



INDITION DASHBOARDS

Indition Dashboards Setup Guide

This guide is for administrators deploying and configuring Indition Dashboards. Use the native Indition installer, environment configuration, roles and scheduled-job system.



September 2026

 © 2026 Indition Corp. All rights reserved. Indition™ and the Indition logo are trademarks of Indition Corp. All other product names, logos, and brands are the property of their respective owners.

Contents

- Prepare the environment
- Deploy and run native migrations
- Configure API data access
- Configure relationships and labels
- Configure Central Records starters
- Roll out Admin Home and role templates
- Configure scheduled delivery
- Verify the job runner
- Configure optional AI and maps
- Acceptance and rollback
- Set up Indition AI for Dashboard users
- Review the AI information boundary
- Configure geographic data for maps
- Plan a staff and publisher rollout
- Complete the instance acceptance checklist

Configure Indition Dashboards and its optional AI assistance through the native platform. This guide connects staff access, API sources, AiCore configuration, readable relationships, maps and scheduled delivery so users can build with the data they are allowed to see.

 Wherever you see this mark, that feature uses Indition AI. AI can propose dashboard and widget settings or explain supported aggregate results. You review proposed settings before applying and saving them. Availability depends on the site's AI settings, provider readiness and your permissions; manual Dashboard building remains available without AI.

FROM A BUSINESS QUESTION TO A DASHBOARD DRAFT

Describe what you need. Review the dashboard it builds.

Create a whole dashboard. In New Dashboard, describe the subject, period and questions, then choose Create with AI. The proposed board opens as an unsaved draft using sources in your permitted catalog.

Design or revise a widget. Describe the figure or chart you want, or request a change to an existing widget. Build it fills the ordinary editor so you can inspect the source, measure, dates and conditions.

Explain an eligible result. Explain this result produces supported observations from bounded aggregate evidence. It does not prove why a change happened. Inspect matching records when you need to understand the underlying work.

Prepare the environment

Use a working Indition admin environment with Xpress, Admin and staff authentication. Install the business modules whose sources you want to report on. For a standalone API environment, use the standalone API deployment installer to establish its framework, theme, environment file and application database; configure its API data connection separately.

COMPONENT	DASHBOARD RESPONSIBILITY
Xpress	Native framework services, Select2, XLSX writer and PDF engine
Admin	Dashboard builder, templates, storage, exports and delivery orchestration
XUser or native staff authentication	Active staff identities and roles
Api	Sources from enabled tables on the configured API data connection
Messaging	Email and SMS transport when scheduled delivery is wanted

COMPONENT	DASHBOARD RESPONSIBILITY
AiCore	Optional AI creation assistance and aggregate explanations
Other business modules	Their own permitted dashboard providers

Shop is optional. API dashboards do not need commerce tables. Storage is not required for Dashboard XLSX exports. Avoid customer-specific code or debug parameters that bypass the native environment.

Deploy and run native migrations

Deploy the tested module tags with your normal SVN release process. Run each module's native install or upgrade process. Fresh installs include the curated Dashboard migrations; existing environments must apply the relevant patches.

Required native schema areas include configurable dashboards, templates and staff preferences; scheduled delivery and alert state; PDF delivery and operational tools; metrics and saved views; API Dashboard read permission; API table access, relationships and business field metadata. Use the installer and pending-patch list to establish the complete dependency chain.

Do not apply only the last patch and assume all prerequisites exist. Use the native pending-patch list and verify completion. The Excel controls use the installed Dashboard schema and native Xpress services.

For a fresh PostgreSQL application database, the standalone deployment installer includes the uuid-osp extension needed by the application. Source data does not need to be copied into the application database merely to appear on a dashboard.

Configure API data access

Use the native environment connection named db for Dashboard definitions, permissions and delivery state. API queries use the selected API data connection, normally apiSource. Set the API source through the existing API settings and environment configuration; do not redirect dashboard reads to a customer database with hardcoded credentials.

Enable the API module and its data provider. In API Tables, enable only the tables intended for access. Dashboard sources consider that enabled selection. Enabling a lookup or address table also changes what the API table selection exposes, so review it as an access decision.

For non-administrator staff, grant the API dashboard read route, Api.DashboardData.Read, with the appropriate read access and the required Dashboard routes. API sources still enforce source enabled state and masking. An API token client is not a replacement for an authenticated Dashboard staff account.

Configure relationships and labels

Use native database foreign keys where available. For references without FKs, open API / Relationships as a superadmin. Choose the source table and field, related enabled table, compatible unique related key and readable label fields. Several fields from the related table can form a label, such as Name, City and Region.

