RelativityOne API Decision Matrix for SIEM Alerting

Executive Summary

This guide helps you decide which RelativityOne APIs to poll for your SIEM integration. APIs are organized by **alert priority** (what failures matter most), **polling frequency** (how often to check), and **data returned** (what you can alert on).

Quick Decision Framework

If You Want to Monitor...	Use This API	Poll Frequency	Priority
Processing job failures	Processing Set Manager	1-5 min	Critical
Production job failures	Production Queue Manager	1-5 min	Critical
Imaging job failures	Imaging Job Manager	1-5 min	Critical
Integration Points failures	Object Manager (query Job History RDO)	5 min	High
Security events	Security Alerts (native)	N/A - Push	Critical
User activity & compliance	Audit API	5-15 min	Medium
dtSearch index issues	Object Manager (query dtSearch Index RDO)	15 min	Medium
Structured Analytics failures	Structured Analytics Job Manager	5-10 min	High
Workspace health	Workspace Manager	15-30 min	Low
Worker/server health	Worker monitoring endpoints	1-5 min	Critical

---

Tier 1: Critical Alerting APIs (Poll Every 1-5 Minutes)

1. Processing Set Manager API

Purpose: Monitor processing jobs (inventory, discovery, publishing)

REST Endpoint:

```
GET <host>/Relativity.REST/api/Relativity.Processing.Services.IProcessingModule/Processing Set
Manager/GetProcessingSetAsync
```

Key Fields to Monitor:

Field	Alert Condition	Severity
InventoryStatus	GUID indicates "Failed" or "Stopped"	Critical
DiscoverStatus	GUID indicates "Failed" or "Stopped"	Critical
PublishStatus	GUID indicates "Failed" or "Stopped"	Critical
HasRunningJobs	false when expected true	Warning
EnvironmentErrors	Any value present	Critical
DataSourceHasJobLevelErrors	true	Critical
DataSourceHasDocumentLevelErrors	true	High

Status GUIDs to Watch:

- Poll the SetState object to check status across all processing phases
- A "Paused" status typically indicates agent issues

Recommended Alert Rules:

```
IF EnvironmentErrors != null THEN CRITICAL
IF DataSourceHasJobLevelErrors == true THEN CRITICAL
IF Status == "Paused" for > 30 minutes THEN WARNING
IF HasRunningJobs == false AND expected_jobs > 0 THEN WARNING
```

---

2. Production Queue Manager API

Purpose: Monitor all production jobs across the environment

REST Endpoint:

```
GET <host>/Relativity.REST/api/relativity-productions/{versionNumber}/production-queue
```

Method: `GetAllAsync()` - Returns all production jobs in queue

Key Fields to Monitor:

Field	Alert Condition	Severity
JobID	Track for stuck jobs	Info
ProductionID	Correlate with production name	Info
WorkspaceID	Scope alerts by workspace	Info
Errors	Array not empty	Critical
Status	"Failed", "Error", "Cancelled"	Critical

Recommended Alert Rules:

```
IF Errors array length > 0 THEN CRITICAL
IF job in queue > 4 hours without progress THEN WARNING
IF same JobID appears for > 2 hours THEN investigate stuck job
```

---

3. Imaging Job Manager API

Purpose: Monitor imaging set jobs and mass imaging operations

REST Endpoints:

```
# Get imaging set status
GET <host>/Relativity.Rest/API/relativity-imaging/{versionNumber}/workspaces/{WorkspaceID}/imaging-sets/{ImagingSetID}

# Get document imaging status
GET <host>/Relativity.Rest/api/relativity-imaging/{versionNumber}/workspaces/{WorkspaceID}/documents/{DocumentArtifactID}/status
```

Key Status Values to Monitor:

Status	Meaning	Alert
`Staging`	Job started, not submitted	Info
`Preparing files`	Splitting into batches	Info
`Submitting`	Documents going to queue	Info
`Imaging`	Active processing	Info
`Completed`	Success	None
`Completed with Errors`	Partial success	High
`Error`	Job failed	Critical

Key Fields:

Field	Alert Condition	Severity
`# documents with errors`	> 0	High
`Last Run Error`	Any value	Critical
`Status`	"Error" or "Completed with Errors"	Critical/High

Note: Imaging Sets support native email notifications - configure `Email Notification Recipients` field for completion alerts.

