISO currency codes based on the configured regex pattern: `^([UR|GBP|JPY|CAD|AUD])$`. The values currently failing this validation include:

- Truncated/Malformed codes: G, US, EU
- Non-standard formats: DOLLAR, EURO, \$\$
- System/Error values: XYZ, ERROR, NULL
- Empty fields: 489 events have no value at all.

2. What share of the orders is affected? A total of **28.03%** of the orders are currently failing quality checks (2,415 out of 8,616 total events).

Ask Assistant

Concierge actions

Ask questions and get guidance. The Assistant may suggest Concierge actions.

- 1

The question

Which values fail, what share, what to fix?
- 2

Failing values, by kind

Truncated · non-standard · error values · empty.
- 3

28.03% of orders

22.35% invalid, 5.68% empty.
- 4

What to fix at the source

Standardise outputs, fix truncation, handle nulls.
- 5

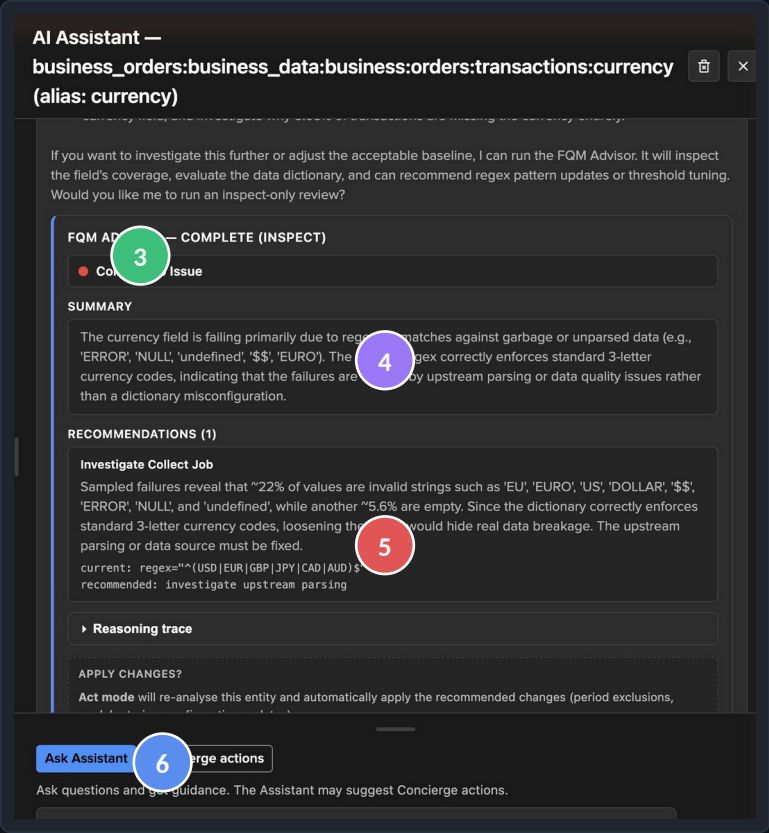
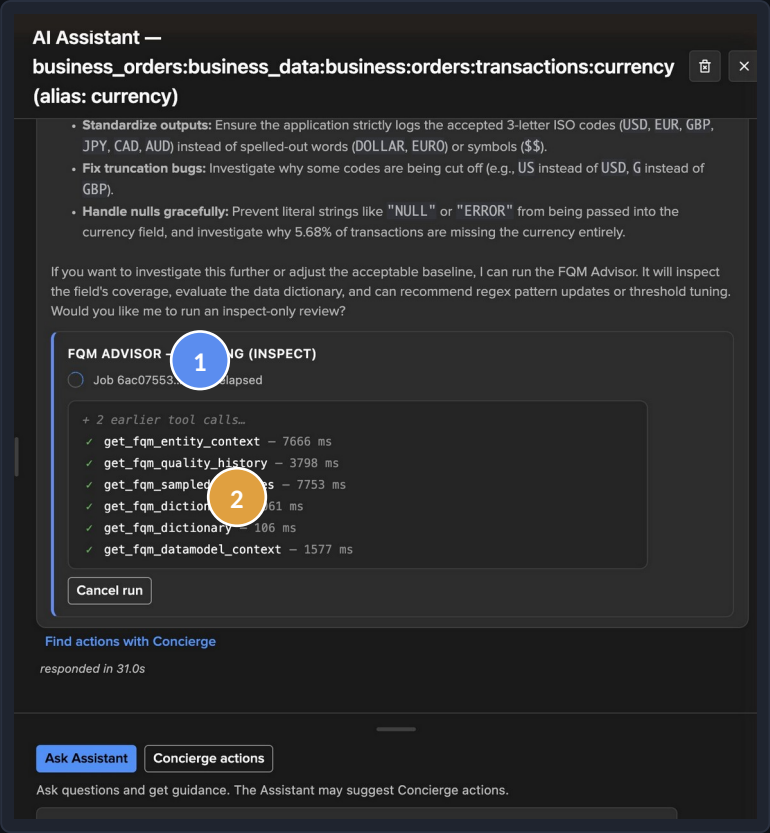
FQM Advisor — proposed action

Inspect the failing values and the dictionary.
- 6

Run now (inspect)

Read-only run, launched from the chat.