The default configuration access is superadmin-only. Relationship routes are registered for later delegation to other roles. Do not widen the API table list just to hide a setup error; enable each necessary safe lookup deliberately.

Open API / Dashboard Fields to assign business table names, field labels, descriptions and numeric formats. Geographic roles identify street, city, state, postal code, country, latitude and longitude. Metadata supplies presentation hints; it does not grant table access or create a relationship.

For optional freshness, choose a permitted timestamp field and threshold in minutes. Zero disables the check. Use a suitable indexed field on large sources; a failed or timed-out check reports unavailable without replacing the widget result with zero.

Refresh the API schema through its native control after schema or FK changes. Reopen the dashboard after changing enabled tables, relationships, masks or metadata so the catalog reflects the current configuration.

Configure Central Records starters

In API settings, enter table=label mappings under Central Records, one per line or comma separated. For example: records=Records, transactions=Transactions and tasks=Tasks. Only permitted enabled tables become starter widgets.

Check the API data section of New Dashboard. Starters contain counts, available date trends and record tables, named for the included subjects. They are editable starting points. Save or publish a layout before expecting it to become a default dashboard.

Roll out Admin Home and role templates

Open Manage Templates using an account with dashboard_template_publisher. The capability is explicit and is not automatically assigned merely because someone is an administrator.

Create a template from a sample or blank board. Set its name, widgets and Role Settings. Choose the target roles, precedence when several roles have templates and whether personal copies are allowed. Save the draft and publish it. Saving a draft alone does not change the role's live layout.

Enable Admin Home uses these dashboards for the Admin Home location. A role's Override Standard Dashboard Route can still choose another landing page; review that setting if a user lands elsewhere.

Verify one ordinary role account. It should see only permitted sources and assigned dashboards. Hidden shared widgets stay in the definition so access changes do not erase the layout.

Configure scheduled delivery

Install and configure Messaging before enabling delivery. Test email at `/admin/Messaging/admin/emailSent/test`. Set the applicable System workflow mail values, including `MAIL_METHOD`, `MAIL_SENDER_ADDRESS` and the SMTP or sendmail settings required by the chosen method. `SMTP_HOST`, `SMTP_PORT` and `SENDMAIL_PATH` are method-specific settings, not interchangeable transports.

Configure Messaging's native phone/SMS settings for SMS. Dashboard Settings reuses those transport credentials; it does not store a separate mail server or SMS account.

Open `/admin/Admin/dashboardSettings/index`.

SETTING	DEFAULT AND PURPOSE
Enable dashboard notifications	No; master switch for scheduled delivery and delivery links
Send via Email	No; enables the email channel
Send via SMS	No; enables the SMS channel
Timezone	America/New_York; initial timezone for new schedules
Public HTTPS address	Empty; this environment's address for SMS PDF links
PDF link lifetime	72 hours; allowed range 1-168 hours
Enable explicit address lookups	No; optional missing-address geocoding
OpenCage API key	Empty; required only when explicit geocoding is enabled

Enable only the intended channels. Use the actual HTTPS host for this instance. Email recipients and SMS links receive a report evaluated with the schedule owner's permissions, so choose recipients appropriate for that data.

Verify the job runner

The native scheduled-job configuration uses service `Admin.DashboardDelivery.Run`. The installed minute polling configuration relies on the site's existing `JobRunner`; do not create a second Dashboard runner alongside it.

Check that cron runs against the intended site root and current code on exactly the intended server. Duplicate cron on an older server can fail even while a manual run on the current server succeeds. Manual and scheduled invocation should both report start, disabled/no-due outcomes and completion in the native execution log and `JobRunner` debug file when debug output is enabled.

The job processes at most ten due occurrences per invocation. Missed periods move to the next future occurrence rather than sending a backlog. Each occurrence is claimed before transport attempts. Review per-recipient history and Messaging logs; transport acceptance is not proof of inbox delivery.

Configure optional AI and maps

For AI, install AiCore and use its native provider settings. Enable `dashboard_create`, `dashboard_widget` and `dashboard_explain` as needed in native AI settings. Aggregate explanations require `dashboard_explain`. Verify provider availability before exposing AI actions; AI is optional for every manual Dashboard workflow.

For map pins, configure safe numeric coordinates and the relevant address fields. Existing coordinates do not require geocoding. To enable Locate missing addresses, configure the explicit address-lookup switch and OpenCage key. Each click sends at most five permitted address strings, uses short timeouts and a 30-day local cache. Page loads do not automatically geocode the entire dataset.

