



INDITION FORMS

User Manual

Building, designing and running forms end to end

FormBuilderCore Module v6.6 · August 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

- Getting around
- Creating a form
- Field reference
- The Advanced Designer
- Conditional logic
- Multi-step forms
- Stop points
- API calls
- Processing: what happens after submission
- Publishing
- Working with submissions
- Everyday tasks
- Troubleshooting

About this manual

This manual walks through the life of a form in Indition Forms: creating it, placing and configuring fields, designing its look, splitting it into steps, adding conditional logic, branching and stop points, wiring API calls, publishing it, and working with the submissions it collects. It is written for the people who build and run forms day to day; administrators setting up the platform should start with the **Setup Guide**.

Getting around

Everything lives under **Forms Management**. The forms list shows every form with its designs, pages, processing chain and submission counts; forms with more than three processing steps show a count that opens the full list. Each row's actions menu is the hub for a form:

Action	What it opens
Edit / Configure	The form's fields and settings.
Form Design	The list of designs; each opens in the Advanced Designer.
Form Links & Embed Codes	Every page the form is published on, with copy buttons and embed snippets.
Submissions	The submissions workspace (Submissions and Partial Submissions tabs).
Processing	The post-submission pipeline.
Clone	A copy of the form with its fields, designs and settings (never submissions).
Update Form Name / Delete	Rename, or remove the form with its designs and submissions.

Breadcrumbs at the top of every screen carry the form's name so you always know whose designer or grid you are in.

Creating a form

- 1 **Create New Form** from the forms list and give it a name.
- 2 Add fields in the field editor: pick a type, name it, set its options, save. Repeat until the form's data model is complete. Fields can be added later at any time.
- 3 Choose or create a **design** — the visual layout the world will see. A form can have several designs, each published at its own page; a short landing variant and a full-detail variant can feed the same submissions.
- 4 Configure **processing** (what happens after submission) and **form settings**.
- 5 Publish: pick up the page link or embed code from **Form Links & Embed Codes**.

Field reference

Every placed field opens a settings drawer with the shared basics — label, placeholder, help text, required flag, validation, default value, layout position and CSS classes — plus the type-specific settings below.

Text and identity fields

Field	Settings and behavior
Text / Text Area	Length limits, format validation, default value; text areas add row height.
First Name / Last Name	Dedicated name entry that maps cleanly to contact records and email personalization.
Email	Format validation; the address processing uses for respondent emails and save-&-resume links.
Phone	Masked domestic phone entry.
Intl Phone	Country picker with per-country number validation.
Website / Password / Hidden	URL validation; obscured entry; invisible values for campaign codes and URL-parameter capture.
Mask Input Field	Define any fixed entry format (IDs, plates, policy numbers) with a mask.

Address fields

Street Address, City, State, Country and Postal Code place individually so a design can ask for exactly the parts it needs; postal code can look up city and state to shorten entry.

Choice fields

Field	Settings and behavior
Radio Button / Checkboxes / Dropdown	Option lists (hand-entered, API-loaded, or dependent on a parent field), default selection, stacked or inline layout.
Single Checkbox / Yes-No Button	Consent boxes and binary choices.
Image Choice Options	Options presented as images — product pickers, style choices.
Star Rating	Configurable rating scale.
Disclaimer	Scrollable terms text the visitor acknowledges.

Option constraints can cap how many times an option may be chosen across all submissions — seats at a session, slots in a class — and hide or refuse the option once its quota is used.

Date, time and number fields

Field	Settings and behavior
Date / Time / Date Time	Pickers with format, min/max bounds and time granularity.
Datetime Range	Start/end pair with timezone awareness — bookings, rental windows.
Number / Price	Numeric and currency entry with validation.
Calculated Field	A formula over other fields that recalculates live as the visitor types — totals, scores, quotes. Repeating-section rows can feed section totals.