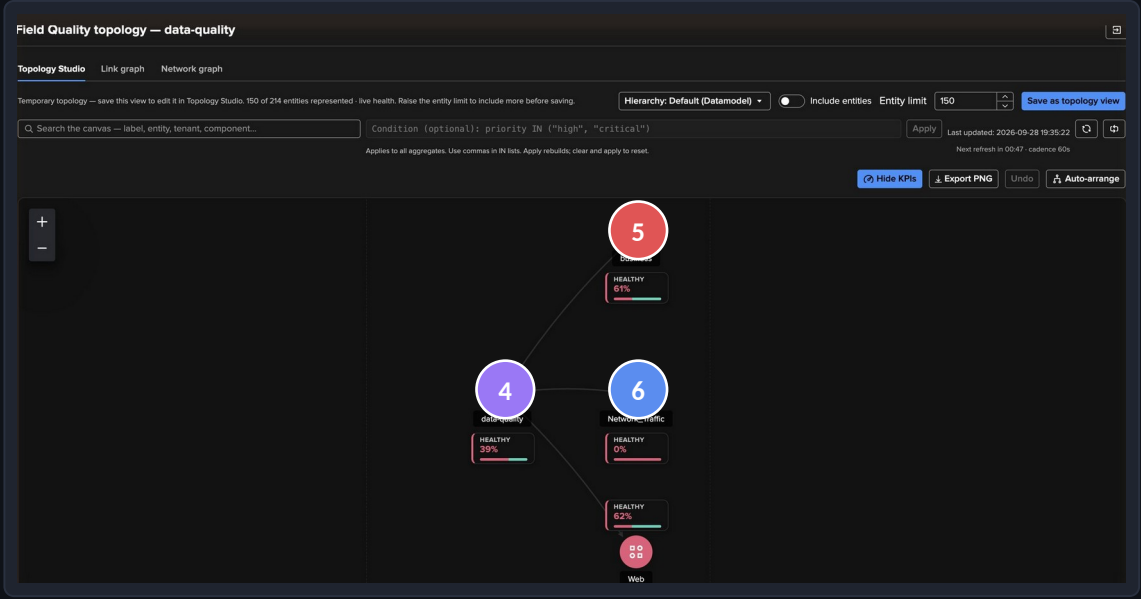
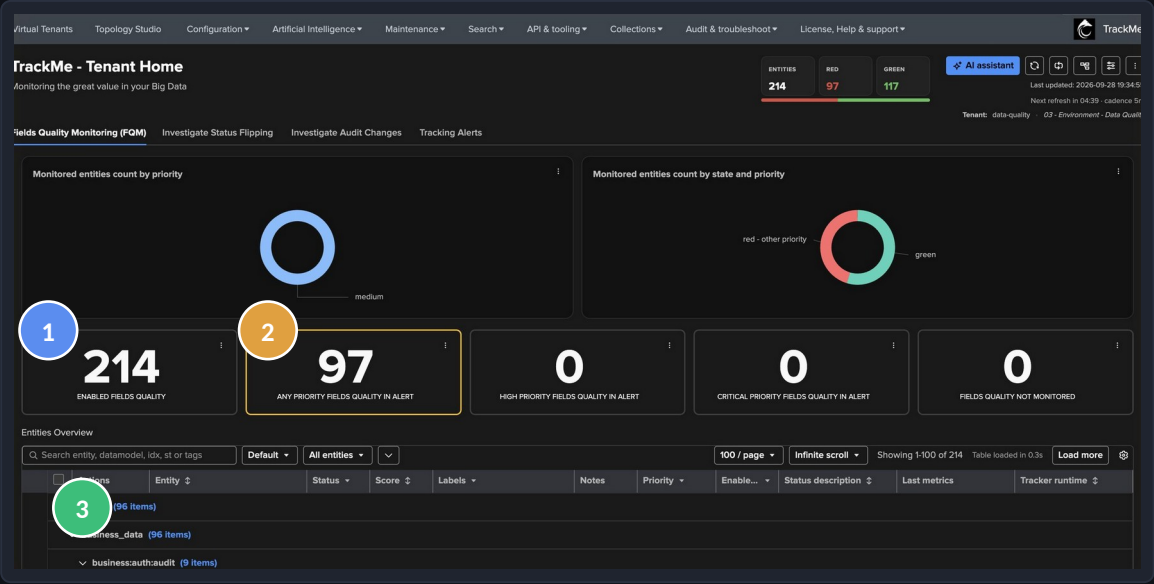
The FQM Advisor verdict — keep the rule, fix the data



- 1 Running (inspect)
Read-only mode: no change is possible.
- 2 Evidence gathered
Entity context, history, sampled failures, dictionary, data model.
- 3 Collect Job Issue
The verdict category.
- 4 The rule is right
Failures come from garbage or unparsed data, not the dictionary.
- 5 Do not loosen the regex
"Loosening the regex would hide real data breakage."
- 6 Act mode
Applies recommended changes — only after consent.

Three AI Advisor steps: draft the dictionary • tighten one rule • review the verdict.

A mature FQM tenant — 214 fields, months of history



- 1 **214 enabled fields**
Across business and CIM trackers.
- 2 **97 in alert**
Quality issues to route to owners.
- 3 **Grouped by data model**
business · Network_Traffic · Web.

- 4 **Field Quality topology**
Tenant → data model, with live health.
- 5 **business — 61% healthy**
Aggregated from every field below it.
- 6 **Network_Traffic — 0%**
Save it as a Topology Studio view.

Where we are

In two walkthroughs

● Created

a Field Quality tenant and two trackers — a CIM Cisco ASA job and a non-CIM JSON orders job

● Measured

with simulations on 10,000 events, then sampling sized by Auto define across the full 24 hours

● Tuned

the CIM dictionary (25% → 66.7%) and an AI-drafted dictionary, tightened per field

● Found

a CIM defect on action, and a currency problem on 28% of orders confirmed by the FQM Advisor

What made it work

Dictionary first

CIM generates it, the AI dictionary advisor drafts it — you decide, rule by rule, in the editor.

Read the category

Value does not match is a defect. Field missing by design is a dictionary setting.

Representative samples

Sampling with Auto define covers the whole window; head mode only sees the latest events.