
Sample report

Ignition MRI — Mapped Risk Inspection

Backup taken	2026-10-12 14:43 Central
Platform	Ignition 8.3.8
Report date	2026-10-12
Prepared by	Artek Integrated Solutions · artekis.io

Every risk found in the supplied gateway backup, ranked by severity and rated for effort to fix, with the evidence for each. The scan reads the backup only — it cannot see the PLC at the far end of a device connection, the database the queries run against, or the operator who clicks the wrong thing at shift change.

About this sample. This is a real MRI, run the same way as a customer's, on Artek's own public demo project. We loaded the project onto a fresh test gateway with no databases or tag providers connected, took a backup, and scanned it. Findings caused only by that test setup are listed separately in section 4. Some evidence is summarized rather than listed, because the same project runs on our public demo server.

1. Summary for the people who own the budget

This gateway runs one large Perspective project: a showcase of demos, community tools and industry screens built up over several years. The one finding that could stop production is in recovery: a restore onto a new server would show error boxes on every screen that uses two third-party chart and grid modules, because a backup does not carry modules. Twenty-three web endpoints answer anyone who can reach the server, with no login. Seventy-one error handlers swallow failures without logging them, so a broken screen shows blank data instead of an error. The cheapest fixes this week are a filter box that has never worked, two alarm-count headers that call an alarm function in a form known to fail, and one query that reads a whole table.

2. What we looked at

Backup	<code>mri-sample.gwbk</code> , 32.8 MB (63.2 MB unpacked, 6,722 files), type ALL, sha256 <code>9ac175a88fb9c997</code>
Gateway	Ignition 8.3.8, standard edition, single gateway (no redundancy)
Projects	1 enabled, 0 snapshots; 888 views; 66 named queries; 23 WebDev endpoints
Code examined	3,917 script blocks, 1,750 Perspective event scripts, 358 inline database calls
Tags	None on this test gateway (see section 4)
Databases referenced	Named queries use 4 connection names; none are configured on the test gateway

3. Findings, ranked

#	SEVERITY	EFFORT	CATEGORY	FINDING
MRI-01	production-stopping	half-day	8	A restore without two third-party modules breaks 21 component instances
MRI-02	embarrassing	half-day	2	23 web endpoints answer without a login
MRI-03	embarrassing	multi-day	1	71 error handlers hide failures instead of logging them
MRI-04	embarrassing	quick-fix	1	A table's quick-filter box has never run
MRI-05	embarrassing	quick-fix	1	Two alarm-count headers call queryStatus in a form known to fail
MRI-06	embarrassing	multi-day	4	3 named queries paste parameters into the SQL as raw text
MRI-07	embarrassing	quick-fix	4	One named query reads a whole table with no limit
MRI-08	embarrassing	multi-day	7	539 of 888 views are not reachable from any page, dock or script
MRI-09	cosmetic	quick-fix	3	Three bindings re-run every one to two seconds
MRI-10	cosmetic	half-day	2	Two hard-coded IP addresses
MRI-11	cosmetic	multi-day	3	One view holds 436 components
MRI-12	cosmetic	half-day	7	12 of 66 named queries are never called
MRI-13	cosmetic	half-day	7	24 library functions have no caller in the project
MRI-14	cosmetic	half-day	8	The gateway is two patch releases behind
MRI-15	cosmetic	quick-fix	8	31 named queries and the message handlers are still in the 8.1 binary format

MRI-01 — A restore without two third-party modules breaks 21 component instances

Severity **production-stopping** · Effort **half-day** · Category 8 Upgrade readiness

What happens. Views use components from the Embr Charts and Embr Periscope modules. A gateway backup does not carry modules, so restoring onto a new server shows a "component not found" error in each of those places until the modules are installed again at a compatible version.

Evidence. Component types `embr.chart.chart-js` (4 instances), `embr.periscope.embedding.react` (15), `embr.periscope.embedding.portal` (1) and `embr.periscope.embedding.view` (1), across the project's views.

Fix. Record each third-party module's name, version and license with every backup, and keep the module files beside it. Check they are installed before restoring.

MRI-02 — 23 web endpoints answer without a login

Severity **embarrassing** · Effort **half-day** · Category 2 Security exposure

What happens. The project's WebDev resources have authentication turned off and no role check, so anyone who can reach the gateway can call them. Several accept uploads or saves: PDF tools, a PDF save handler and a demo API. On a plant network that means anyone on the network; on a server reachable from the internet it means anyone.