Acceptance and rollback

Before rollout, verify a count against the source, a named multi-select condition, a saved metric, a saved view, a record drill-down, XLSX and Actions PDF. For maps, verify state zoom, full-map reset, address-type restrictions and counts for a known set. Preview a scheduled report before a real test delivery.

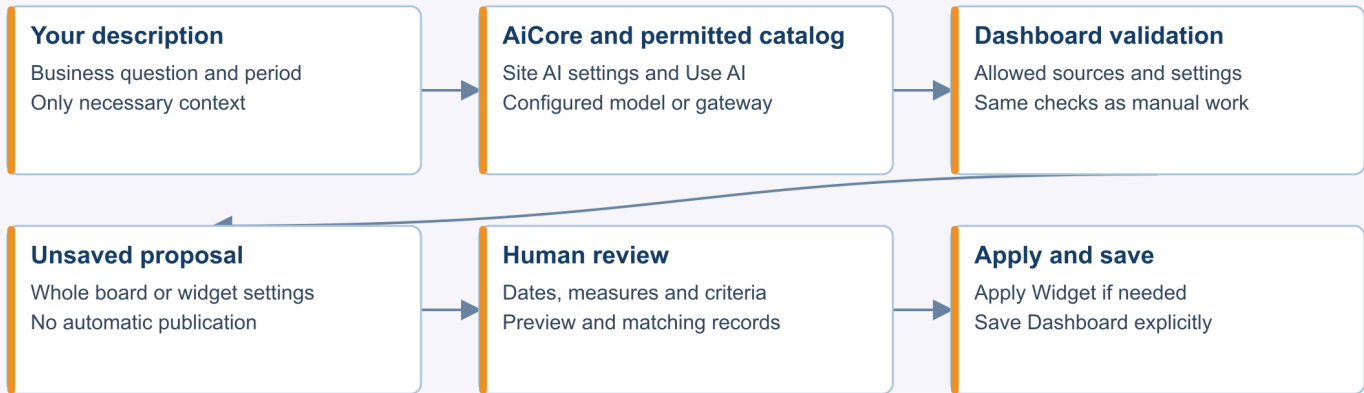
Check active staff access and source masking using an ordinary account. Inspect fresh and warm loads to confirm schema discovery is cached. Document optional service configuration without recording secrets in this pack.

If a deployment fails, restore the previous immutable module tags through the normal process and consult the migrations' actual rollback behavior. Dashboard schemas and saved data should not be deleted as a shortcut. Keep deployment provenance in the controlled operational record.

Set up Indition AI for Dashboard users

AiCore is an optional native module. Install and initialize it through the normal Indition module process. Do not configure separate AI credentials inside a Dashboard widget. Dashboard calls the shared AiCore client and follows the site's AI switches, access rules, model choices and spending controls.

AI proposes a definition and the user decides



AI design receives catalog and prompt text, not computed result figures.

AI integration readiness and review

1. Open AI / AI Settings using an account authorized to manage AI configuration.
2. Confirm that AI is enabled for the website and that the selected provider path is ready. The site can use its configured provider keys or the Indition gateway according to its native AI setup.
3. Under Where staff can use AI, find the Dashboards section. Enable whole-dashboard creation, widget description and result explanation independently according to the intended rollout.
4. Under the model-per-job controls, review Designing dashboards. Choose a supported model and its available effort setting. The same purpose is used by Dashboard's native AI calls.
5. Grant Use AI to the intended staff role in Manage Permissions. A role's data permissions must also allow the sources the staff member is expected to use.
6. Review site and per-person spending limits in AI Settings. Budget exhaustion should be handled through the site's approved AI administration process.
7. Sign in as an ordinary intended user and verify which AI controls are offered. A superadmin-only test cannot establish ordinary staff access.

The native Dashboard features are `dashboard_create`, `dashboard_widget` and `dashboard_explain`. Whole-board creation and widget assistance return settings that the builder validates. They do not save or publish automatically. Aggregate explanation is a separate workflow that supplies bounded evidence rather than individual records.

Try a non-sensitive description using the user's actual catalog, such as a current-month count and daily trend from an enabled source. Inspect the resulting source, measure, period and criteria, then apply and save explicitly. Verify that a restricted source does not appear or become accessible through the AI request.

Review the AI information boundary

The design paths send the permitted source catalog, relevant presentation choices and the text entered by the user. They do not send computed dashboard figures. Catalog names and condition choice labels can themselves describe the business, so review the site's provider and data-handling policy before enablement. Users should not include private record details in a design description.