Document and structure fields

Field	Settings and behavior
File	Allowed extensions, per-file and total size limits, multiple files per field. Dangerous executable types are always refused.
Esignature	Drawn signature with appearance options; pair with DocuSign for formal envelope signing.
Repeating Section	A template row of fields the visitor can repeat — line items, dependants, work history — with minimum/maximum row counts, a required-rows rule and totals. Rows can also auto-generate from data (Platinum).
Header / Rich Text / Divider / Button	Structure and copy for the page; extra buttons can carry API or partial-submission actions.
Partial Submission Button	Captures everything entered so far as a partial submission when clicked.

Connected fields

Field	Settings and behavior
API Populated Field	Its value or option list comes from a configured API call; it can validate the visitor's entry against the API and block submission until it passes. See <i>API calls</i> below.
Cra Email Subscription	Subscribes the respondent to selected platform email lists on submission.
Amazon Order Number	Validated order reference for support and warranty forms.

Map a field to a URL parameter (its mapping name) and links like `?email=ann@example.com` pre-fill it when the page opens — on hosted pages and embeds alike.

The Advanced Designer

Open a design to enter the Advanced Designer: the field palette on the left, your canvas in the middle, and the settings panel on the right.

Laying out the canvas

- Drag fields from **Available Elements** onto the canvas. Rows accept up to four columns; drop a field beside another to split the row, or above/below to make a new row.
- **Add All Element Fields** places every unplaced field at once, following the design's base column layout (one, two or three columns). The operation batches server-side, shows progress, and recovers automatically if an individual field stalls.
- **Additional Design Elements** — headers, rich text, images, dividers and empty rows — come from the panel above the palette.
- **Remove All Rows** clears the canvas and returns every field to the palette, safely resetting placement.

Styling

- **Themes** apply a complete look in one click. Selecting a theme replaces the previous theme's look entirely, including backgrounds the new theme does not set.
- The **Style panel** adjusts typography, colors, input and button styling for the whole design with live preview; each element's drawer can override styling individually via the Advanced tab.
- **Settings → Page Template** chooses the page frame around the hosted form, with visual previews of each template. **Settings → Page URL** controls the published address.
- **Settings → Spam Protection** adds Google reCAPTCHA to the design.

Saving and previewing

Style changes stage as you work and flush when you save; the browser back button is honored. Use **Page View** to see the form inside its page template (multi-step designs show numbered step boundaries there), and the preview link to open the live page. The designer preview renders exactly what the published page will.

Conditional logic

Conditional logic tailors the form to each visitor: every field can show or hide based on earlier answers, evaluated live in the browser as they type.

Building a rule

- 1 Open the field's drawer and its **Logic** tab.
- 2 Choose the action — **Show this field when...** or **Hide this field when...**
- 3 Choose the match mode — **all** of the conditions (every one must hold) or **any** (one is enough).
- 4 Add conditions: pick a field, an operator and a value. Operators: *is*, *is not*, *contains*, *does not contain*, *starts with*, *greater than*, *less than*, *greater or equal*, *less or equal*.
- 5 Save. The rule compiles into the form's condition engine and runs immediately on the published form.

Worked examples

Goal	Rule
Ask for a company name only from businesses	Show <i>Company Name</i> when <i>Customer type</i> is <i>Business</i> .
Reveal a follow-up on low satisfaction	Show <i>What went wrong?</i> when <i>Rating</i> is less than 3.
Hide shipping fields for pickup orders	Hide the address fields when <i>Delivery method</i> is <i>Store pickup</i> .
Gate a section behind two answers	Show <i>Financing details</i> when all match: <i>Purchase type</i> is <i>Financed</i> , <i>Amount</i> greater than 1000.

How logic behaves

- **Required-when-visible** — a required field that logic hides does not block submission; the requirement returns the moment it shows. Hidden fields do not submit values.
- **Conditions are reusable** — the same condition can drive several fields, and editing it updates them together.
- **Legacy rules** — rules created in earlier releases keep working and remain editable as a toggle list alongside builder-managed rules.
- **Multi-step forms** — logic works across steps; a field on step 3 can depend on an answer from step 1.
- **Embeds** — rules are baked into the form's configuration, so they run identically on hosted pages and embeds with no extra requests.

Keep rules positive where you can: *show when X is Y* is easier to reason about than stacked hides. Use *hide* for the exceptional case, not the default.