Tier 2: High Priority APIs (Poll Every 5-10 Minutes)

4. Audit API

Purpose: Security monitoring, compliance, forensic analysis, user activity tracking

REST Endpoints:

```
# Query audit records
POST <host>/Relativity.REST/api/relativity.audit/{versionNumber}/workspaces/{workspaceID}/audits/query

# Query instance-level audits (set workspaceID to -1)
POST <host>/Relativity.REST/api/relativity.audit/{versionNumber}/workspaces/-1/audits/query

# Get audit metrics
GET <host>/Relativity.REST/api/relativity.audit.metrics/workspaces/-1/audit-metrics/
```

Key Actions to Monitor:

Action Type	Use Case	Alert Condition
`RelativityScriptExecution`	Script performance	Execution Time > threshold
`Login`	Security monitoring	Multiple failures
`Export`	Data exfiltration risk	Large export volumes
`MassEdit`	Bulk changes	Unauthorized mass operations
`Delete`	Data loss prevention	Unexpected deletions
`Permission Change`	Security	Elevation of privileges

Query Example for Failed Logins:

```
json

{
  "request": {
    "objectType": {"artifactTypeID": 1000042},
    "fields": [{"Name": "Timestamp"}, {"Name": "User Name"}, {"Name": "Action"}],
    "condition": "'Action' == 'Login Failed'"
  }
}
```

Recommended Alert Rules:

```
IF Login Failed count > 5 in 10 minutes for same user THEN HIGH
IF Export > 10,000 documents THEN MEDIUM (review)
IF MassDelete > 100 documents THEN HIGH
IF RelativityScriptExecution time > 30000ms THEN WARNING
```

5. Integration Points Job History (via Object Manager)

Purpose: Monitor scheduled Integration Points jobs

REST Endpoint:

```
POST <host>/Relativity.Rest/api/Relativity.ObjectManager/{versionNumber}/workspace/{workspaceID}/object/query
```

Query for Job History RDO:

```
json

{
  "request": {
    "objectType": {"Name": "Job History"},
    "fields": [
      {"Name": "Job Status"},
      {"Name": "Start Time (UTC)"},
      {"Name": "Integration Point"},
      {"Name": "Items Transferred"},
      {"Name": "Items with Errors"}
    ],
    "condition": "'Job Status' IN ['Error - Job Failed', 'Completed with Errors', 'Validation failed']"
  }
}
```

Job Status Values to Alert On:

Status	Meaning	Severity
`Pending`	Waiting for agent	Info (if > 30 min, Warning)
`Validation`	Being validated	Info
`Validation failed`	Config error	Critical
`Processing`	Running	Info
`Completed`	Success	None
`Completed with Errors`	Item-level errors	High
`Error - Job Failed`	Job-level failure	Critical
`Suspending`	Being suspended	Warning
`Suspended`	Paused for update	Warning

Recommended Alert Rules:

```
IF Job Status == "Error - Job Failed" THEN CRITICAL
IF Job Status == "Validation failed" THEN CRITICAL
IF Job Status == "Completed with Errors" AND Items with Errors > 100 THEN HIGH
IF Job Status == "Pending" for > 30 minutes THEN WARNING
IF Next Scheduled Runtime (UTC) is blank THEN CRITICAL (scheduled job stopped)
```

---

6. Structured Analytics Job Manager API

Purpose: Monitor Structured Analytics operations (email threading, near-duplicate detection, etc.)

REST Endpoint:

```
POST <host>/Relativity.REST/api/relativity-structured-analytics/{versionNumber}/workspaces/{workspaceID}/jobs/{jobID}/status
```

Key Fields:

Field	Alert Condition	Severity
`Status`	"Failed", "Cancelled"	Critical
`ErrorMessages`	Any present	Critical
`PercentComplete`	Stuck at same value	Warning

Tier 3: Medium Priority APIs (Poll Every 15-30 Minutes)

7. Object Manager API (General RDO Queries)

Purpose: Query any Relativity object for status monitoring

REST Endpoint:

```
POST <host>/Relativity.Rest/api/Relativity.ObjectManager/{versionNumber}/workspace/{workspaceID}/object/query
```

Use Cases:

A. dtSearch Index Monitoring:

```
json
```

```
{
  "request": {
    "objectType": {"Name": "dtSearch Index"},
    "fields": [
      {"Name": "Name"},
      {"Name": "Status"},
      {"Name": "Fragmentation Level"}
    ],
    "condition": "'Status' IN ['Build Error', 'Compression Error']"
  }
}
```

B. Processing Set Status:

```json
{
  "request": {
    "objectType": {"Name": "Processing Set"},
    "fields": [
      {"Name": "Name"},
      {"Name": "Status"},
      {"Name": "Documents Remaining"}
    ]
  }
}
```

Rate Limit Warning: Object Manager has a rate limit of **1,000 requests per minute per web server**. Include `X-Kepler-Referrer` header with your app GUID.