Evidence. 23 WebDev resources with `require-auth=false`, no roles, and HTTPS not required. Endpoint paths are withheld in this public sample.

Fix. Turn on authentication in each resource's settings, require HTTPS, and check the user's role inside each handler before it reads or writes anything.

MRI-03 — 71 error handlers hide failures instead of logging them

Severity **embarrassing** · Effort **multi-day** · Category 1 Scripts that fail quietly

What happens. When these scripts fail, nothing is recorded. The screen shows empty data, and the problem gets blamed on "no data" rather than on the script. 31 handlers only `pass` or `return`; 40 more neither log nor re-raise.

Evidence. For example, the calendar export handler in `views/Exchange/Artek Calendar/Calendar Home` has three `except Exception: pass` blocks (lines 19, 24 and 28), and the same code is copied into three other calendar views.

Fix. Log a warning with the error and the inputs in every `except`, and return a value the caller can check. Fix the shared calendar code once and update its copies.

MRI-04 — A table's quick-filter box has never run

Severity **embarrassing** · Effort **quick-fix** · Category 1 Scripts that fail quietly

What happens. The script that applies the filter is set to Client scope. Perspective scripts run on the gateway, so a Client-scoped action never executes: users type into the box and nothing happens, with no error anywhere.

Evidence. `views/Periscope/Table`, component `Toolbar/QuickFilterField`, property-change script with `scope: "C"`.

Fix. Change the action's scope to Gateway.

MRI-05 — Two alarm-count headers call queryStatus in a form known to fail

Severity **embarrassing** · Effort **quick-fix** · Category 1 Scripts that fail quietly

What happens. `system.alarm.queryStatus` is called with `state` as a list of strings. On Ignition 8.3.4 that raises an error every time, so the alarm count never updates. We have not confirmed whether 8.3.8 still fails, so we rate it as a fix rather than a guess.

Evidence. Header views in the water demos, custom property `numAlarms` (2 sites).

Fix. Leave out the `state` argument and filter on each alarm's state in the loop.

MRI-06 — 3 named queries paste parameters into the SQL as raw text

Severity **embarrassing** · Effort **multi-day** · Category 4 Database queries

What happens. QueryString parameters are inserted into the SQL as text instead of being bound as values. If any caller passes user input, that is SQL injection. Even when it doesn't, the database can't reuse a query plan.

Evidence. `Exchange/Document Management/Documents/Get` and `Get History` (5 search parameters each) and `Embr-Charts/getLogEvents` (1).

Fix. Trace each caller. Replace with value parameters, or build the variable part from a fixed list of allowed fragments.

MRI-07 — One named query reads a whole table with no limit

Severity **embarrassing** · Effort **quick-fix** · Category 4 Database queries

What happens. The query returns every row into the gateway's memory. It gets slower every day the table grows, until the screen times out.

Evidence. `DatacenterDemo/OperatorLog/GetLogs` , `SELECT *` with no `WHERE` .

Fix. Add a date range and a row limit, or page the results.

MRI-08 — 539 of 888 views are not reachable from any page, dock or script

Severity **embarrassing** · Effort **multi-day** · Category 7 Dead code and orphans

What happens. Unused views cost Designer time and export size, and they make it hard to tell which copy is live. Another 117 views sit in folders that a reachable view builds paths into at runtime; those are listed as possible dynamic targets, not as dead.

Evidence. Page routes, docks and script references start from 41 views, which reach 232 in all. The unreachable views are mostly popups, objects and diagnostics screens under the water demo folders, plus Exchange resources nothing links to, such as `Exchange/DataCenter/Overview` and `Exchange/Document Management/Document Viewer` .

Fix. Review the list against the dynamic-path folders, then delete in batches with a backup taken first.

MRI-09 — Three bindings re-run every one to two seconds

Severity cosmetic · Effort quick-fix · Category 3 Performance

What happens. Each open session re-runs these bindings on every tick. Two of them drive the alarm counts in MRI-05, so each tick also runs an alarm query.

Evidence. `now(1000)` and `now(2000)` expression bindings in two header views and the data center alarm label.

Fix. Slow them to 5 to 10 seconds, or update the count from a tag change instead of polling.