Multi-step forms

Multi-step turns one long form into a sequence of steps with its own navigation, progress bar and per-step behavior.

Turning it on and building steps

- 1 Enable **Multi-Step** in the designer. The canvas divides into steps and the step rail appears on the left edge.

- 2 Add steps and drag fields into each. The rail collapses the steps you are not editing; click a step to focus it, or **Expand all** to see everything (which also turns the auto-collapse off, remembered per design).
- 3 Reorder steps by dragging their chips in the rail. Guard rules keep invalid orders out: a stop point cannot be the last step, branch targets must stay valid, and reordering is locked once the form has submissions.
- 4 Turn on **Include Progress Bar** and pick a style; the first style is preselected so save works immediately.

The step settings drawer

Each step's settings are grouped:

Group	Settings
Step content	Step heading and whether it shows; sub-labels; the step's preview title.
Navigation	Continue/back button labels and the Google Analytics label fired on continue.
Flow	Branch point on/off with the branch table; stop point on/off with its name and destination; whether the applicant may edit this step after submitting it; agent-interactive mode; reprocessing enablement.
Checks	Per-step CAPTCHA and query validation — the step will not advance until they pass.
Integrations	API calls triggered when the step completes (see <i>API calls</i> below).

The rail badges each step that carries special behavior — paused (stop point), branch, API call or validation — so the form's flow reads at a glance.

Branching

- 1 Mark a step as a **branch point** in its Flow group.
- 2 Add branches: each has a name and a target step, chosen from the form's steps. Every branch must name a valid target — the designer validates this on save.
- 3 Wire the branch choice to the visitor's answer on the branch-point step; on continue, the visitor routes to the matching step.

Combine branching with conditional logic for fine control: branch to a whole different path, and let logic tune individual fields inside each path.

The preview step

Enable **Include preview** to add a review step: visitors see their answers, grouped by step (each step's *preview title* labels its group), before the final submit. Pair it with *edit step after submission* to let them jump back and correct.

How multi-step submits

Each completed step submits its answers, so a visitor who abandons the form midway still appears under **Partial Submissions** with everything they finished. The final step's submit completes the submission and runs the processing chain.

Stop points

A stop point pauses the form at a chosen step for human review — the pattern behind applications, approvals and any process where a person checks before the workflow continues.

Configuring a stop point

- 1 In the step's **Flow** group, enable the stop point and name it (the name labels the queue and reports).
- 2 Choose the **destination** — where the visitor lands when their submission pauses.
- 3 Decide the post-stop permissions: may the applicant **edit the step after submission**? Is the step **agent-interactive** (your staff completes it with or for the applicant)? Is **reprocessing** allowed?
- 4 Optionally attach stop-point **API calls** and a stop-point **on-success** experience (the message the visitor sees when the submission pauses, distinct from final-submit success).

A stop point cannot be the last step — there must be somewhere for the submission to continue to. The designer enforces this when you reorder.

Reviewing stopped submissions

The **Stop Point Submissions** workspace is the queue:

- **The list** shows every paused submission with its stop point, arrival time and the answers captured so far; completed submissions move to their own list.
- **Open a submission** to review and edit its entries, or view its detail read-only, including a screenshot rendition for records.
- **Email the applicant** from a template — request a document, ask a question, send the save-&-resume link so they can continue themselves.
- **Continue processing** pushes the submission past the stop point — the remaining steps' processing (emails, API calls) runs.
- **Reprocess** re-runs a stopped submission's processing after corrections; each process shows its status so you can see exactly which call failed and run it again.
- **Delete** (or delete all) clears submissions that will not proceed.

The applicant's experience

On reaching a stop point the visitor sees the stop point's success message or destination. If the step allows editing after submission, their emailed link reopens it; agent-interactive steps can be completed by your team on their behalf. When review continues the submission, the normal processing chain takes over.

API calls

Forms call external systems at four trigger points; all share the same configuration model.