---

8. Workspace Manager API

Purpose: Monitor workspace health and configuration

REST Endpoint:

```
GET <host>/Relativity.Rest/API/relativity-environment/{versionNumber}/workspace/{workspaceID}
```

Key Fields:

Field	Use Case
`Status`	Workspace availability
`ResourcePool`	Resource allocation
`SqlServer`	Database health correlation

API Authentication

All APIs require authentication via one of:

1. **Bearer Token (Recommended for SIEM)**:

```bash
# Get token
curl -X POST "<host>/Relativity/Identity/connect/token" \
  -H "Content-Type: application/x-www-form-urlencoded" \
  -d "grant_type=client_credentials" \
  -d "scope=SystemUserInfo" \
  -d "client_id=<your_client_id>" \
  -d "client_secret=<your_client_secret>"

# Use token
curl -X GET "<endpoint>" \
  -H "Authorization: Bearer <access_token>" \
  -H "X-CSRF-Header: -"
```

2. **Basic Authentication** (simpler but less secure):

```bash
curl -X GET "<endpoint>" \
  -H "Authorization: Basic <base64_encoded_credentials>" \
  -H "X-CSRF-Header: -"
```

Required Headers for All Requests:

- `X-CSRF-Header: -` (required, set to empty or single dash)
- `Content-Type: application/json`
- `Authorization: Bearer <token>` or `Basic <credentials>`

Implementation Architecture Recommendations

Option A: Direct SIEM Polling

```
┌──────────────┐    ┌──────────────────┐    ┌──────────┐
│ RelativityOne │───▶│   SIEM Poller    │───▶│  Splunk/  │
│     APIs      │    │   (scheduled)    │    │  Datadog  │
└──────────────┘    └──────────────────┘    └──────────┘
```

Option B: Middleware Service (Recommended)

```
┌──────────────┐    ┌──────────────────┐    ┌──────────┐    ┌──────────┐
│ RelativityOne │───▶│  Azure Function  │───▶│  Event   │───▶│  SIEM/   │
│     APIs      │    │   or Lambda      │    │   Hub    │    │ PagerDuty│
└──────────────┘    └──────────────────┘    └──────────┘    └──────────┘
```

Polling Schedule Template

Time Interval	APIs to Poll
Every 1 min	Processing Queue (if active jobs)
Every 5 min	Production Queue, Imaging Sets, Integration Points Job History
Every 15 min	Audit (security events), dtSearch indexes, Structured Analytics
Every 30 min	Workspace health, general system status
Every 1 hour	Full audit export for compliance

Sample SIEM Alert Definitions

Splunk Alert Examples

```
spl
```

```
# Processing Job Failure
index=relativity sourcetype=processing_jobs
| where Status IN ("Failed", "Error", "Paused") OR EnvironmentErrors!=""
| stats count by WorkspaceID, ProcessingSetName, Status
| where count > 0

# Integration Points Failure
index=relativity sourcetype=integration_points
| where JobStatus IN ("Error - Job Failed", "Validation failed")
| stats count by IntegrationPointName, WorkspaceID
| where count > 0

# Multiple Failed Logins
index=relativity sourcetype=audit Action="Login Failed"
| stats count by UserName
| where count > 5
```

Datadog Monitor Examples

```yaml
# Processing Job Monitor
name: "RelativityOne Processing Job Failure"
type: metric alert
query: "sum(last_5m):sum:relativity.processing.errors{*} > 0"
message: "Processing job has failed in RelativityOne"

# Production Queue Monitor
name: "RelativityOne Production Queue Stuck"
type: metric alert
query: "avg(last_15m):avg:relativity.production.queue_time{*} > 7200"
message: "Production job stuck in queue for >2 hours"
```

Priority Implementation Order

1. **Week 1**: Processing Set Manager + Production Queue Manager
2. **Week 2**: Imaging Job Manager + Integration Points Job History
3. **Week 3**: Audit API (security events)
4. **Week 4**: Structured Analytics + dtSearch monitoring
5. **Ongoing**: Fine-tune thresholds based on baseline data

Key Considerations

1. **No Native Webhook Support**: RelativityOne does not push events - you must poll
2. **Rate Limits**: Object Manager: 1,000 req/min; plan polling intervals accordingly
3. **Authentication Tokens**: Bearer tokens expire - implement refresh logic
4. **Workspace Scope**: Most APIs require iterating across workspaces
5. **Instance vs Workspace**: Some APIs (Audit, Security) support instance-level queries with `workspaceID = -1`

Additional Resources

- RelativityOne Platform APIs
- REST API Authentication
- OpenAPI Specification Files (downloadable OASFiles.zip)