AI integration uses two different data boundaries

Design a dashboard or widget

- Permitted catalog and your description
- No computed figures or arbitrary SQL
- Returned settings are validated

Explain an eligible result

- Bounded server-generated aggregate evidence
- No individual rows, addresses or category values
- Returns supported observations, not causes

AiCore policy for both paths

- Feature enabled, Use AI, provider ready
- Configured model, limits and usage accounting

Information typed into a design prompt is sent with that request. Avoid private record details.

Different AI requests carry different data

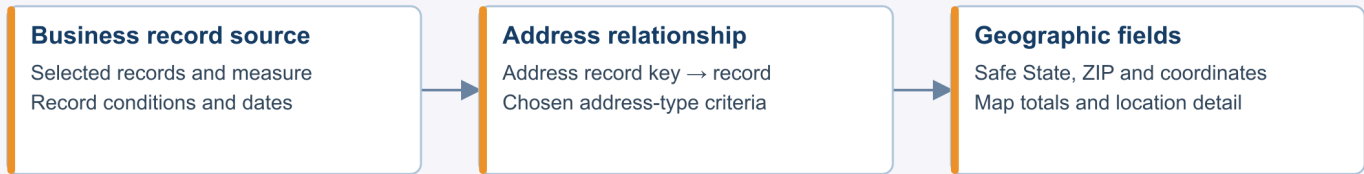
Explain this result sends bounded server-generated aggregate evidence. That path excludes individual rows, addresses and category values. Its observations refer to supplied facts and comparisons; they do not establish causes. Do not describe the entire AI integration as sending no data: a design request contains a prompt and catalog, and an explanation request contains selected aggregate evidence.

Dashboard calls are attributed to the Admin module in native AI usage accounting with the relevant feature and Dashboard purpose. Review usage and limits through AiCore. Keep provider keys, gateway secrets and raw sensitive records out of user manuals and support bundles.

Configure geographic data for maps

A map can use geography exposed by any compatible permitted provider. For API tables, configure a safe State field on the record table or a related address table. Optional ZIP, street, city, latitude and longitude roles supply more detail. For commerce, use the geographic sources contributed by Order where installed and permitted.

Map permitted records through geographic fields



Use one series with its geographic fields or a related address source. Two unrelated series do not create the map.

Address relationships and geographic field roles

Enable each required API table and its safe fields. If the address belongs to a separate table without an FK, configure the record-key relationship through API Relationships. Choose a readable compound label, such as Name, City and Region. Geographic field roles are maintained in API Dashboard Fields, independently of the relationship label.

Build the map from one compatible series and its geographic source. For a related address table, select Map address source and add address-type criteria appropriate to the question, such as Billing or Service Location. Keep business-record status conditions on the record source. Test a known state and inspect matching records before sharing the dashboard.

State totals can overlap when a record has qualifying addresses in several states. Coordinate pins require safe valid latitude and longitude. Explicit address geocoding is optional; configure the provider only when approved and test the bounded lookup action separately.

Plan a staff and publisher rollout

Decide who needs to view personal dashboards, customize them, use AI, manage reusable metrics, publish role templates and administer sources. These are different responsibilities. API Tables, Relationships and Dashboard Fields begin with superadmin configuration access; registered routes allow deliberate later delegation.

A publisher can assign roles and priorities, but publication does not grant data access. Test a representative role with its ordinary permissions. Check that its dashboard resolves to the intended template, that unavailable sources are handled correctly and that a personal copy cannot alter other users' layouts.

For a shared metric, document the business population and owner. Updating the metric changes dependent widgets. For scheduled reports, use an active accountable owner and approved recipients. Recipient-specific views narrow reports within the owner policy; they do not impersonate recipients.

Complete the instance acceptance checklist

Use a small known dataset or an authorized sample. Check count, dates, a named multi-value exclusion, a nested criterion, shared filters and a saved view. Compare an eligible prior period, verify a percentage with a nonzero denominator, and check the unavailable behavior for a zero denominator.

Inspect readable record labels, page and complete Excel exports, chart-data export and Actions / Download PDF. On a geographic map, verify state zoom, reset, region detail, available coordinate pins, record drill-down and export. Test large-result limits deliberately instead of assuming every source can export without bounds.

For AI, test whole-board creation, widget revision and aggregate explanation separately. Confirm provider failure or exhausted limits produce an actionable outcome and do not save an incomplete arbitrary definition. For delivery, preview the rendered report and test email and SMS through approved recipients before enabling the regular schedule. Verify the scheduled runner, not only manual invocation.

Record the responsible administrator, configured providers, acceptance outcomes and deployment provenance in the controlled operational record. The public documentation pack does not contain internal release identifiers, customer database addresses or credentials.