Where API calls run

Trigger	Typical use	Failure behavior
Processing step (on submission)	Push the finished submission to a CRM, ticketing or fulfillment system.	Logged; visible in response logs; reproducible.
Step boundary (multi-step Integrations)	Validate or enrich mid-form — verify an account or membership number before the visitor may continue.	The step does not advance; the field shows the API's message.
Stop point	Notify or query systems when a submission pauses; run again on reprocess.	Logged; reprocess re-runs the call.
Button click	Wire any form button to an API action.	Message shown at the button's field.

Configuring a call

- 1 Create the API processing entry with the endpoint URL and method.
- 2 Choose the **data format**: JSON, name/value pairs, or multipart form data (required when files are sent).
- 3 Choose **authentication**: static credentials in the request, or an *auth API* — a separate call that obtains a token first.
- 4 **Map fields to attributes**: each request attribute takes its value from a form field or a fixed value.
- 5 **Map the response**: response data can populate form fields (API-populated fields) — for example, an account lookup filling name and tier — or be stored with the submission.

API-populated fields

An **API Populated Field** is the visitor-facing end of a call: its value or its option list comes from the API, refreshed when its source fields change. Marked required, it blocks submission until the API confirms a valid value — the standard pattern for “verify before accept” forms.

Logs and reprocessing

- Every call is tracked; **response logs** are viewable per submission, with the request and response bodies.
- Failed calls surface in the stop-point workspace's process status, where **reprocess** re-runs them after you correct data or the remote system recovers.

Processing: what happens after submission

Under a form's **Processing**, add ordered steps; the chain runs top to bottom on every completed submission:

- **On success** — show a message, open a modal or redirect the visitor. Stop points can carry their own on-success variant for the paused experience.
- **Send email** — notify your team and/or send the respondent a confirmation; both templates support personalization tokens drawn from the form's fields.
- **File generation** — generate a document from the submission for records or hand-off.
- **API processing** — the on-submission API calls described above.

The forms list shows each form's processing chain; drag to reorder where order matters (for example, generate the file before the email that references it).

Publishing

Form Links & Embed Codes (in the form's actions menu) lists every page the form is published on, with copy buttons for the links and the embed codes:

- **Iframe Code** — paste into any site; size it from your page's CSS.
- **JavaScript Code** — a script tag that renders the form in a sandboxed frame which sizes itself automatically, on load and as the form grows or shrinks (steps, conditional fields). Available from the Silver plan; the embedding site must be on your allowed-domains list. No libraries are required on the host page.

Both channels serve the same published design, so a change saved in the designer is live everywhere at once.

Working with submissions

- The **Submissions** grid lists every completed submission with the form's fields as columns; **Partial Submissions** has its own tab; stopped submissions live in the **Stop Point Submissions** workspace.
- Per row: view the full submission, download it as a PDF, or delete it.
- The grid menu: **Export** (selected rows) and **Export All** open the export configuration — date range plus the columns to include; **Generate PDF of Selected Submissions** builds a combined PDF; **Delete selected** removes rows after confirmation.
- **Save & resume** submissions wait in the same workspace until the visitor finishes via their emailed resume link.
- **Response logs** on a submission show every API call it triggered.

Everyday tasks

Task	Where
Rename a form	Forms list → actions → Update Form Name
Clone a form	Forms list → actions → Clone (copies fields, designs and settings)
Add a field to a live form	Edit fields, then place it in each design that should show it
Change the page a form publishes to	The design's Settings → Page URL
Turn on spam protection	Advanced Designer → Settings → Spam Protection, or a per-step Check
See a submission's API calls	The submission's response logs
Re-run failed processing	Stop Point Submissions → the submission's process status → Reprocess
See who was blocked from embedding	The blocked-domains log

Troubleshooting

Symptom	Likely cause and fix
A field is missing from the page	It is unplaced in that design — drag it from Available Elements.
A required field never blocks submission	Conditional logic is hiding it; hidden fields are exempt by design.
Continue does nothing on a step	A Check is failing (CAPTCHA or query validation) or a step API call rejected — look for the message at the relevant field.
Embedded form shows “not available on this website”	The embedding domain is not on the allow-list; the blocked-domains log names the refused host.
A stopped submission never continued	It is waiting in Stop Point Submissions — continue or reprocess it there.
The designer looks broken after an upgrade	Hard-refresh to load the new designer assets.