MRI-10 — Two hard-coded IP addresses

Severity cosmetic · Effort half-day · Category 2 Security exposure

What happens. The project breaks when it moves to another server or network.

Evidence. One in the project's message-handler scripts, one in a session property. Addresses withheld in this public sample.

Fix. Read the address from a configuration tag or gateway setting.

MRI-11 — One view holds 436 components

Severity cosmetic · Effort multi-day · Category 3 Performance

What happens. The view is slow to open and to re-render, and hard to maintain.

Evidence. `views/Feature Demos/Design System/Demo` .

Fix. Split it into embedded views or a flex repeater.

MRI-12 — 12 of 66 named queries are never called

Severity cosmetic · Effort half-day · Category 7 Dead code and orphans

What happens. These are removal candidates, unless a caller builds the query path at runtime.

Evidence. Eight alarm-analysis queries in one demo folder, `Exchange/Document Management/Tags/Get` , two TimeClock queries and `Alarms/Alarm History` .

Fix. Confirm there is no dynamic caller, then delete.

MRI-13 — 24 library functions have no caller in the project

Severity cosmetic · Effort half-day · Category 7 Dead code and orphans

What happens. Removal candidates. Some may be called from outside the project or run by hand, so confirm before deleting.

Evidence. Seven statistics helpers in `SPC` , three each in `Helpers.Table` and `API.openweathermap` , and others in `Charts` , `Utils` and `TimeClock` .

Fix. Check each against the entry points the scan can't see (WebDev callers, scheduled tasks, script console use), then remove in batches.

MRI-14 — The gateway is two patch releases behind

Severity **cosmetic** · Effort **half-day** · Category 8 Upgrade readiness

What happens. Fixes released since 8.3.8 are missing, and the gap grows with every release.

Evidence. `backupinfo.xml` : version 8.3.8. 8.3.10 is available.

Fix. Plan a patch upgrade on a test copy first, with MRI-01's modules checked for compatibility.

MRI-15 — 31 named queries and the message handlers are still in the 8.1 binary format

Severity cosmetic · Effort quick-fix · Category 8 Upgrade readiness

What happens. These resources can't be read or compared on disk, so version control and reviews skip them.

Evidence. 31 named queries stored as `data.bin` , and `ignition/event-scripts/data.bin` holding two message handlers.

Fix. Open each in the 8.3 Designer and save it, which converts it to the readable format.

4. Findings caused by the test setup

These came from the empty test gateway, not from the project. On a customer's gateway each one would be a real finding.

FINDING	WHY IT APPEARED HERE
23 named queries point at database connections that don't exist	The test gateway has no databases configured
1,889 tag paths point at tag providers that don't exist	The project's demo tag providers were not copied
Logins and sessions can travel over plain HTTP	A fresh gateway allows HTTP by default
Scheduled gateway backups are off	Off by default on a fresh gateway
The backup contains no database data	A gateway backup never does; on a real system, pair it with a database backup
Single gateway, no redundancy	The test gateway is a single container

5. Quick wins

- 1. MRI-04: change the quick-filter script's scope to Gateway.
- 2. MRI-05 and MRI-09 together: fix the queryStatus call and slow its polling.
- 3. MRI-07: add a date range and a row limit to the operator log query.

6. Acceptable as-is

MRI-02 is intentional here. The demo is open to the public on purpose, so anyone can try the tools without asking us for credentials. On a plant system the same finding would need a login on every endpoint.

Keeping old and experimental views in a demo project is a reasonable choice, so MRI-08, MRI-12 and MRI-13 matter less here than they would in a plant system. The same counts in a production project would be worth a cleanup.

7. Scope and limitations

The scan read the backup only: projects, scripts, queries, tags, alarms and gateway settings. It cannot see PLCs, the databases behind the connections, how operators use the system, the load it runs under, or gateway logs, since a backup carries none. Categories with nothing to report on this test gateway: alarm configuration and tag structure (no tags were present).

CONTROL	EXAMINED
Script blocks (scripts, transforms, handlers)	3,917
Perspective event scripts	1,750
Named queries	66
Inline database calls	358
Polling bindings	19
Tags walked	0

8. Method

Scanner `mri_scan.py` (sha256 `ed1eea42cf1c1b21`) run against the unpacked backup. Each finding was checked against the project source before it went into this report, and severities were adjusted to the actual consequence where the scanner's default overstated it.